

## **BAB I**

### **PENDAHULUAN**

#### **A. Latar Belakang**

Teknologi memegang peran penting di era globalisasi saat ini. Teknologi dapat mencakup berbagai kegiatan sehingga segala kesulitan dapat diatasi dengan adanya penerapan teknologi, penerapan teknologi memudahkan manusia dalam melakukan kegiatan sehari-hari salah satunya pemeliharaan, pengawasan dan perbaikan sarana dan prasarana fasilitas jalan.

*Traffic light* merupakan sebuah indikator lampu yang digunakan untuk mengendalikan lampu lalu lintas di suatu persimpangan dengan cara memberi kesempatan pengguna jalan masing-masing arah.

Masyarakat berhak mendapatkan fasilitas sebagai kompensasi dari pajak yang telah mereka bayar serta berhak melaporkan penyimpangan terhadap fungsi jalan serta pengendalian terhadap pemanfaatan jalan kepada penyelenggara jalan. Namun dibalik itu, ternyata dalam tahapan pelaksanaan banyak sekali terjadi kekurangan.

Salah satu permasalahan yang dihadapi saat ini adalah terdapat banyak fasilitas-fasilitas perlengkapan jalan yang rusak/ tidak berfungsi dengan baik (disfungsi) serta penanganan terhadap suatu fasilitas perlengkapan jalan lamban dilakukan perbaikan. Keterlambatan penanganan dan perbaikan terhadap fasilitas perlengkapan jalan. Selama ini, kegiatan rutin petugas untuk memeriksa lampu normal atau tidaknya dengan dilakukan patroli secara berkala selain menunggu

informasi langsung dari masyarakat. Hal ini dirasakan kurang efisien karena jumlah titik *Traffic Light* sangat banyak yang ada pada tiap daerah, sedangkan jumlah petugas patroli berjumlah sangat sedikit sekali. Hal tersebut dapat memperlambat penanganan masalah yang akhirnya juga mengakibatkan kerugian pada masyarakat, yaitu meningkatnya angka kerawanan sosial, baik itu kecelakaan lalu lintas maupun tindakan kriminal.

Saat ini teknologi informasi berkembang dengan pesat sehingga perlu adanya inovasi pelayanan sektor publik khususnya dalam hal *monitoring* sarana dan prasana fasilitas perlengkapan jalan guna mewujudkan *good governance*. Perangkat komputer dapat menjadi solusi untuk melakukan pengawasan terhadap fasilitas *traffic light* yang ada di jalan dengan cakupan yang luas sehingga dapat dilakukan dimana saja. Dalam pengembangannya *Esp8266* telah banyak digunakan dalam berbagai perangkat dengan berbagai fungsi yang berbeda-beda. *Esp8266* sudah memiliki modul *WiFi* yang terpasang di dalamnya, sehingga sangat membantu untuk mengembangkan arsitektur perangkat lunak *internet of things*.

Performa *Esp8266* mempunyai satu core processor berkecepatan *2.4Ghz*, sehingga mikrokontroler dapat mengurangi lagging saat pemrosesan data *input* dan *output*. Pada *Esp8266* mempunyai *board* yang berukuran kecil namun menyatu oleh jaringan *WiFi* dengan penggunaan energi yang lebih rendah, hal tersebut merupakan beberapa kelebihan dari *Esp8266* ini.

Berdasarkan latar belakang permasalahan yang telah diuraikan diatas maka penulis mengusulkan judul penelitian “**Prototype Sistem Monitoring Disfungsi**

*Traffic Light*". Dengan menggunakan *Esp8266*, sistem *monitoring* disfungsi *traffic light* dapat memberikan informasi yang lebih akurat dan tepat tentang gangguan yang terjadi tanpa harus melakukan pengecekan di lapangan. Bila terjadi *trouble*, sistem ini akan mengirimkan notifikasi untuk segera mengatasi masalah tersebut.

### **B. Rumusan Masalah**

Berdasarkan uraian dari latar belakang diatas maka dirumuskan masalah sebagai berikut :

Bagaimana membangun suatu *prototype* sistem *monitoring* disfungsi *traffic light*?

### **C. Batasan Masalah**

Adapun agar pembahasan menjadi lebih terarah dan tidak menyimpang dan juga sesuai dengan latar belakang yang sudah diuraikan sebelumnya, maka penulis membatasi permasalahan penelitian pada *Prototype Sistem Monitoring Disfungsi Traffic Light*. Batasan masalahnya antara lain :

Penelitian ini tidak dilakukan pembahasan terhadap fasilitas jalan lainnya serta perancangan simulasi lampu lalu lintas (*traffic light*) menggunakan *Esp8266* dibuat dalam bentuk simulasi.

### **D. Tujuan Penelitian**

Rumusan masalah yang telah didefinisikan sebelumnya, adapun tujuan dari penelitian dan perancangan sistem dalam penelitian ini adalah :

Untuk mengetahui cara membangun suatu *prototype* sistem *monitoring* disfungsi *traffic light*.

### **E. Manfaat Penelitian**

Salah satu fungsi dari penelitian adalah bisa memberikan manfaat dan kegunaan yang secara nyata akan diterima oleh berbagai pihak yang terkait dengan obyek penelitian secara luas. Dalam penelitian kali ini peneliti berharap bisa memberikan kegunaan. Penulisan tugas akhir ini memberikan manfaat ke beberapa pihak, antara lain :

#### 1. Bagi Peneliti

- a. Meningkatkan kemampuan berfikir tentang konsep seperti apa yang harus penulis angkat kedalam penelitian, tentunya berkaitan dekat dengan keadaan di Instansi yang menjadi tempat penelitian.
- b. Meningkatkan wawasan mengenai langkah-langkah dan tata cara membuat sistem yang berguna sehingga dapat diukur melalui keefektifan dalam penggunaan dan keefisienan dalam pemanfaatan teknologi IOT.
- c. Penelitian ini diharapkan bisa meningkatkan pengetahuan yang lebih pada spesifik ilmu Administrasi Publik, khususnya dalam konsep pelayanan publik, inovasi layanan publik dan *e-Government*.

#### 2. Bagi Pemerintah dan Masyarakat

- a. Memiliki suatu sistem yang baru sebagai instrumen kunci kebijakan dalam mengawasi dan mengontrol pengelolaan sarana dan prasarana fasilitas perlengkapan jalan sehingga dapat membantu pihak Pemerintahan Daerah.
- b. Sebagai umpan balik strategi pemerintah setempat dalam menjawab

era revolusi industri 4.0.

### 3. Bagi Universitas/ Perguruan Tinggi

- a. Menciptakan mahasiswa dan mahasiswi yang mampu menerapkan sebuah sistem yang menjadi dasar tempat penelitian.
- b. Menambah referensi literatur kepustakaan di Universitas Muhammadiyah Parepare dan bahan masukan bagi Mahasiswa/i yang akan melakukan penelitian lebih lanjut khususnya di bidang yang berkaitan peningkatan kualitas pelayanan maupun teknologi.

## **F. Sistematika Penulisan**

Sistematika penulisan skripsi ini dibagi ke dalam lima bab yang masing-masing menjelaskan bagian-bagian yang berbeda namun tetap memiliki korelasi satu dengan yang lain. Setiap bab-nya terdiri dari beberapa sub tersendiri. Diawali dengan bab pendahuluan dan diakhiri dengan bab penutup yang berisi kesimpulan dan saran-saran. Dalam menyusun sistematika penulisan, penulis menguraikan ke dalam lima bab sebagai berikut :

### **BAB I : PENDAHULUAN**

Pada bab awal ini akan dijelaskan tentang latar belakang penulisan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

### **BAB II : TINJAUAN PUSTAKA**

Pada bab ini berisi uraian tentang teori-teori yang digunakan sebagai referensi dalam "*Prototype Sistem Monitoring Disfungsi Traffic Light*". Bab ini juga membahas mengenai berbagai teori yang

mendasari dari analisis permasalahan yang berhubungan dengan topik yang dibahas.

### **BAB III : METODE PENELITIAN**

Pada bab ini berisi mengenai metode perancangan yang digunakan dalam membangun sistem ini. Pada bab ini juga membahas tentang identifikasi masalah, analisis kebutuhan sistem yang meliputi kebutuhan masukan, proses dan keluaran suatu sistem yang akan dibangun menggunakan *Flowchart* dan UML.

### **BAB IV : HASIL DAN PEMBAHASAN**

Pada bab ini membahas mengenai hasil dari tahap analisis dan perancangan sesuai dengan metode yang digunakan pada alat dan aplikasi yang dibuat serta hasil dari pengujian serta bentuk-bentuk implementasi sistem beserta penerapannya.

### **BAB V : PENUTUP**

Pada bab akhir ini berisi tentang kesimpulan dari penelitian yang dilakukan dan saran-saran yang diberikan untuk pengembangan lebih lanjut agar tercapai hasil yang lebih maksimal.

### **DAFTAR PUSTAKA**

Pada bagian ini akan dipaparkan tentang sumber-sumber literatur, tutorial, buku maupun situs-situs yang digunakan dalam pembuatan tugas akhir ini.

### **LAMPIRAN**

## BAB II

### TINJAUAN PUSTAKA

#### A. Penelitian Terdahulu

Penelitian sebelumnya merupakan suatu hal yang sangat perlu dan dapat dijadikan sebagai data pendukung. Salah satu data pendukung yang menurut peneliti perlu dijadikan bagian tersendiri adalah penelitian terdahulu yang relevan dengan permasalahan yang sedang dibahas dalam penelitian ini. Dalam hal ini, fokus penelitian terdahulu yang dijadikan acuan adalah terkait dengan masalah teknologi informasi. Oleh karena itu, peneliti melakukan kajian terhadap beberapa hasil penelitian berupa tugas akhir/ skripsi atau jurnal melalui *internet*.

Pertama, penelitian yang dilakukan oleh Isman (2020) dalam sebuah jurnal yang berjudul **“Perancangan Aplikasi *Monitoring* Pemadaman Listrik di PT. PLN (Persero) Area Parepare Berbasis *Android*”**. Penelitian ini bertujuan untuk mengembangkan perangkat lunak aplikasi informasi pemadaman listrik berbasis *Android*. Dari hasil pengujian menggunakan *Black Box*, dengan menguji *listing* program diperoleh bahwa sistem telah berhasil berjalan setelah dilakukan pengujian penginputan informasi pemadaman sesuai dengan fungsional aplikasi.

Kedua, penelitian yang dilakukan oleh R. Sulistyorini, A. Ilham (2019) dalam sebuah jurnal yang berjudul **“Pengembangan Sistem *Monitoring* Lampu Lalu Lintas Berbasis *Microcontroller* Dengan Fitur SMS Jaringan GSM”**. Penelitian ini bertujuan untuk membuat sistem perangkat lampu lalu lintas

dengan kemampuan untuk mendiagnosa sendiri (*self-diagnose*) fungsi kerja rangkaian elektroniknya, dimana jika mengalami gangguan, segera dapat terdeteksi dan melaporkannya menggunakan layanan *Short Message Service* (SMS) ke suatu unit *monitoring* yang dapat diletakkan dimanapun dan terjangkau layanan komunikasi nirkabel *Global System for Mobile communications* (GSM).

Ketiga, penelitian yang dilakukan oleh Agus Setiawan, Achiruddin (2018) dalam penelitian sebuah jurnal yang berjudul “**Aplikasi Prasarana Lalu Lintas Pada Dinas Perhubungan Pemprov Kalimantan Selatan Berbasis Website**”. Penelitian ini bertujuan menyampaikan informasi petugas bagian lalu lintas dan memudahkan masyarakat dalam memberikan laporan-laporan tentang kerusakan-kerusakan pada prasarana-prasarana lalu lintas tersebut. Hasil dari pengujian aplikasi ini menyimpulkan bahwa fungsi yang diharapkan semuanya berhasil sesuai uji fungsionalitas. Kesimpulannya aplikasi ini bersifat multi *user* dan menggunakan jaringan internet untuk pemberitahuan perbaikan jalan melalui *email gateway*.

Keempat, penelitian yang dilakukan Jufri (2017) dalam penelitian sebuah skripsi yang berjudul “**Simulasi Pengaturan Lampu Lalu Lintas (*Traffic Light*) Berbasis Mikrokontroler**”. Penelitian ini bertujuan untuk mengurai kemacetan atau kepadatan kendaraan lalu lintas pada persimpangan jalan berdasarkan kepadatan lalu lintas. Didalam Perancangan alat ini, digunakan beberapa rangkaian yaitu rangkaian *mikrokontroler* ATmega 8535, rangkaian sensor, rangkaian lampu led, Rangkaian lcd, modul rangkaian relay dan rangkaian *power*

*supply*. Metode ini harus ada implementasi sistem yang dikembangkan sebelum dibuat sebuah sistem *final*.

## B. Kajian Teori

### 1. *Prototype*

*Prototype* merupakan salah satu metode pengembangan perangkat lunak yang menggunakan pendekatan untuk membuat rancangan dengan cepat dan bertahap sehingga dapat segera dievaluasi oleh calon pengguna/ klien. Dengan metode *prototype* ini pengembang dan klien dapat saling berinteraksi selama proses pembuatan *prototype* sistem. Terkadang seringkali terjadi, klien hanya mendefinisikan secara umum apa yang dikehendaki tanpa menyebutkan proses masukan (*input*) dan keluaran (*output*) dari sistem yang akan dibuat. Untuk mengatasi ketidakselarasan tersebut maka harus dibutuhkan kerjasama yang baik di antara keduanya, sehingga pengembang akan mengetahui dengan benar apa yang dibutuhkan klien. Dengan demikian nantinya akan menghasilkan sebuah rancangan sistem yang interaktif sesuai dengan kebutuhan.

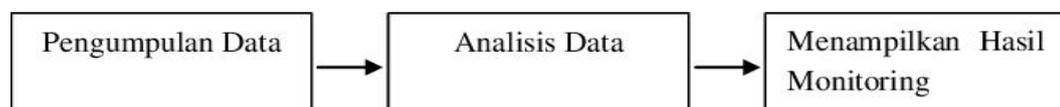
Sementara itu dalam suatu proses pengembangan sistem tidak jarang menghadapi keraguan mengenai efektifitas, efisiensi dan kualitas algoritma yang sedang dikembangkan kemampuan adaptasi sistem terhadap sistem operasinya atau tampilan yang sedang dirancangnya (Roger, S Presmman, 2012).

## 2. Sistem Monitoring

Sistem *monitoring* adalah pemantauan yang dapat dijelaskan sebagai suatu kesadaran tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari sesuatu (Rohayati, 2014). Sedangkan menurut Mudjahidin (2010) *Monitoring* adalah penilaian yang terus menerus terhadap fungsi kegiatan proyek di dalam konteks jadwal-jadwal pelaksanaan dan terhadap penggunaan *input-input* proyek oleh kelompok sasaran di dalam *konteks* harapan-harapan rancangan.

Berdasarkan dari kedua definisi *monitoring* tersebut, dapat disimpulkan bahwa *monitoring* adalah pengawasan atau pemantauan terhadap suatu kegiatan sehingga menghasilkan sebuah informasi yang berguna. Informasi yang dihasilkan dapat mempermudah dalam mengambil keputusan terhadap kegiatan kedepannya.

Menurut Ramayasa (2015) secara garis besar tahapan dalam sebuah sistem Monitoring terbagi ke dalam tiga proses besar seperti yang terlihat pada gambar 2.1 berikut ini



**Gambar 2.1** Proses dalam *Monitoring*

Proses-proses yang terjadi pada suatu sistem *Monitoring* dimulai dari pengumpulan data seperti data dari *network traffic*, *hardware information*, dan lain-lain yang kemudian data tersebut dianalisis pada proses analisis data dan pada akhirnya data tersebut akan ditampilkan (Ramayasa, 2015).

Beberapa keuntungan menggunakan sistem *monitoring* untuk kegiatan bisnis. Dalam jurnal Aprisa (2015) tentang sistem *monitoring* perkembangan proyek, menghasilkan sebuah sistem yang dapat membantu *admin* dalam proses pengolahan data perkembangan proyek, memudahkan *manager* dalam melakukan pemantauan perkembangan proyek setiap harinya tanpa harus datang langsung ke lokasi pengerjaan proyek. Keuntungan menggunakan sistem *monitoring* juga terdapat dalam Rohayati (2014) tentang sistem informasi *monitoring* data inventori. Seperti, proses pembuatan laporan dapat dilakukan secara cepat dan mudah dengan mencetak file laporan, dapat menghasilkan informasi tentang laporan data pegawai, laporan data barang, laporan data peminjaman, dan transaksi pengembalian yang dilihat dalam periode tahun dan bulan.

Beberapa keuntungan diatas tersebut, dapat disimpulkan bahwa sistem *monitoring* dapat membantu dalam proses pemantauan kegiatan bisnis, merekap kegiatan bisnis dan pembuatan pelaporan. *Monitoring* dapat dilakukan pada setiap tahapan kegiatan, apakah dari perencanaan ataupun setelah bagian pekerjaan tertentu diselesaikan.

### ***3. Traffict Light***

Lampu lalu lintas (*Traffict Light*) merupakan suatu perangkat yang diperlukan dalam proses pengaturan arus lalu lintas pada persimpangan. Pengaturan arus lintas tersebut dimaksudkan agar kendaraan-kendaraan pada masing-masing jalur bergerak bergantian tanpa ada saling ganggu antar arus lintas yang terjadi di persimpangan (Bram, 2011).

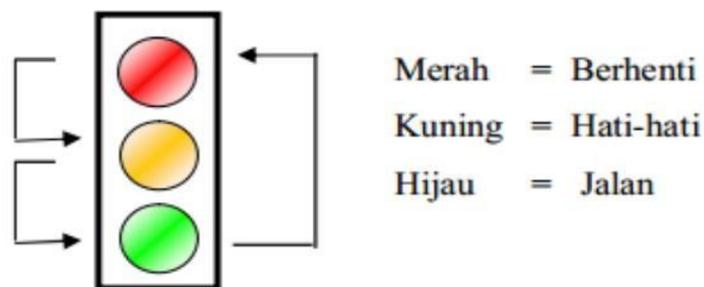
Lampu lalu lintas menurut UU no. 22/2009 tentang lalu lintas dan angkutan jalan ialah alat pemberi isyarat lalu lintas atau (APILL) merupakan lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyeberangan pejalan kaki (*zebra cross*), dan tempat arus lalu lintas lainnya. Lampu ini yang menandakan kapan kendaraan harus berjalan dan berhenti secara bergantian dari berbagai arah. Pengaturan lalu lintas di persimpangan jalan dimaksudkan untuk mengatur pergerakan kendaraan pada masing-masing kelompok pergerakan kendaraan agar dapat bergerak secara bergantian sehingga tidak saling mengganggu antar-arus yang ada.

Lampu lalu lintas telah diadopsi di hampir semua kota di dunia ini. Lampu ini menggunakan warna yang diakui secara universal untuk menandakan berhenti adalah warna merah, hati-hati yang ditandai dengan warna kuning dan hijau yang berarti dapat berjalan. Tujuan adanya lampu lalu lintas diantaranya adalah

1. Menghindari hambatan karena adanya suatu perbedaan arus jalan bagi pergerakan kendaraan.
2. Memfasilitasi persimpangan antara jalan utama untuk kendaraan dan pejalan kaki dengan jalan sekunder sehingga kelancaran arus lalu lintas dapat terjamin.
3. Mengurangi tingkat kecelakaan yang diakibatkan oleh tabrakan karena perbedaan arus jalan.

Sebelum melewati suatu persimpangan para pengemudi diwajibkan untuk mematuhi rambu-rambu yang telah ditetapkan. Lampu tersebut dipasang dalam sebuah *box* yang diberi tiang dan ditempatkan diujung sebelah kiri, ditengah-

tengah ruas jalan atau diatas setiap jalan pada persimpangan sehingga memudahkan para pengemudi untuk melihatnya. adapun warna lampu yang digunakan pada *traffic light* untuk memberikan rambu-rambu kepada para pengemudi adalah lampu merah, kuning, dan hijau (Sihombing, 2007). Gambar 2.2 di bawah memperlihatkan arti dari kode-kode warna yang digunakan pada *traffic light*.



**Gambar 2.2** Fase dan Warna *Traffict Light*

#### 4. Durasi Lampu Lalu Lintas

Durasi hidup lampu merah, kuning, dan hijau pada lampu lalu lintas di Indonesia tidak memiliki standar yang sama di setiap tempat, karena durasinya disesuaikan dengan kondisi lalu lintas di lokasi tersebut. Namun, ada beberapa pedoman umum yang sering diterapkan:

1. **Lampu Merah:** Biasanya memiliki durasi antara 60 hingga 90 detik. Di beberapa persimpangan dengan volume lalu lintas yang sangat tinggi, durasi ini bisa lebih panjang.
2. **Lampu Hijau:** Durasi lampu hijau biasanya berkisar antara 30 hingga 60 detik, tergantung pada arus kendaraan di persimpangan tersebut.

3. **Lampu Kuning:** Durasi lampu kuning umumnya lebih pendek, sekitar 3 hingga 5 detik. Lampu kuning ini berfungsi sebagai peringatan sebelum lampu berubah menjadi merah.

Setiap persimpangan bisa memiliki pengaturan yang berbeda, tergantung pada analisis lalu lintas di daerah tersebut.

## 5. *XAMPP*



**Gambar 2.3** Logo *XAMPP*

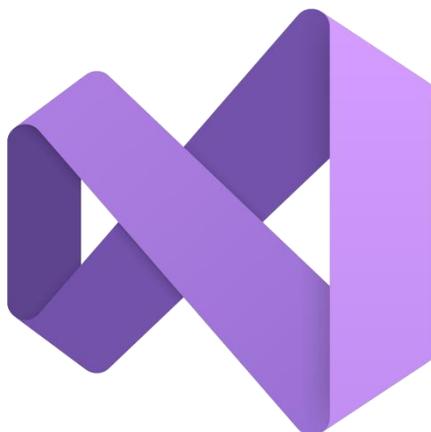
(Sumber : <https://www.exabytes.co.id/blog/apa-itu-xampp-adalah/>)

*XAMPP* adalah paket PHP dan MySQL berbasis open source, yang bisa dipergunakan pada tool pembantu pengembangan aplikasi berbasis PHP (Riyanto, 2010). Menurut Jubilee Enterprise (2018:3), bahwa *XAMPP* adalah server gratis yang paling sering diaplikasikan dalam keperluan belajar PHP secara mandiri, terutama bagi programmer pemula.

*XAMPP* merupakan perangkat lunak bebas, yang menyokong banyak sistem operasi yang merupakan kompilasi dari beberapa program. Fungsi dari *XAMPP* yaitu sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL* database, serta penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* adalah singkatan dari X (empat sistem operasi apapun), *Apache* (webserver), *MySQL* (database), *PHP*

(server side scripting) dan *Perl*. Program ini tersedia dalam GNU *General Public License* dan bebas yang merupakan webserver yang mudah diaplikasikan yang bisa mengoperasikan tampilan halaman web yang dinamis (Hendri dkk, 2019). Dengan menginstal *XAMPP* maka tidak perlu lagi melakukan instalasi serta melakukan konfigurasi web server *Apache*, *PHP*, dan *MySQL* secara manual. *XAMPP* akan otomatis menginstalasi dan mengkonfirmasi (Fraditya dkk, 2024).

## 6. *Visual Studio*



**Gambar 2.4** *Logo Visual Studio*

(Sumber : [https://en.wikipedia.org/wiki/Visual\\_Studio](https://en.wikipedia.org/wiki/Visual_Studio))

*Microsoft Visual Studio* adalah sebuah perangkat lunak lengkap yang diaplikasikan dalam melakukan peningkatan aplikasi, baik itu aplikasi bisnis, aplikasi personal, serta komponen aplikasi lainnya dalam bentuk aplikasi *console*, aplikasi *Windows*, dan aplikasi *Web*. Kompiler yang dimasukkan ke dalam paket *Visual Studio* yaitu *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*” (Caryl, 2020).

*Visual studio* atau *VB.NET* adalah suatu aplikasi yang telah berkembang yang

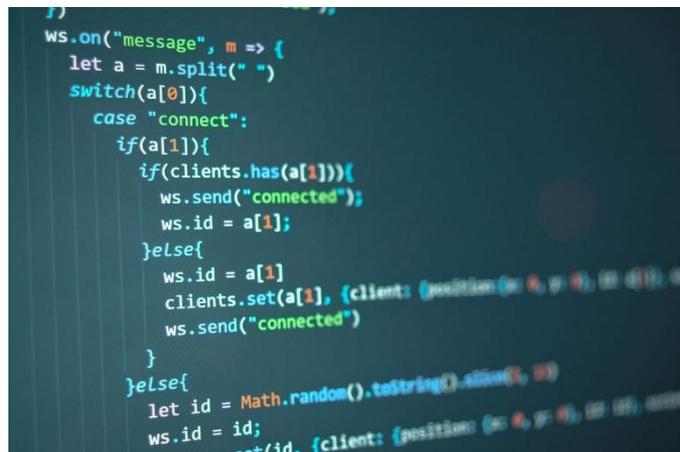
dapat digunakan pada platform NET sehingga aplikasi yang menggunakan Visual Studio atau VB.NET Ini dapat berjalan di komputer apapun, dan dari sever manapun dengan tipe apapun asalkan terinstal NET (Haris, 2021). Visual Studio merupakan alat pengembangan aplikasi yang komprehensif yang dapat diaplikasikan dalam merancang aplikasi komersial, aplikasi pribadi, serta komponen aplikasi lain dalam bentuk aplikasi konsol, aplikasi kendala atau aplikasi web (Gilang, 2022).

### **7. *Java Script***

JavaScript merupakan bahasa yang terdiri dari kumpulan kode yang berfungsi untuk dieksekusi pada dokumen HTML. Seiring berjalannya sejarah internet, JavaScript menjadi bahasa skrip pertama yang digunakan dalam pengembangan *web*. Selain itu, bahasa pemrograman ini memungkinkan penggunaan perintah peristiwa dalam meningkatkan kapabilitas bahasa HTML, memberikan fleksibilitas serta interaktivitas yang lebih besar dalam pengembangan aplikasi *web*. JavaScript adalah salah satu bahasa pemrograman atau dalam hal ini bahasa script populer yang sering diimplementasikan dalam membuat suatu halaman website yang berinteraksi dengan pengguna dan juga bisa merespon event yang terjadi pada halaman. JavaScript yaitu perekat antara halaman-halaman website (Chandra, 2023).

JavaScript sangat identik dengan browser, sehingga sangat terintegrasi dengan HTML. Ketika browser memuat suatu halaman, server akan mengirimkan konten utuh dari dokumen termasuk HTML dan statemen-statemen dari JavaScript. Konten HTML selanjutnya dibaca baris demi baris hingga sampai

pada pembacaan JavaScript, pada saat itu interpreter JavaScript akan mengambil alih. Ketika tag penutup JavaScript diperoleh, pemrosesan HTML kemudian berlanjut. Program JavaScript digunakan dalam melakukan deteksi dan bereaksi terhadap tindakan yang dilakukan oleh pengguna. JavaScript bisa digunakan dalam mengatur tampilan halaman. JavaScript juga digunakan dalam memvalidasi input dari pengguna ke dalam sebuah form sebelum pengiriman form ke server dilakukan. JavaScript berfungsi sebagai bahasa pemrograman yang mempunyai konstruksi-konstruksi dasar seperti variabel dan tipe data. JavaScript juga bisa menangani event yang diinisiasi oleh pengguna dan menetapkan timing. Kombinasi dari HTML, CSS, dan JavaScript akan menjadikan suatu website yang atraktif bagi pengguna (Putawa, 2022).



```

})
ws.on("message", m => {
  let a = m.split(" ")
  switch(a[0]){
    case "connect":
      if(a[1]){
        if(clients.has(a[1])){
          ws.send("connected");
          ws.id = a[1];
        }else{
          ws.id = a[1]
          clients.set(a[1], {client: {position: {x: 0, y: 0}, id: a[1]}})
          ws.send("connected")
        }
      }else{
        let id = Math.random().toString().slice(2, 8)
        ws.id = id;
        clients.set(id, {client: {position: {x: 0, y: 0}, id: id}})
      }
    }
  }
}

```

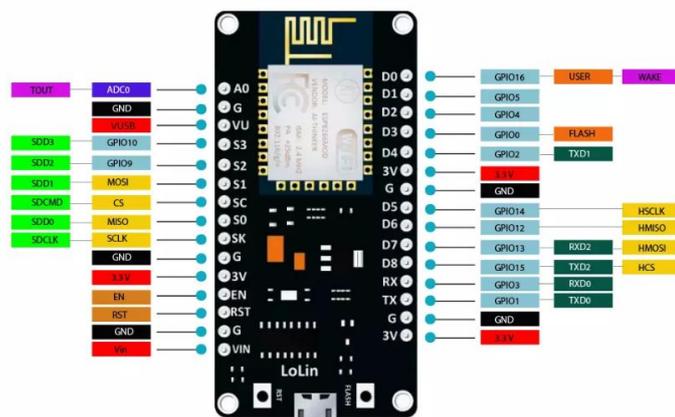
**Gambar 2.5** Bahasa Pemrograman JavaScript

(Sumber: <https://medium.com/@yunanadil/mengenaljavascript>)

## 8. Esp8266

ESP8266 adalah papan pengembangan mode ganda *WI-FI + Bluetooth* yang menggunakan antena dan papan PCB berbasis *chip ESP8266*. Modul ini bisa digunakan dalam segala keperluan, seperti pada CCTV, mengambil gambar dan

sebagainya. Mikrokontroler ini sudah memiliki modul *WiFi* yang terpasang di dalamnya, sehingga sangat membantu untuk mengembangkan arsitektur perangkat lunak *internet of things* (Adi dkk, 2023).



**Gambar 2.6** *Esp8266*

(Sumber: <https://student-activity.binus.ac.id/himtek/2022/07/27/Esp8266/>)

Terlihat pada gambar 2.6 merupakan *pin out* dari ESP8266. Pin ini bisa digunakan sebagai *input* atau *output* dalam menjalankan motor DC, perlengkapan penerangan, atau layar kristal cair. Mikrokontroler SoC (*device on chip*) terintegrasi, *ESP8266* dilengkapi dengan *WiFi* 802.11 *b/g/n*, Bluetooth 4.2, dan berbagai periferal. *ESP8266* merupakan *chip* yang berfungsi penuh yang dapat menggantikan Arduino karena memiliki prosesor, akses GPIO (*preferred motif enter Output*), dan fitur lainnya.

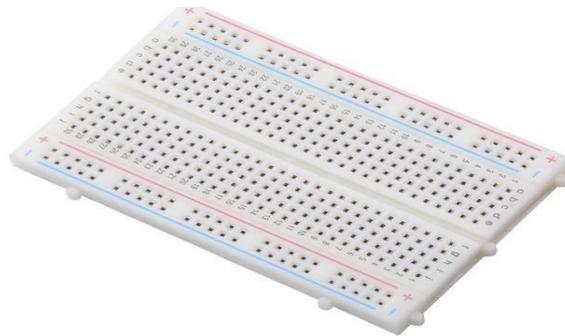
*ESP8266 Dev Kit V1* adalah sebuah *board* pengembangan (*development board*) yang didukung oleh mikrokontroler *Tensilica 32-bit Single-/Dual-core CPU Xtensa LX6* dengan kecepatan *clock 240 Mhz*. *Board* ini dilengkapi dengan 520 KiB SRAM dan 4 MB *flash memory* untuk menyimpan program dan data. *Board* ini juga memiliki 25 digital *input/output* (DIO) pins, 6 analog *input* (ADC) pin,

dan 2 analog *output* (DAC) pin yang dapat digunakan untuk berbagai keperluan seperti mengendalikan perangkat elektronik atau membaca sensor. Selain itu, *ESP8266 Dev Kit V1* juga dilengkapi dengan 3 UARTs, 2 SPIs, dan 3 I2Cs, yang memungkinkan *board* dalam berkomunikasi dengan perangkat lain secara serial atau menggunakan protokol komunikasi seperti SPI dan I2C. Tidak hanya itu, *board* dilengkapi dengan *Wi-Fi* yang mendukung standar IEEE 802.11 *b/g/n/e/*, sehingga *board* ini dapat terhubung ke jaringan *Wi-Fi* dan berkomunikasi dengan perangkat lain melalui jaringan tersebut. Dengan ukuran 51.5x29x5mm, *board* ini cukup kecil dan mudah untuk diintegrasikan ke dalam proyek-proyek yang lebih kompleks. Keseluruhan, *ESP8266 Dev Kit V1* adalah *board* pengembangan yang kuat dan serbaguna, yang cocok untuk berbagai aplikasi *Internet of Things (IoT)* atau proyek-proyek elektronika yang membutuhkan konektivitas *Wi-Fi* (Aulia, 2021).

### **9. Project Board (Bread Board)**

*Project Board* atau yang sering disebut sebagai *Bread Board* adalah dasar konstruksi sebuah sirkuit elektronik dan merupakan prototipe dari suatu rangkaian elektronik. Di zaman modern istilah ini sering digunakan untuk merujuk pada jenis tertentu dari papan tempat merangkai komponen, dimana papan ini tidak memerlukan proses menyolder (langsung tancap). Karena papan ini solderless atau tidak memerlukan solder sehingga dapat digunakan kembali, dan dengan demikian dapat digunakan untuk *prototipe* sementara serta membantu dalam melakukan eksperimen desain sirkuit dari elektronika. Berbagai sistem elektronik

dapat di *prototipe* dengan menggunakan *breadboard*, mulai dari sirkuit analog dan digital kecil sampai membuat unit pengolahan terpusat (CPU).



**Gambar 2.7** *Project Board*

(Sumber : <https://images.prismic.io/>)

2 Pasang jalur Atas dan bawah terhubung secara horizontal sampai ke bagian tengah dari *breadboard*. Biasanya jalur ini digunakan sebagai jalur *power* atau jalur sinyal yg umum digunakan seperti clock atau jalur komunikasi. 5 lubang komponen di tengah merupakan tempat merangkai komponen. Jalur ke 5 lobang ini terhubung vertikal sampai bagian tengah *breadboard*. Pembatas tengah *breadboard* biasanya digunakan sebagai tempat menancapkan komponen IC.

### **10. Kabel Jumper**

Kabel jumper merupakan suatu kabel-kabel pendek yang digunakan untuk menghubungkan antara komponen yang satu dengan komponen lainnya pada *breadboard* sehingga terdapat hubungan listrik dan terbentuk rangkaian elektronik.(Syahwil, 2018).

Kabel jumper disebut juga dengan istilah kabel *dupont*. Kabel *dupont* adalah kabel yang dikedua ujungnya dilengkapi dengan bagian yang memudahkan

untuk dihubungkan ke komponen lain. Ada beberapa jenis-jenis kabel jumper yang dibedakan berdasarkan konektor kabelnya, yaitu :

**a. *Male – Male***

Kabel jumper jenis ini digunakan untuk koneksi *male to male* pada kedua ujung kabelnya.



**Gambar 2.8** *Kabel Jumper Male - Male*

(Sumber : <https://images.prismic.io/>)

**b. Male – Female**

Kabel jumper jenis ini digunakan untuk koneksi *male to female* dengan salah satu ujung kabel dikoneksi male dan satu ujungnya lagi dengan koneksi *female*.



**Gambar 2.9** Kabel Jumper Male – Female

(Sumber : <https://images.prismic.io/>)

**c. Female – Female**

Kabel jumper jenis ini digunakan untuk koneksi *female to female* pada kedua ujung kabelnya.



**Gambar 2.10** Kabel Jumper Female – Female

(Sumber : <https://images.prismic.io/>)

Kabel *jumper* memiliki beberapa macam tipe seperti tipe sambungan *male to male*,

*female to male*, dan *male to female*. Dapat disesuaikan dengan kebutuhan namun untuk merangkai rangkaian transistor maka dibutuhkan *jumper* dengan tipe *male to male*, karena *header* yang digunakan pada *training kit transistor* ini merupakan *header* dengan tipe *female*.

### **11. Use Case**

*Use case diagram* adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* bekerja mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

*Use Case Diagram* merupakan pemodelan untuk melakukan (*behavior*) sistem informai yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa dan Shalahuddin, 2014).

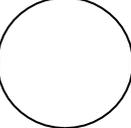
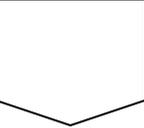
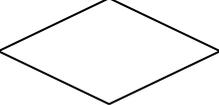
### **12. Bagan Alir Program**

Bagan alir program merupakan bagan yang menjelaskan rinci langkah- langkah dari proses program. Bagan alir program dibuat dari derifikasi bagan alir sistem. Menurut Jogiyanto (2005) “Bagan alir program (*program flowchart*) merupakan bagan alir yang mirip dengan suatu bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem”

Supardi (2013), “*flowchart* merupakan bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Pada waktu akan menggambar suatu bagan alir”. Adapun simbol-simbol dari

*Flowchart* adalah sebagai berikut :

**Tabel 2.1** *Simbol Flowchart*

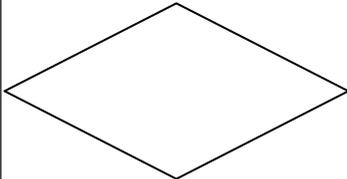
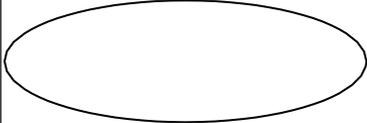
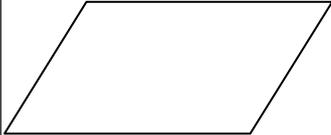
SIMBOL	KETERANGAN
	Simbol <i>input/output</i> digunakan untuk mewakili data <i>input/output</i> .
	Simbol proses, digunakan untuk mewakili suatu proses.
	Simbol proses terdefinisi, digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain.
	Simbol titik terminal, digunakan untuk awal dan akhir suatu proses.
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus di halaman yang masih sama.
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus di halaman yang berbeda.
	Simbol garis alir, menunjukkan arus dari proses.
	Simbol keputusan, digunakan untuk suatu penyelesaian kondisi didalam program.

### 13. *Entity Relationship Diagram*

*Entity Relationship Diagram (ERD)* merupakan komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi atribut-atribut yang merepresentasikan seluruh fakta dari dunia nyata yang kita tinjau. Simbol-simbol yang sering digunakan dalam *entity relationship diagram* dapat

dilihat pada tabel dibawah.

**Tabel 2.2** *Simbol Entity Relationship Diagram*

NOTASI	KETERANGAN
	<i>Entitas</i> , yaitu kumpulan dari objek yang diidentifikasi secara unik.
	Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antaralain : satu ke satu, satu ke banyak dan banyak ke banyak.
	Atribut, yaitu karakteristik dari <i>entity</i> atau relasi yang merupakan penjelasan detail tentang <i>entitas</i> .
	Garis, hubungan antara <i>entity</i> dengan atributnya dan himpunan <i>entitas</i> dengan himpunan relasi.
	<i>Input/ Output</i> data, yaitu proses <i>input/ output</i> data, parameter, informasi.

*Entity Relationship Diagram (ERD)* adalah suatu pemodelan konseptual yang didesain secara khusus untuk mengidentifikasi *entitas* yang menjelaskan data dan hubungan antar data. *ERD* untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga komponen yang digunakan, yaitu

a. *Entity*

*Entity* merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain.

b. Atribut

Setiap *entitas* pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari *entitas* tersebut.

c. Hubungan/ Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan *entitas* yang berbeda. Relasi dapat digambarkan sebagai berikut:

Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

i. Satu ke satu (*One to one*)

Hubungan relasi satu ke satu yaitu setiap *entitas* pada himpunan entitas A berhubungan paling banyak dengan satu *entitas* pada himpunan entitas B.

ii. Satu ke banyak (*One to many*)

Setiap entitas pada himpunan *entitas* A dapat berhubungan dengan banyak entitas pada himpunan *entitas* B, tetapi setiap *entitas* pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

iii. Banyak ke banyak (*Many to many*)

Setiap *entitas* pada himpunan *entitas* A dapat berhubungan dengan banyak *entitas* pada himpunan *entitas* B.

#### 14. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) menurut Munawar dalam buku “Pemodelan Visual dengan UML” (2005 : 17) menyatakan UML adalah salah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan *visual* yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Sedangkan menurut menurut Adi nugroho (2010:6), “*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek”. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Jurnal CCIT (2008:70) *Unified Modelling Language* (UML) didefinisikan oleh *Good Corporate* sebagai berikut :

*Metode Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan suatu sistem atau perangkat lunak yang akan dikembangkan, yaitu :

- a. Sesuatu (*things*)

Ada empat *things* dalam *Unified Modelling Language* (UML) :

- 1) *Structural things*, bagian yang relative statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.

- 2) *Behaviorial things*, bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang ruang dan waktu.
- 3) *Grouping things*, bagian pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta *subsitem-subsistem*.
- 4) *An notational things*, merupakan bagian yang meperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap element dalam model UML.

b. Relasi (*relationship*)

Untuk upaya tersebut UML menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya *statis* atau *dinamis*. Ke 9 jenis diagram dalam UML itu adalah :

- 1) Diagram kelas bersifat statis Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.
- 2) Diagram objek bersifat statis. Diagram ini memperlihatkan objek serta relasi-relasi antar objek. Diagram objek memperlihatkan instalasi statis dari segala sesuatu yang dijumpai pada diagram kelas.
- 3) *Use Case diagram* bersifat statis. Diagram ini memperlihatkan

himpunan *use case* dan aktor-aktor. Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

- 4) *Sequence diagram* bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
- 5) *Statechart diagram* bersifat dinamis. Diagram ini memperlihatkan *state-state* pada sistem memuat *state*, *transisi*, *event*, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
- 6) *Activity diagram* Bersifat dinamis. Diagram aktivitas ini adalah tipe khusus dari diagram *state* yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.
- 7) *Component diagram* bersifat *statis*. Komponen ini memperlihatkan organisasi serta kebergantungan sistem atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya.
- 8) *Deployment diagram* bersifat *statis*. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Diagram ini membuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram ini sangat berguna pada banyak mesin (Sholih, 2006).

Tabel 2.3 Table Use Case Diagram

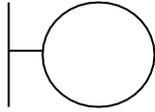
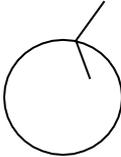
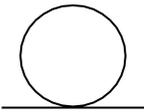
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya ( <i>sinergi</i> ).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2.4 *Symbol Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Tabel 2.5 *Symbol Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1	<i>Object</i>	<i>Objekct</i> <i>(Partisipan)</i>	<i>Object</i> atau biasa juga disebut partisipan merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara <i>horizontal</i> . Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama objek di dalamnya yang diawali dengan sebuah titik koma.
2		<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.
3		<i>Life line</i>	<i>Life line</i> mengidentifikasi keberadaan sebuah object dalam basis waktu. Notasi untuk <i>lifeline</i> adalah garis putus-putus <i>vertical</i> yang di tarik oleh sebuah <i>object</i>
4		<i>Collaborations</i>	<i>Activation</i> dinotasikan sebagai kotak segi empat yang digambar pada sebuah <i>lifeline</i> <i>action</i> mengidentifikasi sebuah <i>object</i> yang akan beraksi.

NO	GAMBAR	NAMA	KETERANGAN
5		<i>Boundary</i>	<i>Boundary</i> terletak diantara system dengan dunia di sekelilingnya. Semua <i>form</i> , laporan-laporan, antara muka ke perangkat keras seperti <i>printer</i> atau <i>scenner</i> dan antara muka ke system lainnya adalah termasuk dalam kategori.
6		<i>Control</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Entity</i>	<i>Entity</i> digunakan untuk menengani informasi yang akan disimpan secara permanen.

Metode pengujian yang dilakukan pada penelitian ini adalah metode pengujian langsung yaitu dengan menggunakan pengujian *White Box* dan *Black Box*. *White box testing* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara *procedural* untuk membagi pengujian ke dalam beberapa kasus pengujian. *Black Box* Digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang.

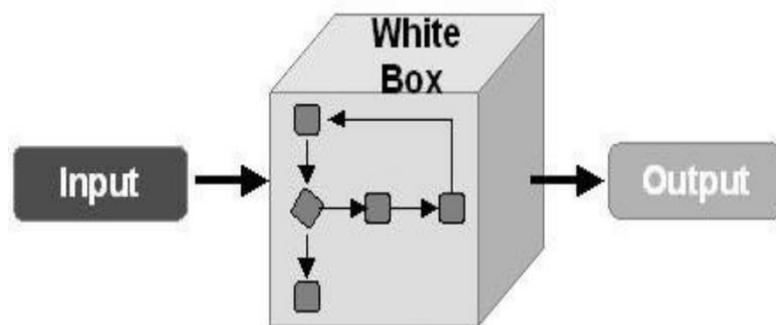
Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, *query*, atau jalur navigasi yang mewakili penggunaan sistem yang *tipical*, kritis atau abnormal. Pengujian harus mencakup *unit testing*, yang mengecek validasi dari *prosedur* dan fungsi-fungsi secara *independen* dari komponen sistem yang

lain. Kemudian modul *testing* harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit *testing*, yaitu :

### 15. *White Box Testing*

Uji coba *white box testing* merupakan metode perancangan *test case* yang menggunakan struktural untuk mendapatkan *test case*, *test* ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika), perangkat lunak di *test* dengan kondisi dan perulangan secara fisik.



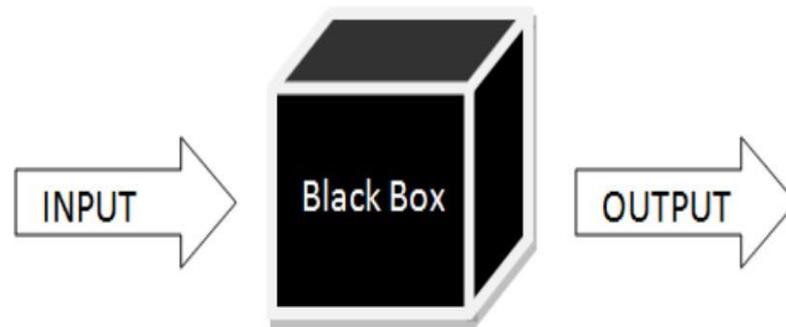
**Gambar 2.11** *Pengujian White Box*

(Sumber : <https://www.codester.com/>)

Contoh pengujian *white box testing* ini merupakan peringatan ketika *user* menginputkan *password user* yang salah, untuk kesalahan semacam ini akan memberikan suatu informasi kepada *user* mengenai kesalahan yang dilakukan.

## 16. *Black Box Testing*

*Black box testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitamnya. Sama seperti pengujian *black box testing*, mengevaluasi hanya dari tampilan luarnya (*interfacenya*), fungsionalitas tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui *input* dan *output*).



**Gambar 2.12** *Pengujian Black Box*

(Sumber : <https://www.codester.com/>)

Teknik yang digunakan dalam *Black Box Testing* antara lain:

- a. Digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak.
- b. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran (*output*) yang dihasilkan.
- c. Kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya.

### 17. *Sensor Arus ACS712*

Sensor arus ACS712 merupakan sensor yang menggunakan medan magnet yang dihasilkan dari aliran arus yang mengalir pada bagian kawat tembaga dari board sensor. Sensor ACS712 ini menggunakan rangkaian *low-offset linier hall*, sehingga pembacaan dari sensor ACS712 ini mempunyai tingkat akurasi yang tinggi. Sensor arus ACS712 memanfaatkan medan listrik yang selanjutnya diproses oleh IC ACS712 dan menghasilkan keluaran berupa tegangan proporsional (Andreas & Dimas, 2023).

ACS712 merupakan sensor arus efek Hall. Hall Effect Allegro ACS712 pada umumnya diaplikasikan untuk control motor, pendeteksi beban listrik, power switching, serta proteksi beban lebih. Pengaplikasian sensor ini yaitu bahwa arus baca mengalir melalui kabel tembaga di dalamnya, menciptakan medan magnet yang diambil oleh sirkuit Hall. Kekuatan pembacaan sensor dioptimalkan dengan memasang komponen antara konduktor penghasil medan magnet dan transduser efek Hall dalam jarak dekat (Rizal dkk, 2023).

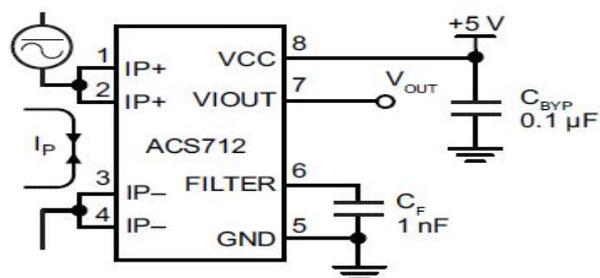


**Gambar 2.13** *Sensor Arus ACS712*

Penggunann sensor ini dipasang secara seri serta mampu mengukur arus positif dan negatif dari -5 A hingga 5 A, dan memerlukan inputan daya sebesar 5 V.

Untuk pembacaan nilai tengah atau 0 A maka harus di kalibrasi atau di set dari setengah nilai  $V_{cc} = 5\text{ V}$  yakni 2.5V. Modul ACS712 memiliki tingkat sensitifitas sebesar 400 mV/A (Makruf dkk, 2020).

### Typical Application



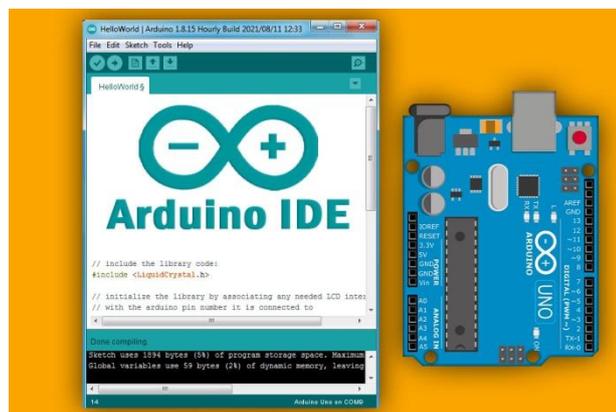
**Gambar 2.14** Rangkaian Sensor Arus ACS712

### 18. Arduino IDE

Arduino IDE merupakan software yang digunakan dalam membuat sketch pemrograman atau dengan kata lain arduino IDE sebagai media dalam pemrograman pada board yang akan diprogram. Arduino IDE ini berfungsi untuk mengedit, membuat, meng-upload ke board yang ditentukan, serta meng-coding program tertentu. Arduino IDE dibuat dari bahasa pemrograman JAVA, yang dilengkapi dengan library C/C++(wiring), yang membuat operasi *input/output* lebih mudah (Kamal dkk, 2023).

Arduino IDE merupakan bagian software opensource yang memungkinkan seseorang dalam memprogram bahasa Arduino dalam bahasa C. Arduino IDE memungkinkan kita dalam menulis sebuah program secara step by step selanjutnya instruksi tersebut diupload ke papan Arduino (Adriansyah & Oka, 2013).

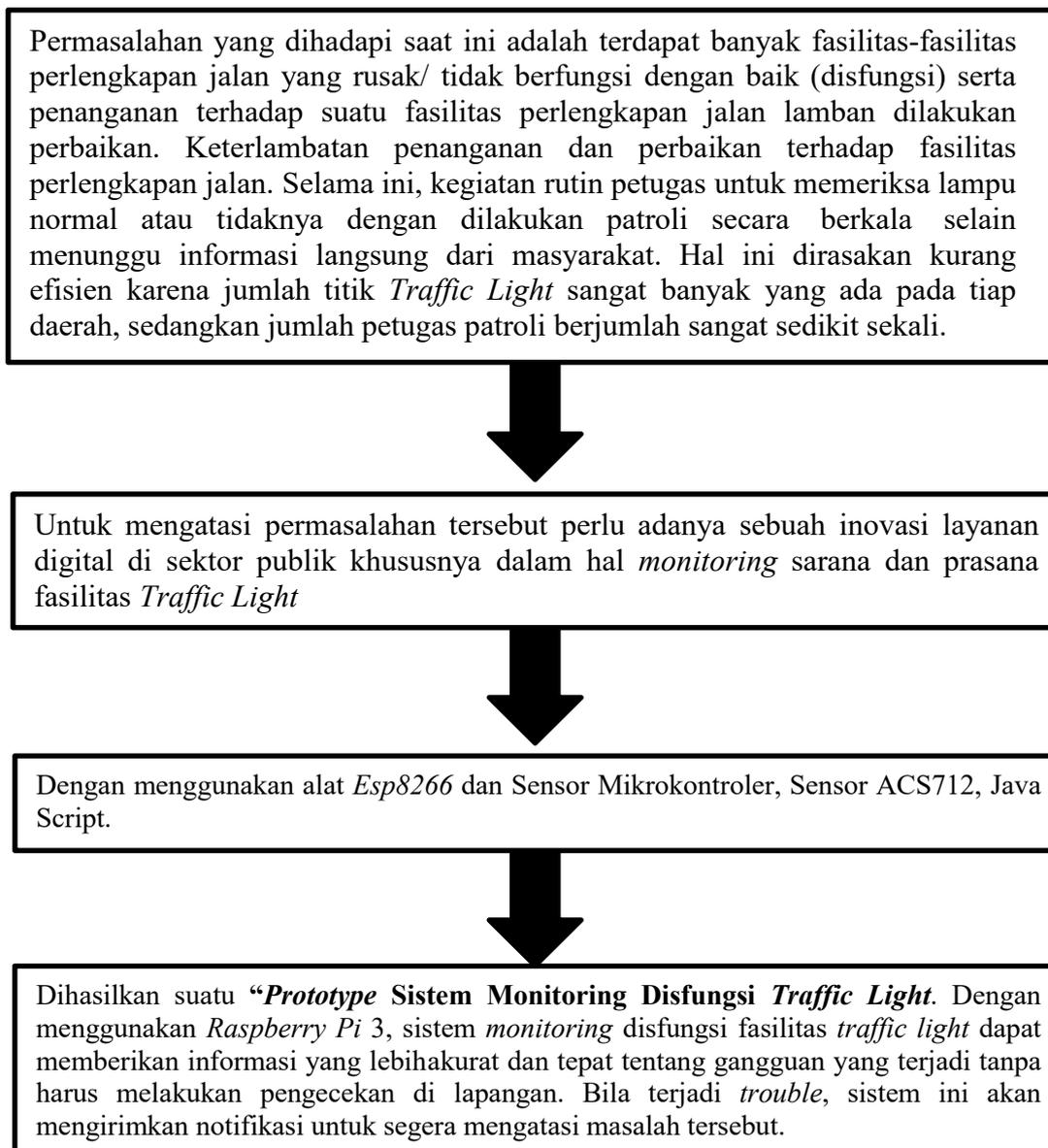
Arduino IDE adalah kependekan dari Integrated Development Environment, atau secara bahasa mudahnya merupakan lingkungan yang terintegrasi yang diimplementasikan dalam melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman dalam melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) telah dilakukan perubahan dalam memudahkan pemula untuk melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino sudah ditanamkan suatu program bernama *Bootlader* yang berperan sebagai penengah antara compiler Arduino dengan mikrokontroler. Arduino IDE ini dikembangkan dari *software Processing* yang dirombak menjadi Arduino IDE khusus dalam pemrograman dengan Arduino (Sendi dkk, 2024).



**Gambar 2.15** *Software Arduino IDE*

### C. Kerangka Pikir

Untuk memperjelas suatu kerangka berpikir yang akan dituliskan dalam penelitian akhir ini. Maka berikut digambarkan kerangka fikir tersebut.



Gambar 2.16 Kerangka Berpikir

## **BAB III**

### **METODE PENELITIAN**

#### **A. Lokasi dan Waktu Penelitian**

Lokasi penelitian dikondisikan (di Dinas Perhubungan Kabupaten Sidenreng Rappang) dan waktu penelitian kurang lebih 1 bulan.

#### **B. Jenis Penelitian**

Jenis penelitian yang digunakan adalah penelitian kuantitatif dengan metode eksperimental. Jenis penelitian yang dilakukan yaitu penelitian lapangan dengan metode *survei* secara langsung ke titik lokasi penelitian, Tujuan dari metode ini untuk mendapatkan rangkaian alat yang akan dibangun, memodifikasi rangkaian-rangkaian elektronika untuk tugas akhir serta melakukan pengujian terhadap rangkaian simulasi yang telah dibuat.

#### **C. Metode Penelitian**

Data dikumpulkan melalui tiga metode: peninjauan literatur, wawancara, dan observasi, serta kajian pustaka, untuk membuat kamus digital yang lebih terarah dan terfokus pada data dan informasi yang akurat tentang subjek yang dibahas.

##### *Penelitian Kepustakaan (Literature rievew)*

Penulis menggunakan teori, literatur, jurnal, dan sumber kepustakaan lainnya untuk melakukan penelitian kepustakaan untuk menemukan informasi yang relevan dengan judul tugas akhir. Penulis berharap program aplikasi yang dibuat sesuai dengan definisi-definisi yang ada. Salah satu cara pengumpulan data adalah

studi literatur, yang melibatkan membaca literatur yang berkaitan dengan pemikiran penulis.

#### **D. Metode Pengumpulan Data**

Tahap ini peneliti melakukan wawancara dengan pihak yang terkait dengan penelitian, serta melakukan pencatatan dan pengamatan (*observation*) langsung di tempat penelitian. Adapun uraian dari tahapan tersebut adalah sebagai berikut :

##### 1. Persiapan Penelitian

Tahap ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta alat yang digunakan selama penelitian.

##### 2. Studi *Literature*

Tahap ini peneliti memperoleh semua informasi seperti mencari referensi dari jurnal, buku, paper internasional, *youtube* dan referensi lainnya yang berhubungan dengan penelitian ini. Referensi yang digunakan adalah referensi yang berhubungan dengan penelitian.

##### 3. Pengumpulan Data

Tahap ini peneliti melakukan pengumpulan data yang akan digunakan pada penelitian ini yaitu data-data yang berbentuk fisik.

##### 4. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

##### 5. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

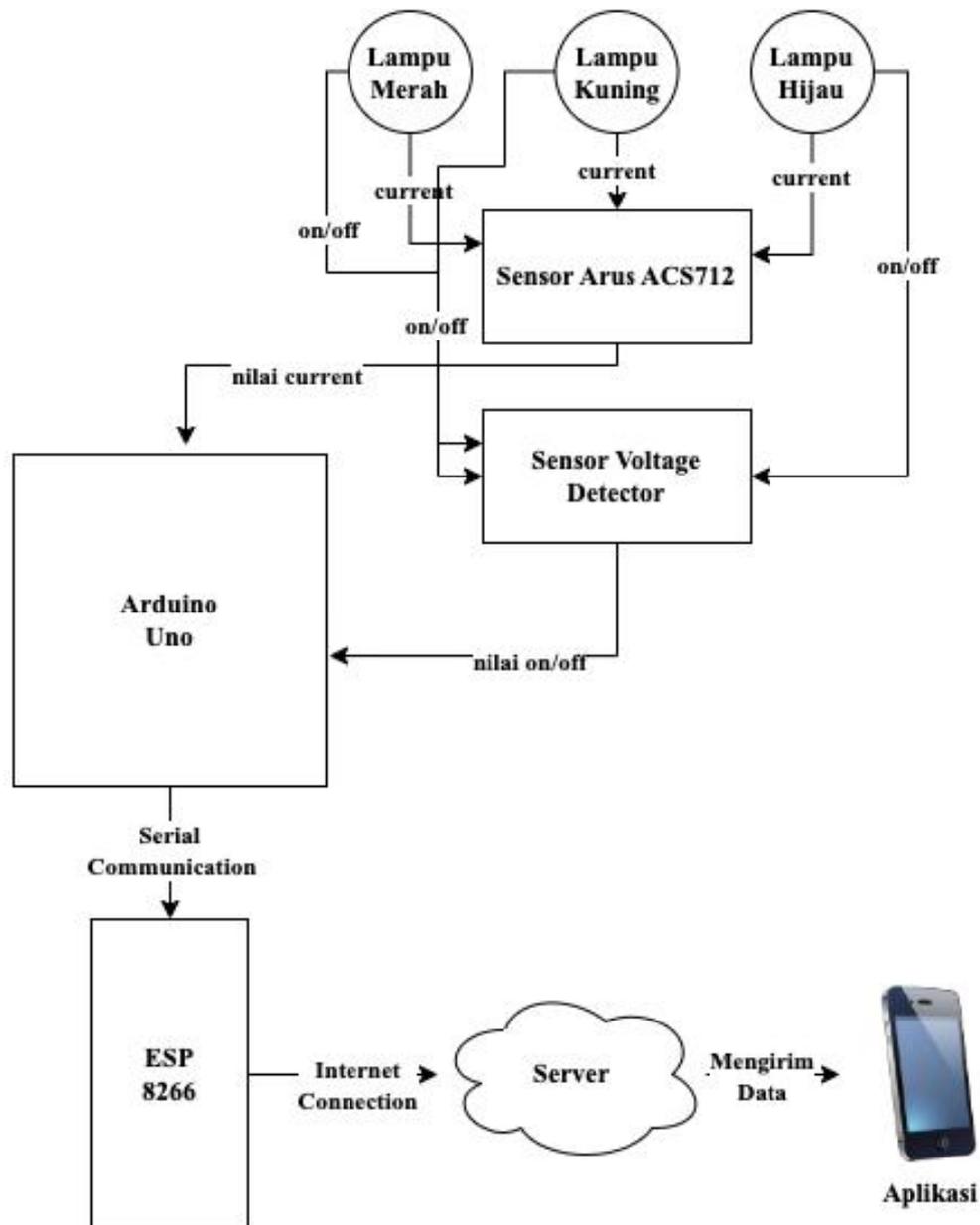
### E. Alat dan Bahan

Dalam melakukan penelitian, maka diperlukan alat dan bahan penelitian yang mendukung kegiatan penelitian tersebut. Alat dan bahan yang diperlukan antara lain:

**Table 3.1** Alat dan Bahan

ALAT	BAHAN
Laptop <i>Asus</i>	Perangkat lunak ( <i>Software</i> )
<i>Firebase</i>	<i>Buld Lampu</i>
<i>ESP8266</i>	<i>Project Board</i>
<i>Node JS</i>	<i>Kabel Jumper</i>
<i>Flutter</i>	<i>Sensor Voltage Detector</i>
<i>Tolls Visual Studio Code</i>	<i>Sensor Arus ACS712</i>
<i>Uml</i>	

## F. Sistem Yang Diusulkan

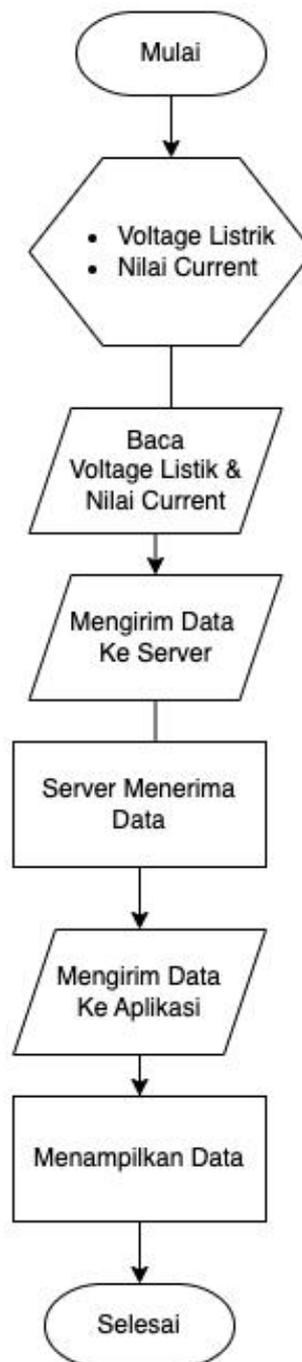


**Gambar 3.1** Sistem Yang Diusulkan

Diatas merupakan sistem yang diusulkan oleh peneliti, dimana tiap lampu merah, kuning dan hijau terhubung kepada sensor *voltage detector* dan sensor arus *acs712* yang masing-masing bekerja untuk mendapatkan nilai *on/off voltage* dan nilai *current*. Nilai *on/off* dan *current* kemudiannya dikirim ke *arduino uno*.

Alasan menggunakan *arduino uno* dikarenakan memiliki lebih banyak pin analog yang mana dibutuhkan oleh sensor arus *acs712* sebanyak tiga. Kemudian nilai *current* dan *on/off* yang didapat dikirim ke *esp8266* melalui *serial communcation*. Alasan menggunakan *esp8266* karena memiliki *built in wifi* yang memungkinkan untuk terhubung ke *internet* melalui *wifi*. Kemudian nilai *current* dan *on/off* dikirimkan ke *server* yang kemudiannya mengirimkan data tersebut ke aplikasi untuk ditampilkan.

### G. Flowchart Aplikasi



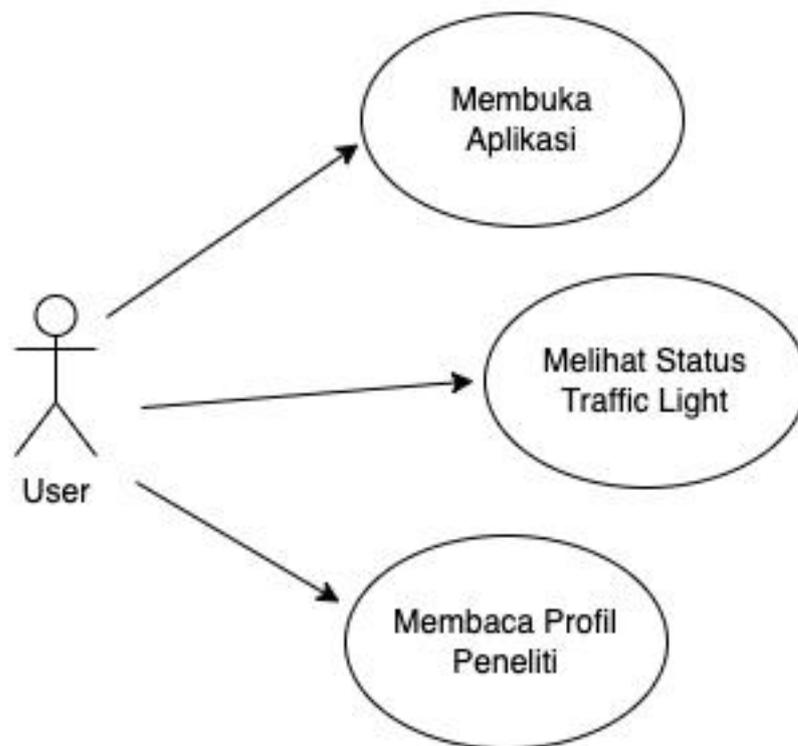
**Gambar 3.2** Flowchart Aplikasi

## BAB IV

### HASIL DAN PEMBAHASAN

#### A. Use Case Diagram

*Use case* diagram merupakan penggambaran interaksi antara aktor dan proses atau sistem yang dibuat. *Use case* dan *actor* menggambarkan ruang lingkup sistem yang sedang dibangun, sedangkan *use case* mencakup semua hal yang ada pada sistem; *actor* dapat mencakup seseorang atau apa pun yang terkait dengan sistem



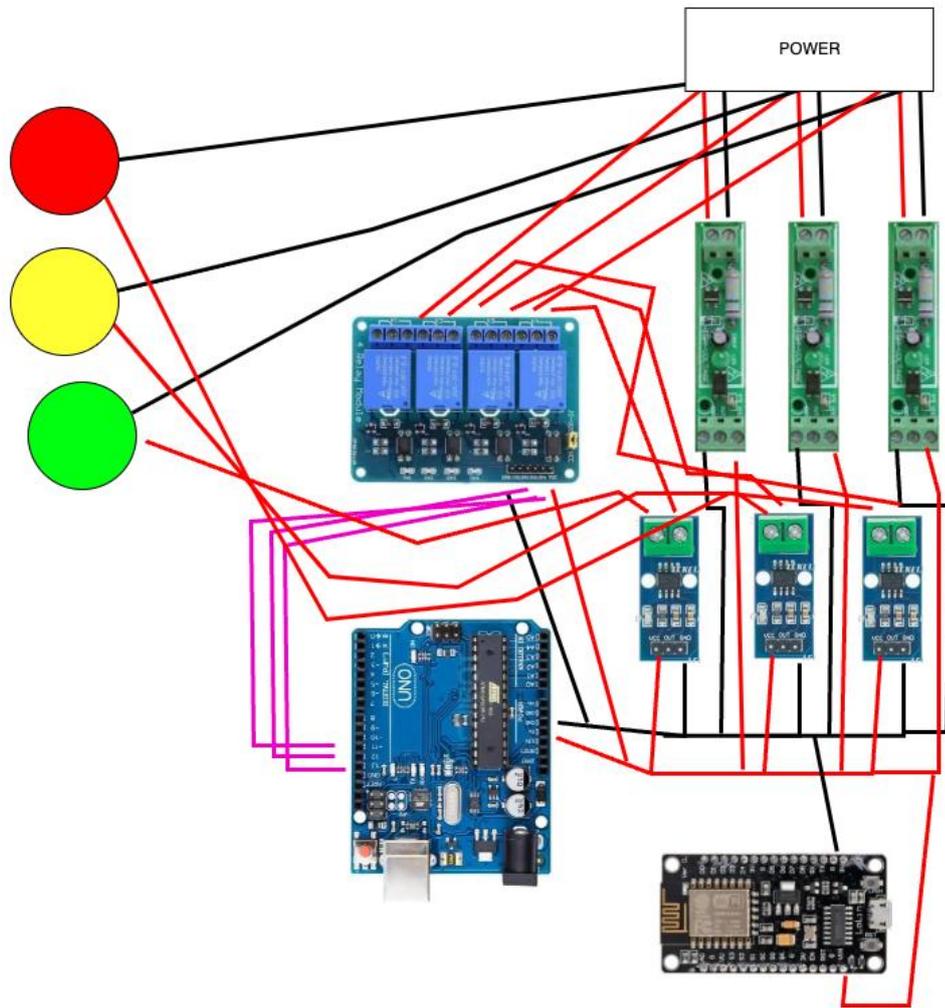
**Gambar 4.1** *Use Case Diagram*

**Tabel 4.1** *Penjelasan Use Case*

<i>Use Case</i>	<i>Penjelasan Use Case</i>
-----------------	----------------------------

<i>Use Case</i>	<b>Penjelasan Use Case</b>
<i>User</i>	<i>Use case</i> ini menjelaskan bahwa <i>user</i> merupakan aktor dari sistem yang dibuat
Membuka Aplikasi	<i>Use case</i> ini menjelaskan bahwa <i>user</i> harus membuka aplikasi untuk
Melihat Status <i>Traffic Light</i>	<i>Use case</i> ini menjelaskan <i>user</i> dapat melihat status <i>traffic light</i>
Membaca Profil Peneliti	<i>Use case</i> ini menjelaskan <i>user</i> dapat membaca profil peneliti

## **B. Desain Konstruksi**



**Gambar 4.2** Desain Konstruksi Alat

Diatas merupakan konstruksi alat dimana terdapat tiga buah sensor arus *acs712* dan sensor *voltage detector* untuk masing-masing lampu, semua sensor terhubung ke *arduino uno* dan sensor akan mengirim nilai ke *arduino uno* yang kemudian diteruskan ke *esp8266* untuk mengirimkan data ke *server* melalui *internet*. Dibawah merupakan tabel koneksi kabel antara *controller* dan *sensor*.

**Tabel 4.2** Koneksi Uno ke Relay

<i>Arduino Uno</i>	<i>Relay</i>
<i>Gnd</i>	<i>Gnd</i>
<i>5V</i>	<i>Vcc</i>
<i>Pin 13</i>	<i>In 1</i>
<i>Pin 12</i>	<i>In 2</i>
<i>Pin 11</i>	<i>In 1</i>

**Tabel 4.3** Koneksi Uno Ke ACS712

<i>Arduino Uno</i>	<i>ACS712</i>
<i>Gnd</i>	<i>Gnd</i>
<i>5V</i>	<i>Vcc</i>
<i>Pin 7</i>	<i>Analog 0</i>
<i>Pin 6</i>	<i>Analog 1</i>
<i>Pin 5</i>	<i>Analog 2</i>

**Tabel 4.4** Koneksi Uno Ke Voltage Detector

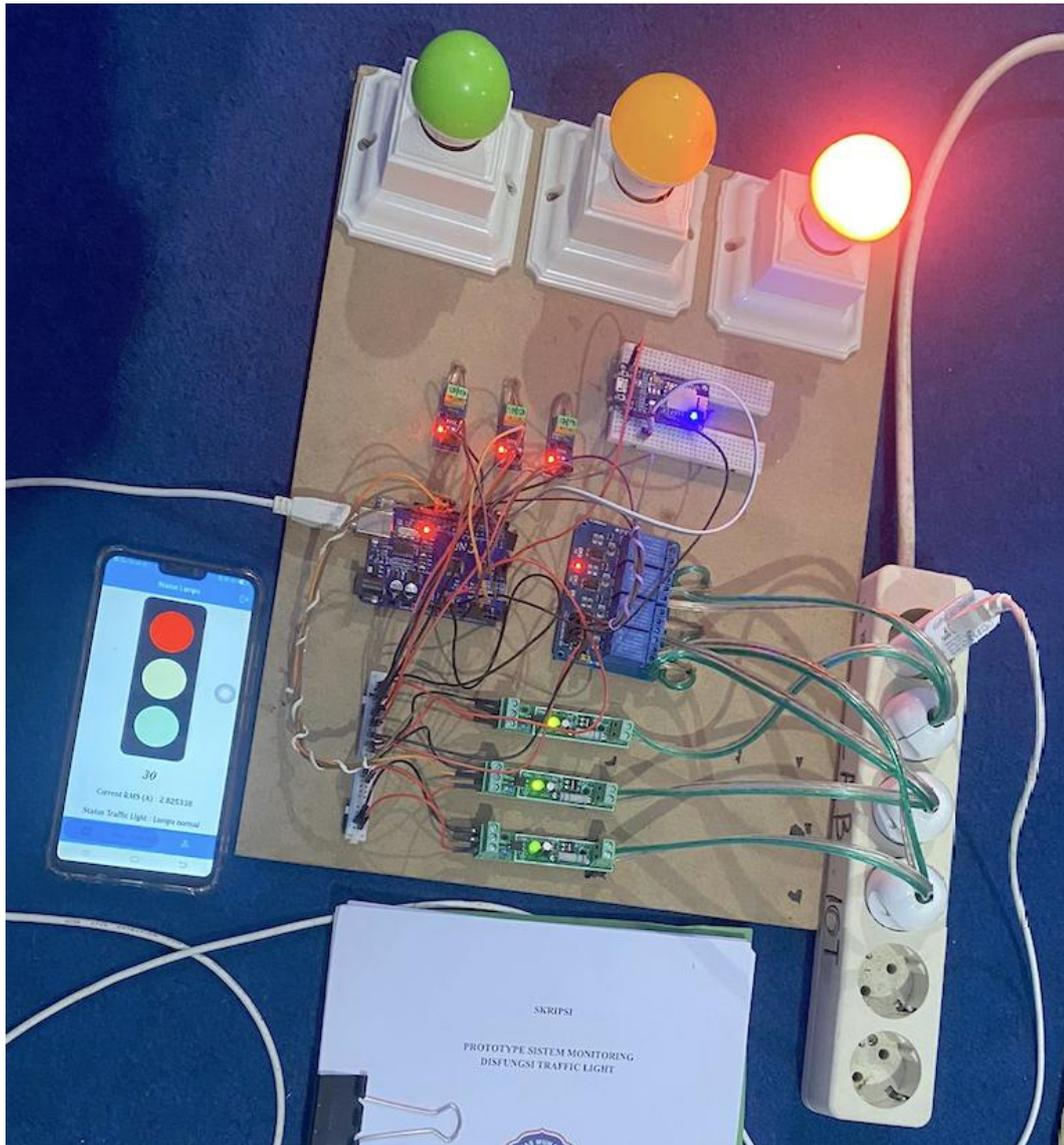
<i>Arduino Uno</i>	<i>Voltage Detector</i>
<i>Gnd</i>	<i>Gnd</i>
<i>5V</i>	<i>Vcc</i>
<i>Pin 7</i>	<i>Out 1</i>
<i>Pin 6</i>	<i>Out 2</i>
<i>Pin 5</i>	<i>Out 3</i>

**Tabel 4.5** Koneksi Uno Ke ESP8266

<i>Arduino Uno</i>	<i>ESP8266</i>
<i>Gnd</i>	<i>Gnd</i>
<i>5V</i>	<i>Vin</i>
<i>Pin 2</i>	<i>D7</i>
<i>Pin 3</i>	<i>D6</i>

### C. Tampilan Alat

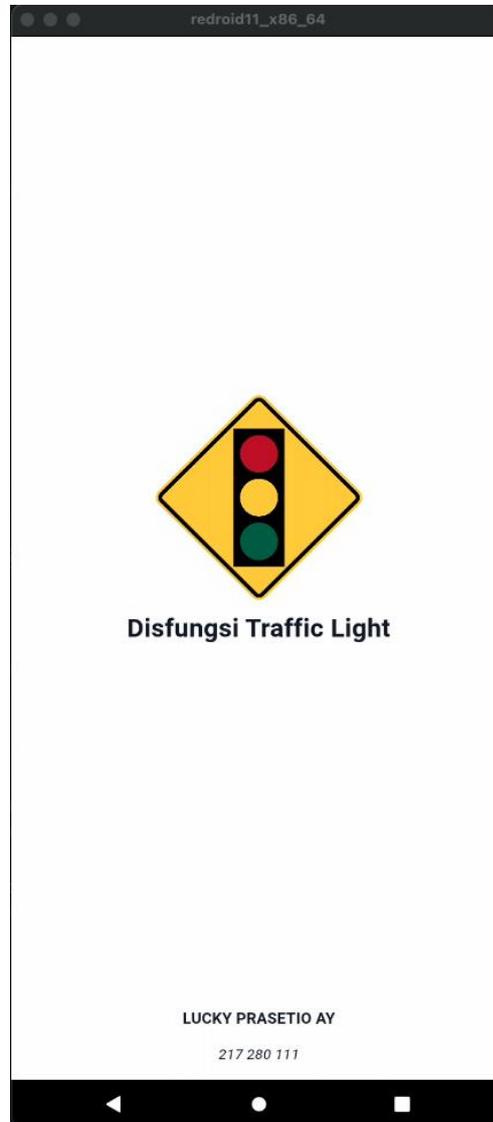
Dibawah merupakan tampilan alat dan aplikasi yang selesai dirancang dan dibuat oleh penulis :



**Gambar 4.3** Tampilan Alat

## D. Tampilan Aplikasi

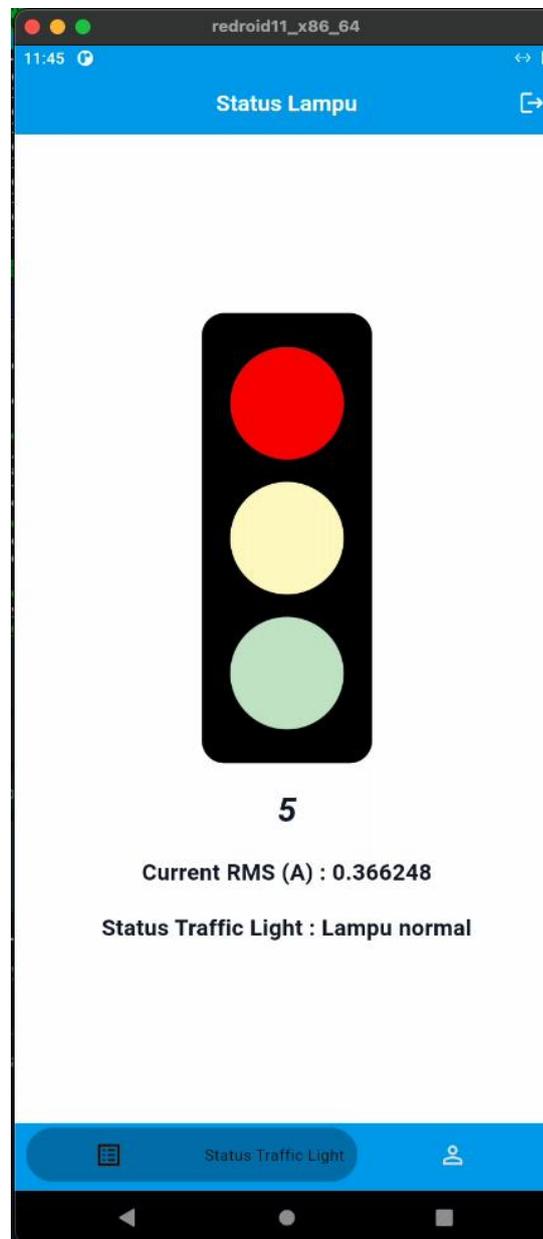
### 1. Tampilan *Splashscreen*



**Gambar 4.4** Tampilan *Splashscreen*

Diatas merupakan tampilan *splashscreen* aplikasi, dimana pada tampilan ini menampilkan logo aplikasi, judul aplikasi serta nama peneliti

## 2. Tampilan Status *Traffic Light*

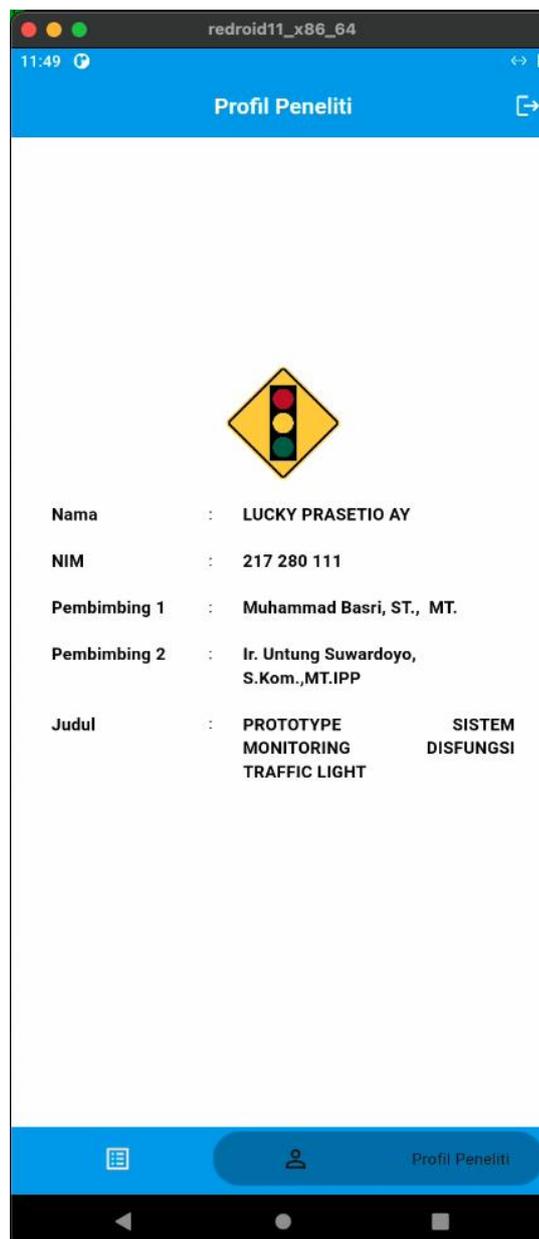


**Gambar 4.5** Tampilan Status *Traffic Light*

Diatas merupakan tampilan status *traffic light*, disini aplikasi menampilkan data yang telah diterima oleh server sebelumnya melalui internet. Aplikasi menampilkan waktu traffic light, *current RMS* dan status

*traffic light* yang mana merupakan percabangan dari *current* dan nilai *sensor voltage*.

### 3. Tampilan Profil Peneliti



**Gambar 4.6** Tampilan Profil Peneliti

Diatas merupakan tampilan profil peneliti, disini user dapat mengetahui nama peneliti, judul peneliti dan juga pembimbing peneliti

## E. Syntax Alat dan Aplikasi

### 1. Arduino Uno

```

#include <Arduino.h>
#include <SoftwareSerial.h>

SoftwareSerial toEsp(2, 3); // RX, TX

// green light on relay
const int relayPin_green = 5;
const int checkPLN_green = 11;
const int analogIn_green = A2;

// yellow light on relay
const int relayPin_yellow = 6;
const int checkPLN_yellow = 12;
const int analogIn_yellow = A1;

// red light on relay
const int relayPin_red = 7;
const int checkPLN_red = 13;
const int analogIn_red = A0;

const int mVperAmp = 66; // 66 mV per Amp for ACS712-30A
const int ACSoffset = 2500; // Offset for zero current (mV)

float amplitude_current = 0.0; // Amplitude current
float effective_value = 0.0; // Effective current (RMS)

int getMaxValue(int duration, int analogPin)
{
    int sensorValue; // Value read from the sensor
    int sensorMax = 0;
    uint32_t start_time = millis();

    while ((millis() - start_time) < duration)
    {
        sensorValue = analogRead(analogPin);
        if (sensorValue > sensorMax)
        {

```

```

    sensorMax = sensorValue; // Record the maximum sensor value
  }
}
return sensorMax;
}

void controlRelay(int relayPin, int checkPLN, int analogIn)
{
  digitalWrite(relayPin, LOW);
  String relayStat;
  if (relayPin == relayPin_green)
  {
    relayStat = "Green";
    Serial.println("Green ON");
    digitalWrite(relayPin_yellow, HIGH);
    digitalWrite(relayPin_red, HIGH);
  }
  else if (relayPin == relayPin_yellow)
  {
    relayStat = "Yellow";
    Serial.println("Yellow ON");
    digitalWrite(relayPin_green, HIGH);
    digitalWrite(relayPin_red, HIGH);
  }
  else if (relayPin == relayPin_red)
  {
    relayStat = "Red";
    Serial.println("Red ON");
    digitalWrite(relayPin_green, HIGH);
    digitalWrite(relayPin_yellow, HIGH);
  }
  for (int i = 0; i < 10; i++)
  {
    int ii = 10 - i;
    String pln;
    byte x = digitalRead(checkPLN);
    Serial.println(relayStat);
    if (x == 0)
    {
      Serial.println("PLN ON");
    }
  }
}

```

```

    pln = "PLN ON";
}
else
{
    Serial.println("PLN OFF");
    pln = "PLN OFF";
}

int sensorMax = getMaxValue(1000, analogIn); // Get max value over 1 second

float voltage = (sensorMax / 1024.0) * 5000.0; // Convert to millivolts
amplitude_current = (voltage - ACSoffset) / mVperAmp;
effective_value = amplitude_current / 1.414; // RMS calculation

Serial.print("Current Max (A) : ");
Serial.println(amplitude_current, 6);
Serial.print("Current RMS (A) : ");
Serial.println(effective_value, 6);
Serial.println();

    toEsp.print(relayStat + "," + pln + "," + String(amplitude_current, 6) + "," +
String(effective_value, 6) + "," + String(voltage, 6) + "," + ii + "\n");
    // delay(1000); // Delay between readings
}
}

void setup()
{
    pinMode(relayPin_green, OUTPUT);
    pinMode(analogIn_green, INPUT);
    pinMode(checkPLN_green, INPUT);

    pinMode(relayPin_yellow, OUTPUT);
    pinMode(analogIn_yellow, INPUT);
    pinMode(checkPLN_yellow, INPUT);

    pinMode(relayPin_red, OUTPUT);
    pinMode(analogIn_red, INPUT);
    pinMode(checkPLN_red, INPUT);
    Serial.begin(115200);

```

```

toEsp.begin(9600);
digitalWrite(relayPin_green, HIGH); // Ensure relay starts OFF
digitalWrite(relayPin_yellow, HIGH); // Ensure relay starts OFF
digitalWrite(relayPin_red, HIGH); // Ensure relay starts OFF
}

void loop()
{
  controlRelay(relayPin_green, checkPLN_green, analogIn_green);
  controlRelay(relayPin_yellow, checkPLN_yellow, analogIn_yellow);
  controlRelay(relayPin_red, checkPLN_red, analogIn_red);
}

```

**Gambar 4.7** Syntax Arduino Uno

Diatas merupakan syntax *arduino uno*. Dimana pada *syntax* ini *sensor acs712* dan *sensor voltage detection* akan memberikan nilai masing-masing ke *arduino uno* untuk tindakan lanjutan. *Arduino uno* juga terhubung ke *esp8266* melalui *serial communication*.

## 2. ESP8266

```

#include <Arduino.h>
#include "SoftwareSerial.h"
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
const int ledPin = 2; // GPIO2 is typically the onboard LED pin on ESP8266

const char *ssid = "KARAN";
const char *password = "12345679";

WiFiClient client;

SoftwareSerial fromUno(13, 15); // RX, TX

void setup()
{
  Serial.begin(9600);

```

```
fromUno.begin(9600);

// Initialize the LED pin as an output
pinMode(ledPin, OUTPUT);
digitalWrite(ledPin, HIGH); // Turn off LED initially

Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
digitalWrite(ledPin, LOW); // Turn on LED to indicate Wi-Fi connected
}

void loop()
{
  if (WiFi.status() == WL_CONNECTED)
  {
    digitalWrite(ledPin, LOW); // Turn on LED to indicate Wi-Fi connected
    String data = "";
    while (fromUno.available() > 0)
    {
      data += char(fromUno.read());
      // delay(1);
    }
    // Serial.println(data);

    // Ensure the data string is properly terminated
```

```

data.trim();

// Split the data string by commas
int commaIndex1 = data.indexOf(',');
String light = data.substring(0, commaIndex1);
int commaIndex2 = data.indexOf(',', commaIndex1 + 1);
String pln = data.substring(commaIndex1 + 1, commaIndex2);
int commaIndex3 = data.indexOf(',', commaIndex2 + 1);
String value1 = data.substring(commaIndex2 + 1, commaIndex3);
int commaIndex4 = data.indexOf(',', commaIndex3 + 1);
int commaIndex5 = data.indexOf(',', commaIndex4 + 1);
String value2 = data.substring(commaIndex3 + 1, commaIndex4);
String value3 = data.substring(commaIndex4 + 1, commaIndex5);
String value4 = data.substring(commaIndex5 + 1);

// Ensure no extra data is included
value4 = value4.substring(0, value4.indexOf('\n') >= 0 ? value4.indexOf('\n') :
value4.length());

// Construct the JSON payload
String jsonSend = "{\"light\":\"" + light + "\",\"pln\":\"" + pln + "\",\"value1\":\"" +
value1 + "\",\"value2\":\"" + value2 + "\",\"value3\":\"" + value3 + "\",\"no\":\"" +
value4 + "\"}";

Serial.println(jsonSend);

HTTPClient http;
http.begin(client, "http://192.168.20.45:3004");
http.addHeader("Content-Type", "application/json");
int httpResponseCode = http.POST(jsonSend);
if (httpResponseCode > 0)
{
  Serial.print("HTTP Response code: ");
  Serial.println(httpResponseCode);
}
else
{
  Serial.print("Error code: ");
  Serial.println(httpResponseCode);
}

```

```

    http.end();
    delay(1000);
}
else
{
    Serial.println("WiFi is disconnected, attempting to reconnect");
    digitalWrite(ledPin, HIGH);
    WiFi.begin(ssid, password);

    // Wait for connection
    int timeout = 0;
    while (WiFi.status() != WL_CONNECTED && timeout < 20)
    { // wait for 10 seconds max
        delay(500);
        Serial.print(".");
        timeout++;
    }

    if (WiFi.status() == WL_CONNECTED)
    {
        Serial.println("");
        Serial.println("Reconnected to WiFi");
        Serial.println("IP address: ");
        Serial.println(WiFi.localIP());
        digitalWrite(ledPin, LOW); // Turn on LED to indicate Wi-Fi connected
    }
    else
    {
        Serial.println("");
        Serial.println("Failed to reconnect to WiFi");
    }
}
}
}

```

**Gambar 4.8** Syntax ESP8266

Diatas merupakan *syntax esp8266* yang mana terhubung dengan *arduino uno* melalui *serial communication*. *ESP8266* akan melakukan koneksi *wifi* agar dapat terkoneksi ke internet untuk mengirim data ke *server*.

### 3. Node.js

```
const express = require('express');
const http = require('http');
const cors = require('cors');
const dotenv = require('dotenv');
const socket = require('./socket');
const app = express();
const server = http.createServer(app);
const io = socket.init(server);
const iosend = socket.getIO();

dotenv.config();

app.use(express.json());
app.use(express.urlencoded({ extended: true }));
// app.use(fileUpload());
app.options('*', cors());
app.use(cors());

function isValidDate(dateString) {
  // Check if the string matches the format YYYY-MM-DD using a regular expression
  const regex = /^\\d{4}-\\d{2}-\\d{2}$/;
  if (!regex.test(dateString)) {
    return false;
  }

  // Parse the string to a Date object
  const date = new Date(dateString);

  // Check if the date is valid by comparing it with the components of the original
  string
  const timestamp = date.getTime();
  if (isNaN(timestamp)) {
    return false;
  }

  // Further check to ensure that the date components match the input string
  const [year, month, day] = dateString.split('-').map(Number);
```

```

    if (date.getUTCFullYear() !== year || date.getUTCMonth() + 1 !== month ||
date.getUTCDate() !== day) {
        return false;
    }

    return true;
}

function timeStringToDate(timeStr) {
    const [hours, minutes, seconds] = timeStr.split(':').map(Number);
    const date = new Date();
    date.setHours(hours, minutes, seconds, 0);
    return date;
}

// Handle POST requests to '/post'
app.post('/', async (req, res) => {
    // Access the POST data sent in the request body
    try {
        var postData = req.body;
        // console.log(postData);
        const {no, light, value2, pln} = postData;
        // rms = value2
        iosend.emit('datanya', {
            no : no,
            light : light,
            pln : pln,
            rms : value2});

        return res.status(200).send({ message: 'OK data' });

    } catch (error) {
        console.error(error);
        return res.status(500).send({ message: 'Internal server error' });
    }
});

app.get('/', async (req, res) => {
    return res.status(200).send({ message: 'This is a GET request' });
});

```

```

})

// app error handler
app.use((err, req, res, next) => {
  console.log(err);
  res.status(500).send('Something broke!');
});

io.on('connection', (socket) => {
  let userID = socket.id;
  console.log('A user connected: ' + userID);

  socket.on('scan_dia', (data) => {
    console.log('Received scan_dia event: ' + data);
  });

  socket.on('disconnect', () => {
    console.log('User disconnected: ' + userID);
  });
});

module.exports = {
  app,
  server,
  io
};

const port = process.env.PORT || 3004;

// Start the server
server.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});

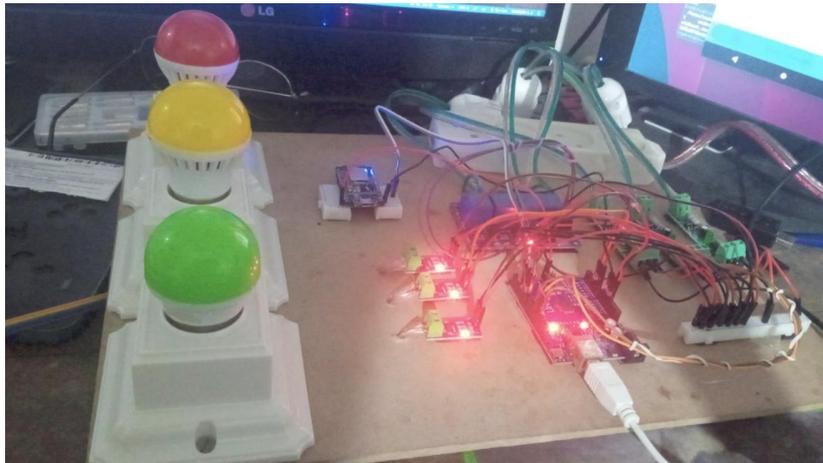
```

**Gambar 4.9** *Syntax Node.js*

Diatas merupakan *syntax server node.js*, disini *.server* menerima data dari *esp8266* dan kemudian mengirim data tersebut kepada aplikasi *user*.

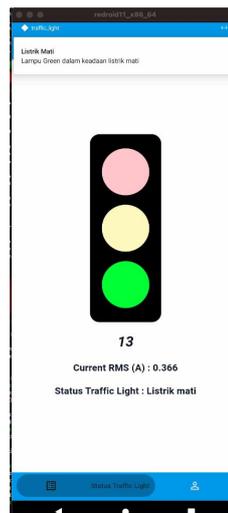
## F. Pengujian Alat

Berikut merupakan pengujian alat yang dilakukan oleh peneliti, pengujian ini terfokus kepada tegangan yang didapatkan oleh *sensor voltage detector* dan juga nilai arus yang didapatkan oleh *sensor acs712*. Pengujian pertama adalah dengan mematikan listrik pada semua lampu seperti pada gambar dibawah



**Gambar 4.10** Pengujian Alat 1

Terlihat pada gambar diatas bahwa semua listrik lampu tidak dicolokkan dan lampu tidak menyala. Dibawah merupakan tampilan aplikasi



**Gambar 4.11** Tampilan Aplikasi

Seperti yang terlihat, status berubah menjadi listrik mati disebabkan listrik tidak disambungkan ke colokan. Dibawah merupakan tabel pengujian

**Tabel 4.6** Pengujian 1 (Lampu Merah)

	Lampu Merah	Lampu Kuning	Lampu Hijau
Tegangan	OFF	OFF	OFF
Arus	0.0 RMS (A)	0.262 RMS (A)	0.262 RMS (A)
Status	Listrik Mati	-	-

**Tabel 4.7** Pengujian 1 (Lampu Kuning)

	Lampu Merah	Lampu Kuning	Lampu Hijau
Tegangan	OFF	OFF	OFF
Arus	0.0 RMS (A)	0.262 RMS (A)	0.262 RMS (A)
Status	-	Listrik Mati	-

**Tabel 4.8** Pengujian 1 (Lampu Hijau)

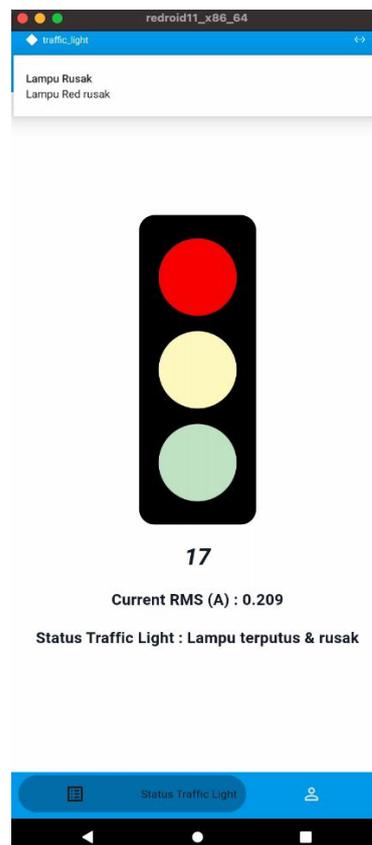
	Lampu Merah	Lampu Kuning	Lampu Hijau
Tegangan	OFF	OFF	OFF
Arus	0.0 RMS (A)	0.262 RMS (A)	0.262 RMS (A)
Status	-	-	Listrik Mati

Pengujian selanjutnya adalah dengan menyambungkan listrik lampu ke colokan colokan dan melonggarkan sambungan lampu seperti pada gambar dibawah



**Gambar 4.12** Pengujian Alat 2

Dapat terlihat diatas bahwa listrik lampu telah tersambung di colokan dan melonggarkan sambungan lampu. Dibawah merupakan tampilan aplikasi setelah listrik lampu dicolokan.



**Gambar 4.13** Tampilan Aplikasi

Seperti yang terlihat, status berubah menjadi Lampu terputus dan rusak disebabkan nilai arus yang berada dibawah 0.262 RMS (A). Dibawah merupakan tabel pengujian

**Tabel 4.9** Pengujian Alat 2 (Lampu Merah)

	<b>Lampu Merah</b>	<b>Lampu Kuning</b>	<b>Lampu Hijau</b>
<b>Tengangan</b>	ON	OFF	OFF
<b>Arus</b>	<b>0.262 RMS (A)</b>	0.262 RMS (A)	0.262 RMS (A)
<b>Status</b>	<b>Lampu terputus &amp; rusak</b>	-	-

**Tabel 4.10** Pengujian Alat 2 ( Lampu Kuning )

	<b>Lampu Merah</b>	<b>Lampu Kuning</b>	<b>Lampu Hijau</b>
<b>Tengangan</b>	OFF	ON	OFF
<b>Arus</b>	0.262 RMS (A)	<b>0.262 RMS (A)</b>	0.262 RMS (A)
<b>Status</b>	-	<b>Lampu terputus &amp; rusak</b>	-

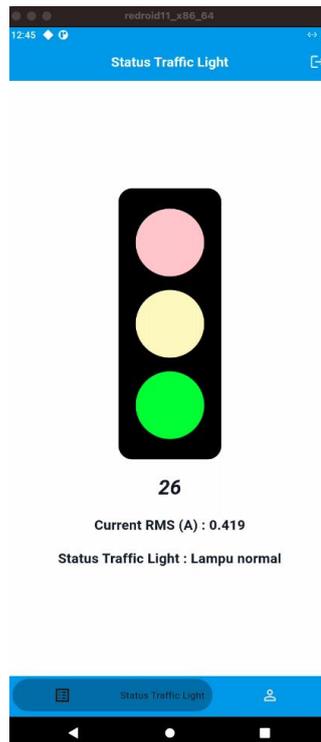
**Tabel 4.11** Pengujian Alat 2 ( Lampu Hijau )

	<b>Lampu Merah</b>	<b>Lampu Kuning</b>	<b>Lampu Hijau</b>
<b>Tengangan</b>	OFF	OFF	ON
<b>Arus</b>	0.262 RMS (A)	0.262 RMS (A)	<b>0.262 RMS (A)</b>
<b>Status</b>	-	-	<b>Lampu terputus &amp; rusak</b>

Pengujian selanjutnya adalah dengan menyambungkan listrik lampu ke colokan dan mengencangkan sambungan seperti pada gambar dibawah

**Gambar 4.14** Pengujian Alat 3

Dapat terlihat diatas bahwa listrik lampu telah tersambung di colokan dan mengencangkan sambungan lampu. Dibawah merupakan tampilan aplikasi setelah listrik lampu dicolokan.



**Gambar 4.15** Tampilan Aplikasi

Seperti yang terlihat, status berubah menjadi Lampu normal disebabkan nilai arus yang berada sekitar  $0.314 - 0.419 \text{ RMS (A)}$ . Dibawah merupakan tabel pengujian

**Tabel 4.12** Pengujian Alat 3 (Lampu Merah)

	Lampu Merah	Lampu Kuning	Lampu Hijau
<b>Tengangan</b>	ON	OFF	OFF
<b>Arus</b>	<b>0.31 – 0.471 RMS (A)</b>	0.262 RMS (A)	0.262 RMS (A)
<b>Status</b>	Lampu nomal	-	-

**Tabel 4.13** Pengujian Alat 3 (Lampu Kuning)

	Lampu Merah	Lampu Kuning	Lampu Hijau
<b>Tengangan</b>	OFF	ON	OFF
<b>Arus</b>	0.262 RMS (A)	<b>0.31 – 0.471 RMS (A)</b>	0.262 RMS (A)
<b>Status</b>	-	Lampu nomal	-

**Tabel 4.14** Pengujian Alat 3 (Lampu Hijau)

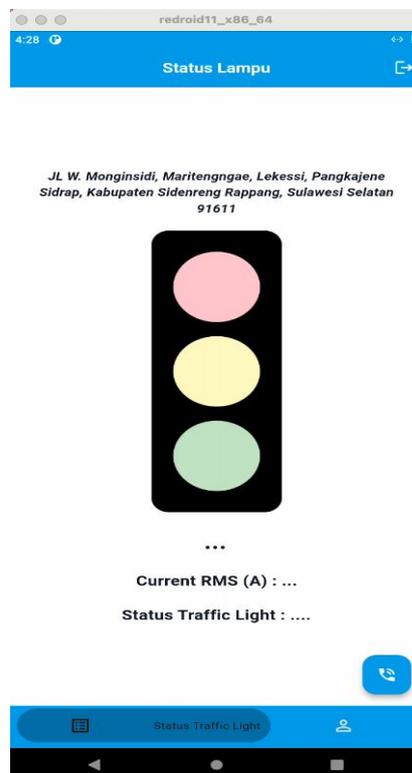
	Lampu Merah	Lampu Kuning	Lampu Hijau
Tengangan	OFF	OFF	ON
Arus	0.262 RMS (A)	0.262 RMS (A)	<b>0.31 – 0.471 RMS (A)</b>
Status	-	-	<b>Lampu nomal</b>

Dibawah merupakan pengujian jika listrik terpadam. Alat tidak dapat mengirim data kepada server hingga menyebabkan tidak adanya data ditampilkan ke aplikasi

**Tabel 4.15** Pengujian Listrik Padam

	Lampu Merah	Lampu Kuning	Lampu Hijau
Tengangan	-	-	-
Arus	-	-	-
Status	-	-	-

Berikut dibawah merupakan tampilan aplikasi jika listrik terpadam.

**Gambar 4.16** Tampilan Aplikasi Jika Listrik Padam

Berikut merupakan pengujian jika terdapat balon lampu mati pada salah satu lampu, seperti yang dilihat, status balon mati sama seperti status lampu rusak pada tabel sebelumnya.

**Tabel 4.16** Pengujian Balon Merah Mati

	<b>Lampu Merah</b>	<b>Lampu Kuning</b>	<b>Lampu Hijau</b>
<b>Tengangan</b>	<b>ON</b>	OFF	OFF
<b>Arus</b>	<b>0.262 RMS (A)</b>	0.262 RMS (A)	0.262 RMS (A)
<b>Status</b>	<b>Lampu terputus &amp; rusak</b>	-	-

**Tabel 4.17** Pengujian Balon Kuning Mati

	<b>Lampu Merah</b>	<b>Lampu Kuning</b>	<b>Lampu Hijau</b>
<b>Tengangan</b>	OFF	<b>ON</b>	OFF
<b>Arus</b>	0.262 RMS (A)	<b>0.262 RMS (A)</b>	0.262 RMS (A)
<b>Status</b>	-	<b>Lampu terputus &amp; rusak</b>	-

**Tabel 4.18** Pengujian Balon Hijau Mati

	<b>Lampu Merah</b>	<b>Lampu Kuning</b>	<b>Lampu Hijau</b>
<b>Tengangan</b>	OFF	OFF	<b>ON</b>
<b>Arus</b>	0.262 RMS (A)	0.262 RMS (A)	<b>0.262 RMS (A)</b>
<b>Status</b>	-	-	<b>Lampu terputus &amp; rusak</b>

## **BAB V**

### **PENUTUP**

#### **A. Kesimpulan**

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan pada bab-bab sebelumnya, Dihasilkannya sebuah alat dan aplikasi Sistem Monitoring Disfungsi *Traffic Light* yang sudah bebas dari kesalahan. Maka dapat ditarikkan kesimpulan seperti berikut :

1. Penelitian ini berhasil merancang dan membangun sebuah prototipe Sistem Monitoring Disfungsi Traffic Light.
2. Kontroller yang digunakan adalah *arduino uno* dan *esp8266* yang mana *arduino uno* berfungsi untuk mengambil nilai yang dikirim oleh *sensor acs712* dan *sensor voltage detector* dan kemudiannya mengimkan. Nilai tersebut ke *esp8266* untuk dikirimkan ke *server*.
3. Sensor yang digunakan adalah *sensor voltage detector* yang berfungsi untuk mendeteksi tegangan listik lampu dan *sensor acs712* yang nerfungsi untuk mengambil nilai *current* lampu.
4. Aplikasi menampilkan waktu *taffic light*, nilai *current RMS (A)* dan juga status *traffic light*.

#### **B. Saran**

Berdasarkan kesimpulan ang telah dibuat, maka penulis memberikan saran untuk pengembangan aplikasi kedepannya seperti berikut yang mana dapat

menyimpan data kedalam database yang nantinya digunakan memonitoring keadaan lampu traffic light secara berkala.

## DAFTAR PUSTAKA

- Adi Chandra, Idham M, Nurul Faizah A, (2023). *“Prototype For Prohibiting Drivers To Passage Zebra Cross In Esp32 Cam-Based Traffic Light”*. Lajagoe Journal Informatic, Multimedia, Dan Technology (LAJUTEK), Vol 1, No.1
- Adi Nugroho. (2019). *Rekayasa Perangkat Lunak Menggunakan UML & Java*.
- Adriansyah, A., Oka, H. (2013). *Rancang Bangun Prototipe Elevator menggunakan Microcontroller Arduino ATMEGA 328P*. Jurnal Teknologi Elektro, 4(3).
- Agus Setiawan, Achiruddin. (2018). *“Aplikasi Prasarana Lalu Lintas Pada DinasPerhubungan Pemprov Kalimantan Selatan Berbasis Website”*.
- Andreas, L., Dimas, S, N. (2023). *Pengaplikasian Sensor Arus ACS712 sebagai Sistem Proteksi pada Alat Penghitung Kertas Otomatis Berbasis IoT*. Journal of Mechanical and Electrical Technology, 2(1).
- Aprisa, 2015. *Rancang Bangun Sistem Informasi Monitoring Perkembangan Proyek Berbasis Web (Studi Kasus: PT Inti Pratama Semesta)*. JurnalRekayasa dan Manajemen Sistem Informasi, 1(1), pp. 49- 54.
- Bram. (2011). *Rancang Bangun Prototipe Pengatur Lampu Lalu Lintas Memanfaatkan Sensor Tekan, Skripsi Teknik Elektro, Universitas Indonesia*.
- Jufri. (2017). *“Simulasi Pengaturan Lampu Lalu Lintas (Traffic Light) Berbasis Mikrokontroler”*.
- Kamal,. Firdayanti., Ulfa, M, T., Andi, A, B., Pattasang. (2023). *Implementasi Aplikasi Arduino Ide pada Mata Kuliah Sistem Digital*. TEKNOS (Jurnal Pendidikan dan Teknologi), 1(1).
- Makruf, A., Rizal, R., Pratika, S, N., Wiki, J., Rani, N, A. (2020). *Pengukuran Tegangan, Arus, Daya pada Prototype PLTS Berbasis Mikrokontroller Arduin Uno*. SainETIn (Jurnal Sain, Energi, Teknologi & Industri), 5(1).
- Munawar. (2018), *Pemodelan Visual dengan UML*, Graha Ilmu, Yogyakarta, 17-100.
- Penfold. (2002). *Dasar-Dasar Elektronik Untuk Pemula*. Bandung : CV. Pionir Jaya.

- R. Sulistyorini, A. Ilham. (2019). "*Pengembangan Sistem Monitoring Lampu Lalu Lintas Berbasis Microcontroller Dengan Fitur SMS Jaringan GSM*".
- Republik Indonesia. (2009). *Undang-Undang No.22 tahun 2009, Tentang Lalu Lintas dan Angkutan Jalan*.
- Rizal, P., Yuliarman, S., Ulinuha, L. (2023). *Rancangan Bangun Alat Monitoring Arus dan Tegangan Berbasis Mikrokontroler pada Studi Kasus Prototype Gardu Distribusi PLN*. RELE (Rekayasa Elektrikal dan Energi) : Jurnal Teknik Elektro, 5(2).
- Roger, S. Pressman, Ph.D. (2012). *Rekayasa Perangkat Lunak (Pendekatan Praktisi) Edisi 7 : Buku 1* ", Yogyakarta: Andi.
- Sendi, A, S., Manurul, H., Sutarti. (2024). *Prototype Lampu Penerangan Jalan Otomatis menggunakan Sensor Ldr berbasis Arduino UNO*. Jurnal PROSISKO, 11(1).
- Sholihq. (2019) . *Pemodelan Sistem Informasi Berorientasi Objek dengan UML*. Graha Ilmu. Yogyakarta.
- Texas Instrument. (2015). *Adafruit Learning System. Adafruit INA219 Current Sensor Breakout. Page 3 of 33. Adafruit Industries*.
- Ulumudin, N. Faizah, and W. Nurcahyo. (2023). "*Aplikasi Sistem Presensi Pegawai PT. Berkah Pena Ilmu dengan Metode Location Based Service (LBS) Berbasis Android Menggunakan Firebase*," Des. J, 1(1).
- Universitas Muhammadiyah Parepare. (2021). *Panduan Akademik 2020-2021*, Parepare : UMPAR Press.
- Yohanes, D., Sri, N, W. (2024). *Perancangan Sistem Informasi Berbasis Web pada D'lofa Laundry menggunakan Metode Waterfall*. IJCSR: The Indonesian Journal of Computer Science Research, 3(1).
- Zainul, J, dkk. (2023). *Pelatihan Pembuatan Web Design Menggunakan HTML dan CSS di SMK Letris Indonesia 2*. AMMA (Jurnal Pengabdian Masyarakat), 1(12).