

**SMART TRAFFIC LIGHT BERBASIS OUTSEAL PLC  
MENGUNAKAN WEBCAM DAN KY-037**

**SKRIPSI**

**OLEH:**

**DIAN PUTRA SARUMAHA  
198120058**



**PROGRAM STUDI TEKNIK ELKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2023**

**UNIVERSITAS MEDAN AREA**

© Hak Cipta Di Lindungi Undang-Undang

Document Accepted 18/9/23

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Access From (repository.uma.ac.id)18/9/23

# **SMART TRAFFIC LIGHT BERBASIS OUTSEAL PLC MENGUNAKAN WEBCAM DAN KY-037**

## **SKRIPSI**

Diajukan sebagai Salah Satu Syarat untuk Memperoleh  
Gelar Sarjana di Fakultas Teknik  
Universitas Medan Area



**Oleh:**

**DIAN PUTRA SARUMAHA  
198120058**

**PROGRAM STUDI TEKNIK ELKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2023**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

Document Accepted 18/9/23

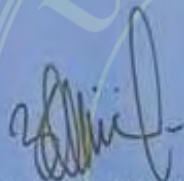
1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Access From (repository.uma.ac.id)18/9/23

## HALAMAN PENGESAHAN

Judul Skripsi : *Smart Traffic Light* Berbasis Outseal PLC menggunakan Webcam dan Sensor KY-037  
Nama : Dian Putra Sarumaha  
NPM : 19.812.0058  
Fakultas : Teknik

Disetujui Oleh  
Komisi Pembimbing

  
Moranain Mungkin, S.T, M.Si  
Pembimbing I

  
Ir. Habib Satria, M.T, IPP  
Pembimbing II

  
Dik Rahmat Syah, M.Kom  
Dekan

  
Ir. Habib Satria, M.T, IPP  
Ka. Prodi

Tanggal Lulus : 30 Agustus 2023

### HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 12 September 2023



Dian Putra Sarumaha  
19.812.0058

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR/SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

---

---

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan dibawah ini :

Nama : Dian Putra Sarumaha  
NPM : 19.812.0058  
Program Studi : Teknik Elektro  
Fakultas : Teknik  
Jenis Karya : Tugas Akhir/Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul : “*Smart Traffic Light Berbasis Outseal PLC menggunakan Webcam dan Sensor KY-037*”.

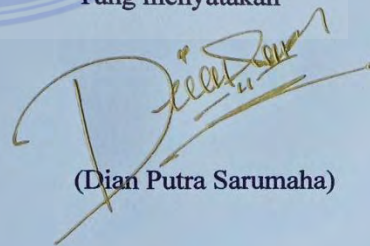
beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Medan Area berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir/skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Medan

Pada Tanggal : 12 – 09 – 2023

Yang menyatakan



(Dian Putra Sarumaha)

## ABSTRAK

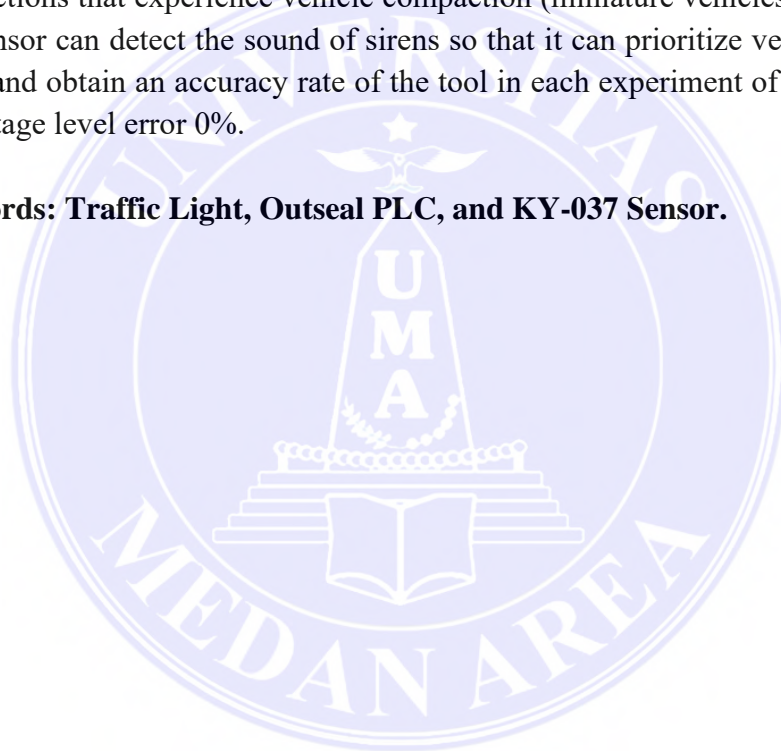
Lampu lalu lintas adalah sinyal yang mengontrol dan mengatur arus lalu lintas. Lampu lalu lintas biasanya dipasang di persimpangan jalan (lintasan penyeberangan). Kemacetan disebabkan oleh beberapa faktor antara lain lampu lalu lintas tidak berfungsi secara efektif. Sistem *smart traffic light* mampu mendahulukan persimpangan berdasarkan tingkat kepadatan lalu lintas yang berhenti di persimpangan, dimana ketentuan nyala lampu tersebut diatur menggunakan Outseal PLC. Dan juga dapat mengatur *traffic light* untuk dapat memprioritaskan kendaraan khusus menggunakan sensor KY-037. Dalam penelitian ini melakukan beberapa tahapan antara lain : mengenali masalah, perancangan, pembuatan dan pengujian alat. Dari hasil pengujian yang dilakukan sistem *smart traffic light* mampu mendahulukan simpang yang mengalami pematatan kendaraan (kendaraan miniatur) dan sensor KY-037 mampu mendeteksi suara sirine sehingga bisa memprioritaskan kendaraan yang menggunakan sirine serta didapatkan tingkat akurasi alat pada setiap percobaan sebesar 100% dan tingkat presentase error 0%.

**Kata Kunci :** *Traffic Light, Outseal PLC, dan Sensor KY-037*

## ABSTRACT

Traffic lights are signals that control and regulate traffic flow. Traffic lights are usually installed at crossroads (crosswalks). Congestion is caused by several factors, including traffic lights not functioning effectively. The smart traffic light system can prioritize intersections based on the density level of traffic stopping at the intersection, where the conditions for the lights to turn on are regulated using Outseal PLC. And also can set the traffic light to be able to prioritize special vehicles using the KY-037 sensor. In this study, several stages were carried out including recognizing the problem, designing, manufacturing, and testing the tool. From the results of tests carried out by the smart traffic light system, it can prioritize intersections that experience vehicle compaction (miniature vehicles) and the KY-037 sensor can detect the sound of sirens so that it can prioritize vehicles that use sirens and obtain an accuracy rate of the tool in each experiment of 100% and the percentage level error 0%.

**Keywords: Traffic Light, Outseal PLC, and KY-037 Sensor.**



## RIWAYAT HIDUP

Penulis dilahirkan di Siwalawa Pada tanggal 12 September 2001 dari ayah Niada Sarumaha dan ibu Suliami Sarumaha. Penulis merupakan anak ke-2 dari 5 bersaudara.

Tahun 2019 Penulis lulus dari SMK NEGERI 5 DUMAI dan pada tahun 2019 terdaftar sebagai mahasiswa Fakultas Teknik Universitas Medan Area.

Pada tanggal 20 Juli sampai 25 Agustus tahun 2022 Penulis melaksanakan Praktek Kerja Lapangan (PKL) di PT. Kuala Lumpur Kepong (KLK) Dumai.





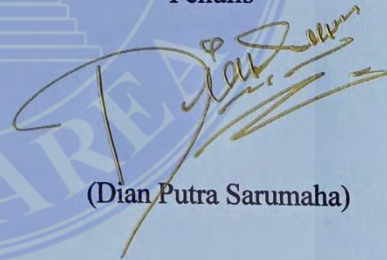
## KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala karunia-Nya sehingga skripsi ini berhasil diselesaikan. Tema yang dipilih dalam penelitian ini ialah sistem kendali dengan judul “*Smart Traffic Light Berbasis Outseal PLC menggunakan Webcam dan Sensor KY-037*”.

Terima kasih penulis sampaikan kepada Bapak Moranain Mungkin, S.T, M.Si dan Ir. Habib Satria, M.T, IPP Selaku pembimbing yang telah membimbing dan mengarahkan serta memberikan saran dalam penyusunan skripsi ini. Ungkapan terima kasih juga disampaikan kepada ayah, ibu, serta seluruh keluarga atas segala do’a dan perhatiannya.

Penulis menyadari bahwa tugas akhir/skripsi ini masih memiliki kekurangan, oleh karena itu kritik dan saran yang bersifat membangun sangat penulis harapkan demi kesempurnaan tugas akhir/skripsi ini. Penulis berharap tugas akhir/skripsi ini dapat bermanfaat baik untuk kalangan pendidikan maupun masyarakat. Akhir kata penulis ucapkan terima kasih.

Penulis



(Dian Putra Sarumaha)

## DAFTAR ISI

|  | Halaman |
|--|---------|
| HALAMAN PENGESAHAN.....                                | ii      |
| HALAMAN PERNYATAAN .....                               | iii     |
| HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....          | iv      |
| ABSTRAK  |         |
| ABSTRACT   |         |
| RIWAYAT HIDUP.....                                     | v       |
| KATA PENGANTAR .....                                   | vi      |
| DAFTAR GAMBAR .....                                    | x       |
| DAFTAR TABEL.....                                      | xi      |
| I PENDAHULUAN .....                                    | 1       |
| 1.1 Latar Belakang .....                               | 1       |
| 1.2 Perumusan Masalah .....                            | 2       |
| 1.3 Tujuan Penelitian .....                            | 2       |
| 1.4 Batasan Masalah .....                              | 3       |
| 1.5 Manfaat Penelitian .....                           | 3       |
| 1.6 Sistematika penulisan.....                         | 3       |
| II TINJAUAN PUSTAKA .....                              | 5       |
| 2.1 <i>Traffic Light</i> .....                         | 5       |
| 2.2 PLC ( <i>Programmable Logic Controller</i> ) ..... | 5       |
| 2.3 Outseal PLC .....                                  | 6       |
| 2.4 Perangkat Lunak Outseal Studio.....                | 7       |
| 2.4.1 Ladder Diagram .....                             | 8       |
| 2.4.2 Kontak NO ( <i>Normally Open</i> ) .....         | 8       |
| 2.4.3 Kontak NC ( <i>Normally Close</i> ) .....        | 9       |
| 2.4.4 Output .....                                     | 10      |
| 2.4.5 <i>Timer TON (Timer On Delay)</i> .....          | 10      |
| 2.4.6 EQU ( <i>EQUAL</i> ).....                        | 11      |

|        |   |    |
|--------|---|----|
| 2.4.7  | GEQ ( <i>Greater than or equal to</i> ) .....                         | 12 |
| 2.4.8  | ADD ( <i>Addition</i> ) .....   | 12 |
| 2.4.9  | COPY .....  | 12 |
| 2.4.10 | CLR ( <i>Clear</i> ) .....  | 13 |
| 2.5    | Sensor .....  | 13 |
| 2.5.1  | Sensor Suara .....  | 13 |
| 2.6    | Program OpenCV-Python .....   | 14 |
| 2.6.1  | Sejarah OpenCV .....  | 14 |
| 2.6.2  | Fitur pada OpenCV .....   | 15 |
| 2.6.3  | OpenCV dan Python .....   | 15 |
| III    | METODOLOGI PENELITIAN .....   | 16 |
| 3.1    | Waktu Dan Tempat Penelitian .....                                     | 16 |
| 3.1.1  | Tempat Penelitian .....   | 16 |
| 3.1.2  | Waktu Penelitian .....  | 16 |
| 3.2    | Tahapan dan Metodologi Penelitian .....                               | 16 |
| 3.3    | Alat dan Bahan .....  | 21 |
| 3.4    | Desain Alat Miniatur <i>Traffic Light</i> dan Tata Letak Komponen ... | 23 |
| 3.5    | Diagram Blok Alat .....   | 24 |
| 3.6    | Diagram Alir Sistem Cara Kerja Alat .....                             | 26 |
| 3.7    | Pembuatan Perangkat Keras .....                                       | 27 |
| 3.8    | Pembuatan Perangkat Lunak .....                                       | 34 |
| IV     | PENGUJIAN DAN ANALISIS .....  | 44 |
| 4.1    | Hasil Pembuatan Alat .....  | 44 |
| 4.2    | Pengujian Alat .....  | 44 |
| 4.2.1  | Pengujian Alat (Bagian “a”) .....                                     | 45 |
| 4.2.2  | Pengujian Alat (Bagian “b”) .....                                     | 47 |
| V      | KESIMPULAN DAN SARAN .....  | 49 |
| 5.1    | Kesimpulan .....  | 49 |

|                      |    |
|----------------------|----|
| 5.2 Saran .....      | 49 |
| DAFTAR PUSTAKA ..... | 51 |

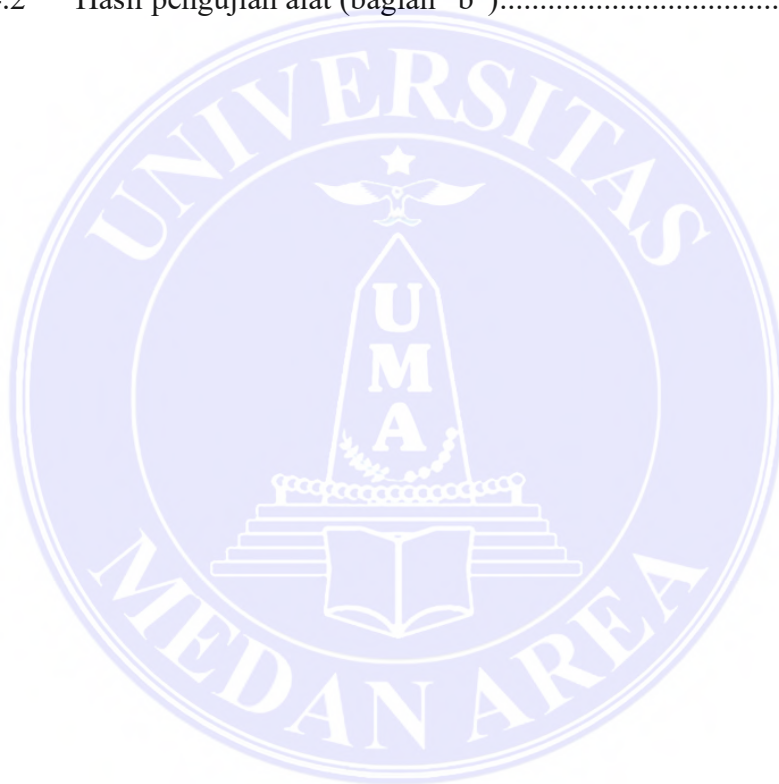


## DAFTAR GAMBAR

|   | Halaman |
|---|---------|
| Gambar 2.1 Blok Diagram PLC .....   | 6       |
| Gambar 2.2 Outseal PLC.....   | 7       |
| Gambar 2.3 Bentuk perangkat lunak outseal studio.....   | 7       |
| Gambar 2.4 Istilah dalam diagram ladder outseal PLC.....  | 8       |
| Gambar 2.5 Instruksi kontak NO .....  | 9       |
| Gambar 2.6 Instruksi kontak NC.....   | 9       |
| Gambar 2.7 Instruksi output outseal studio.....   | 10      |
| Gambar 2.8 Parameter <i>timer</i> TON .....   | 11      |
| Gambar 2.9 Penggunaan ADD.....  | 12      |
| Gambar 2.10 CLR (Clear).....  | 13      |
| Gambar 2.11 Sensor suara.....   | 14      |
| Gambar 3.1 Diagram alir penelitian.....   | 20      |
| Gambar 3.2 Desain Miniatur <i>traffic light</i> 3 simpang tampak atas .....                           | 23      |
| Gambar 3.3 Desain tampak depan <i>smart traffic light</i> beserta tata letak sensor                   | 24      |
| Gambar 3.4 Diagram blok alat .....  | 24      |
| Gambar 3.5 Diagram alir cara kerja <i>smart traffic light</i> .....                                   | 27      |
| Gambar 3.6 Bentuk fisik adaptor tegangan 12 Vdc .....   | 28      |
| Gambar 3.7 Rangkaian skematik adaptor 12 Vdc.....   | 28      |
| Gambar 3.8 Tata letak komponen .....  | 29      |
| Gambar 3.9 Rangkaian instalasi kamera webcam dengan komputer atau laptop .....                        | 30      |
| Gambar 3.10 Rangkaian sensor suara dengan Outseal PLC .....   | 31      |
| Gambar 3.11 Rangkaian hubungan modul <i>traffic light</i> dengan outseal PLC.....                     | 33      |
| Gambar 3.12 Rangkaian komunikasi antara modbus atmega 32 dengan outseal PLC.....                      | 34      |
| Gambar 4.1 Smart <i>traffic light</i> berbasis outseal PLC menggunakan webcam dan sensor KY-037 ..... | 44      |

## DAFTAR TABEL

|           | Halaman  |
|-----------|--|
| Tabel 2.1 | Keterangan status pada <i>timer</i> TON..... 11  |
| Tabel 2.2 | EQU ..... 11                                     |
| Tabel 2.3 | GEQ ..... 12                                     |
| Tabel 2.4 | Jadwal Penelitian ..... 16                       |
| Tabel 3.1 | Tabel Alat dan Bahan dan Spesifikasinya ..... 21 |
| Tabel 4.1 | Hasil pengujian alat (bagian “a”)..... 46        |
| Tabel 4.2 | Hasil pengujian alat (bagian “b”)..... 47        |



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Lampu lalu lintas adalah sinyal yang mengontrol dan mengatur arus lalu lintas. Menurut Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, lampu lalu lintas adalah alat pemberi isyarat lalu lintas atau APILL. Lampu lalu lintas biasanya dipasang di persimpangan jalan (lintasan penyeberangan). Masalah kemacetan lalu lintas saat ini menjadi masalah yang melanda hampir setiap kota besar di dunia, termasuk kota-kota besar di Indonesia. Menurut (Usman dkk., 2020) Kemacetan lalu lintas disebabkan oleh sejumlah faktor, salah satunya lampu lalu lintas yang tidak efektif. Ketidakseimbangan ini menyebabkan kemacetan berulang dan terus menerus ketika kendaraan menumpuk di beberapa jalur (Anas, Sumaryo dan Estanto, 2019).

Lampu lalu lintas yang digunakan saat ini berwarna merah, kuning atau hijau dengan durasi tetap. Lampu lalu lintas ini sangat tidak efisien dalam mengatur arus lalu lintas yang terus meningkat. Menurut (Nursalim, 2022) pengurangan kemacetan lalu lintas diimplementasikan dengan memperkenalkan kontrol lampu lalu lintas terintegrasi atau sistem kontrol gabungan.

Telah dilakukan penelitian sebelumnya (Setiawan, 2022) tentang “Mengatasi Kemacetan di Lampu Merah dengan Pendekatan Image Processing”. Penelitian ini menggunakan pengolahan citra untuk mendeteksi kendaraan yang melintas dengan tujuan menghitung jumlah kendaraan berdasarkan jenisnya untuk memprediksi jumlah kendaraan yang melintas di persimpangan. Penelitian ini masih berupa pengenalan pengolahan citra kendaraan yang dapat digunakan untuk mengatasi kemacetan simpang.

Penelitian sebelumnya pernah dilakukan oleh (Rosyady, Ikhsan dan Feter, 2022) dengan judul “Prototype Lampu Lalu Lintas Adaptif Berdasarkan Panjang Antrian Kendaraan Berbasis Arduino Uno”. Pada penelitian ini, sensor infra merah digunakan untuk mendeteksi kepadatan lalu lintas. Pada penelitian ini kepadatan lalu lintas dideteksi dengan sensor infra merah, dimana letak sensor di pinggir jalan

dapat terganggu oleh kendaraan lain yang berhenti di sebelah sensor, yang menyebabkan pengatur waktu lampu lalu lintas terganggu.

Penelitian ini akan merancang sebuah sistem *smart traffic light* yang dapat mendahulukan persimpangan berdasarkan tingkat kepadatan lalu lintas yang berhenti di persimpangan, dimana ketentuan nyala lampu tersebut diatur menggunakan Outseal PLC. Dan juga dapat mengatur *traffic light* untuk dapat memprioritaskan kendaraan khusus menggunakan sensor KY-037. Untuk mengetahui tingkat kemacetan yang terjadi, maka pada setiap simpang jalan akan ditempatkan satu buah kamera Webcam untuk mengetahui kondisi kepadatan persimpangan lalu lintas.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, masalah yang akan dibahas dalam penelitian ini adalah sebagai berikut :

1. Bagaimana membuat sistem *traffic light* agar dapat memprioritaskan simpang yang mengalami pemadatan lalu lintas?
2. Bagaimana membuat sistem *traffic light* agar dapat memprioritaskan kendaraan darurat?
3. Bagaimana memonitoring kepadatan lalu lintas dan kendaraan darurat?

## 1.3 Tujuan Penelitian

Berdasarkan perumusan masalah tujuan penelitian yang akan dicapai antara lain:

1. Untuk dapat memprioritaskan kepadatan lalu lintas di persimpangan.
2. Untuk dapat memprioritaskan kendaraan darurat yang melewati persimpangan.
3. Untuk dapat memonitoring kepadatan lalu lintas dan kendaraan darurat.



#### 1.4 Batasan Masalah

Adapun batasan masalah dari penelitian ini sebagai berikut:

1. Pembuatan *smart traffic light* menggunakan Outseal PLC Mega V.3, Webcam dan sensor KY-037.
2. Dalam pengoperasian *smart traffic light* memerlukan PC/Laptop agar dapat memproses gambar yang terekam oleh Webcam.
3. Pembuatan *smart traffic light* dalam bentuk prototype 3 simpang.
4. Dalam pembuatan program *smart traffic light* ini menggunakan program *library* OpenCV-Python deteksi warna objek kendaraan miniatur (mobil mainan) bukan menggunakan program algoritma *CascadeClassifier* yang biasa digunakan untuk mendeteksi kendaraan (mampu mendeteksi jenis kendaraan).

#### 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Sistem *traffic light* mampu mengurangi atau memprioritaskan simpang lalu lintas yang mengalami pemadatan arus lalu lintas.
2. Sistem *traffic light* mampu memprioritaskan kendaraan darurat agar tidak terjebak macet di persimpangan lalu lintas.

#### 1.6 Sistematika penulisan

Sistematika penulisan Tugas Akhir ini, tersusun dalam beberapa bab, dengan susunan sebagai berikut:

##### A. BAB. I PENDAHULUAN

Bab ini menerangkan secara singkat latar belakang penelitian, rumusan masalah penelitian, batasan masalah, tujuan penelitian dan manfaat penelitian serta sistematika penulisan.

##### B. BAB. II TINJAUAN PUSTAKA

Bab ini berisi tentang pembahasan teori-teori yang berhubungan dengan pokok pembahasan dalam penelitian tugas akhir sehingga hasil yang akan didapat lebih optimal.

C. BAB III. METODOLOGI PENELITIAN

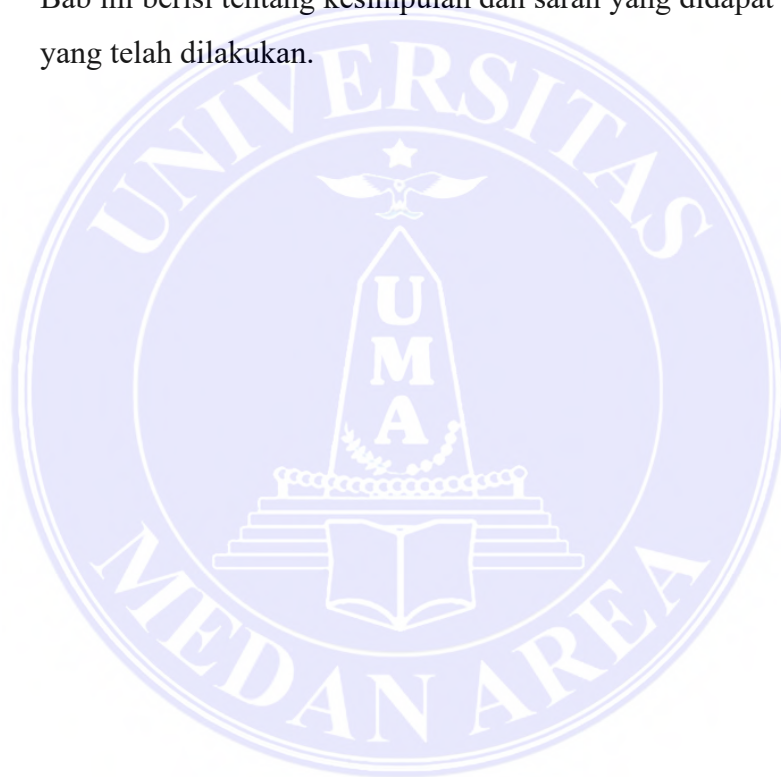
Bab ini memuat tentang metode penelitian yang meliputi waktu dan lokasi penelitian, desain dan metode penelitian.

D. BAB. IV HASIL DAN PEMBAHASAN

Bab ini berisi inti dari semua pembahasan dalam penelitian tugas akhir, yang menjelaskan tentang kegunaan, efisiensi dan manfaat dari penelitian rancang bangun *smart traffic light*.

E. BAB. V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dan saran yang didapat dari penelitian yang telah dilakukan.



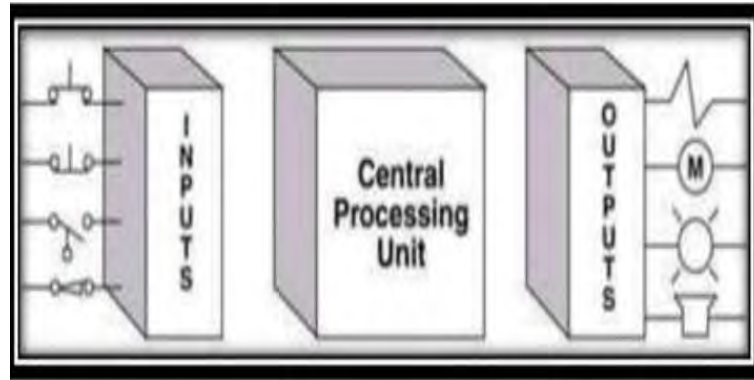
## BAB II TINJAUAN PUSTAKA

### 2.1 *Traffic Light*

Lampu lalu lintas adalah alat pemberi isyarat lalu lintas atau APILL yang mengatur arus lalu lintas di persimpangan, tempat penyeberangan (*zebra cross*), dan area arus lalu lintas lainnya. Lampu ini menandakan bahwa kendaraan harus bergantian melaju dan berhenti dari arah yang berbeda. Pengaturan lalu lintas pada persimpangan dimaksudkan untuk mengatur pergerakan kendaraan secara berkelompok secara bergantian agar kendaraan tidak saling mengganggu dalam lalu lintas yang ada. Lampu lalu lintas terdiri dari tiga warna: merah, kuning dan hijau. Arti dari setiap warna lampu tersebut adalah lampu merah berarti harus berhenti, lampu kuning berarti bersiap untuk jalan jika kondisi sebelumnya dalam keadaan berhenti dan menurunkan kecepatan jika kondisi sebelumnya dalam keadaan berjalan, serta lampu hijau yang berarti dapat berjalan (Muhammad, Munadi dan Mayasari, 2019).

### 2.2 *PLC (Programmable Logic Controller)*

PLC merupakan perangkat yang digunakan untuk menggantikan fungsi rangkaian relai sekuensial dari sistem kontrol panel (Fariz Elazar Ahmad dan Endah Fitriani, 2020). PLC bertindak sebagai pengontrol keadaan logika (kondisi hidup atau mati) untuk perangkat lain yang terhubung ke PLC, dan yang skema kontrolnya dapat diubah (diprogram). Pemrograman PLC dilakukan oleh perangkat lunak yang berjalan di komputer (PC). Contoh aplikasi PLC adalah pengontrolan lampu lalu lintas, sistem otomatis mesin pabrik, dan lain – lain. PLC memiliki tiga bagian utama, yaitu input, CPU, dan output.

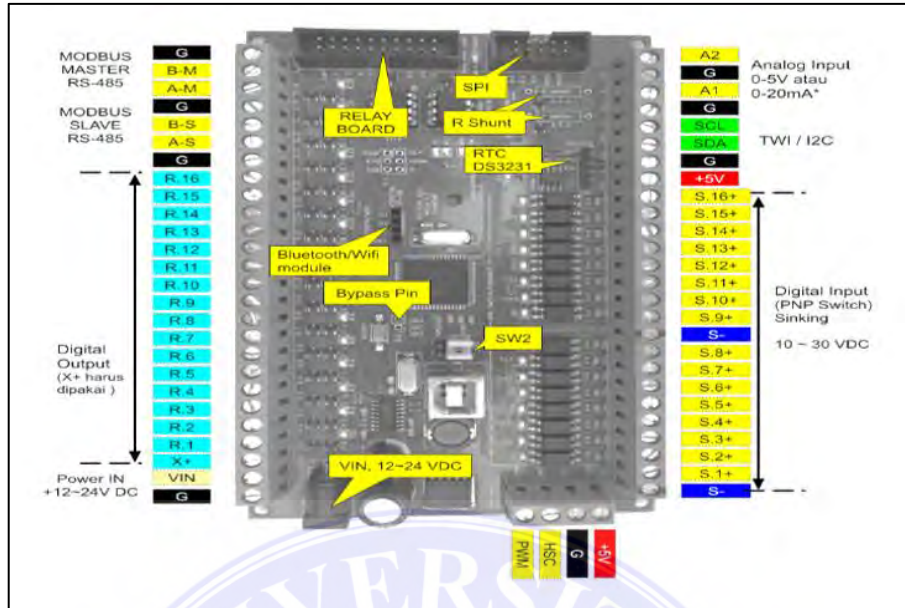


Gambar 2.1 Blok Diagram PLC

Sumber : (Fariz Elazar Ahmad dan Endah Fitriani, 2020)

### 2.3 Outseal PLC

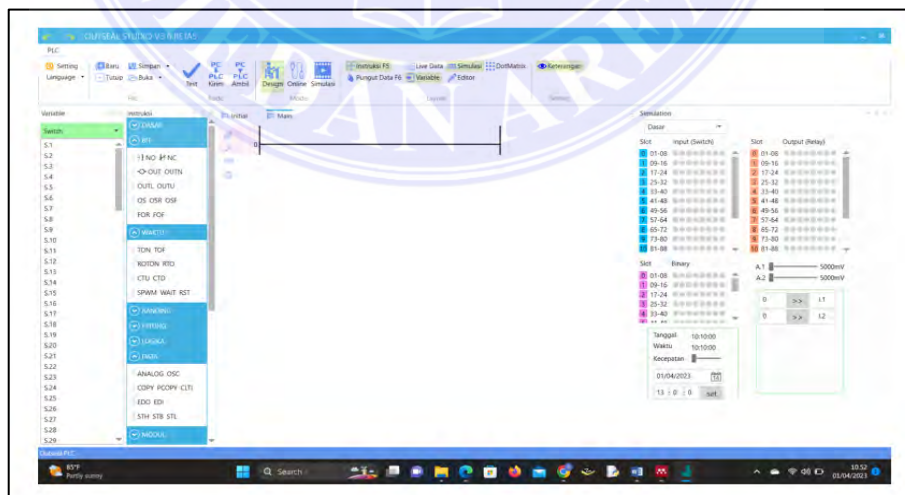
Outseal PLC adalah teknologi otomasi buatan dalam negeri atau buatan anak negeri. Desain logika kontrol outseal memerlukan perangkat lunak yang disebut outseal studio, yang juga merupakan produk outseal. Outseal studio bekerja dalam format pemrograman visual di komputer menggunakan diagram tangga. Diagram tangga adalah hasil dari logika kontrol, yang kemudian dikirim melalui kabel USB ke perangkat keras outseal PLC untuk penyimpanan permanen. Selain itu, kabel USB dapat dilepas, dan outseal dapat secara mandiri melakukan hasil kontrol logika (tidak perlu terhubung ke komputer). PLC biasanya dilengkapi dengan perangkat komunikasi yang dapat dihubungkan ke perangkat eksternal seperti komputer dan HMI. Outseal PLC mencakup semua fungsi perangkat keras dasar yang dimiliki oleh PLC tipikal dan cocok untuk penggunaan industri.



Gambar 2.2 Outseal PLC  
Sumber : (Bakhtiar, 2019)

## 2.4 Perangkat Lunak Outseal Studio

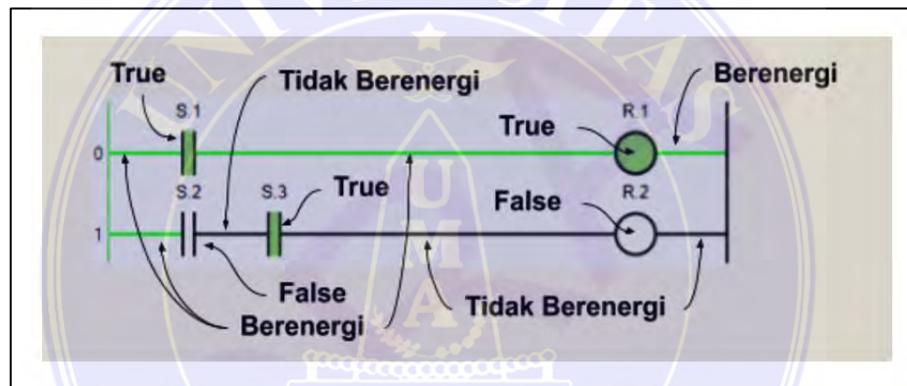
Outseal studio ialah perangkat lunak yang dibuat di komputer (PC) dan menggunakan logika tangga untuk memprogram perangkat keras outseal. Perangkat lunak ini dapat diunduh secara gratis dari situs resmi outseal [www.outseal.com](http://www.outseal.com) (Bakhtiar, 2019).



Gambar 2.3 Bentuk perangkat lunak outseal studio

### 2.4.1 Ladder Diagram

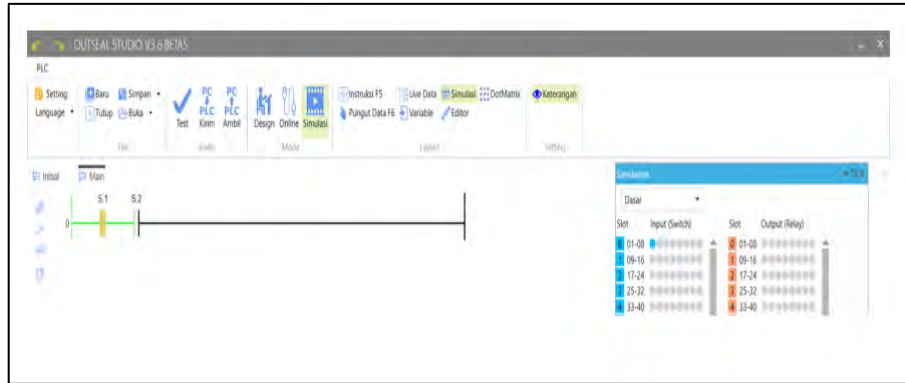
Diagram tangga adalah bahasa pemrograman PLC yang menggunakan simbol untuk menggambarkan sakelar dan perangkat keluaran untuk menggambarkan operasi suatu sistem. Perangkat pemrograman menginterpretasikan diagram ke dalam bahasa yang dapat dibaca oleh prosesor PLC. Diagram tangga mirip dengan simulasi saluran listrik, di mana energi listrik mengalir melalui kabel dari kiri ke kanan. Saat perintah sedang berjalan, energi listrik pada input perintah menyediakan energi listrik melalui output perintah. Istilah berenergi atau tidak berenergi adalah sebutan yang digunakan untuk outseal dalam logika tangga atau kawat, dan istilah benar (*true*) dan salah (*false*) digunakan untuk nilai logika atau instruksi status (Bakhtiar, 2019).



Gambar 2.4 Istilah dalam diagram ladder outseal PLC

### 2.4.2 Kontak NO (*Normally Open*)

NO adalah tombol fisik yang keadaannya terbuka (tidak terhubung) saat tombol tidak ditekan. Pernyataan NO berfungsi seperti tombol fisik NO. Tombol fisik menghubungkan sakelar saat tombol ditekan, dan instruksi ini menghubungkan sakelar saat bit sumber logika benar. Jika perintah ini diberi energi dan bit sumber logika juga benar, perintah ini dapat mentransfer energi ke *ladder*.



Gambar 2.5 Instruksi kontak NO

Pada gambar diatas dapat dilihat input S.1 mendapat tegangan yang berarti berstatus terhubung dan yang tidak mendapat tegangan seperti pada S.2 tidak terhubung.

### 2.4.3 Kontak NC (*Normally Close*)

Kontak NC ini berfungsi seperti tombol NC fisik. Artinya, keadaan tertutup (terhubung) saat tombol tidak ditekan, dan terputus saat tombol ditekan. Fungsi dari pernyataan ini adalah kebalikan dari pernyataan NO. Ketika ada energi pada jalur input dan bit sumber secara logika benar, tidak ada energi yang dikirim ke jalur output, tetapi ketika bit sumber salah secara logika, energi dikirimkan.

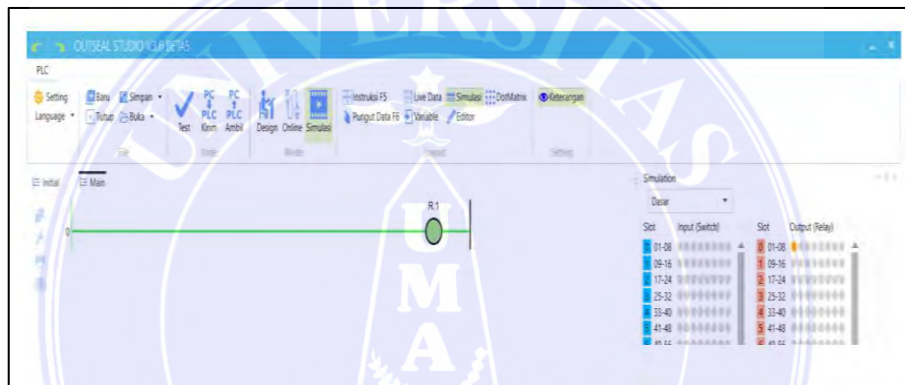


Gambar 2.6 Instruksi kontak NC

Pada gambar diatas dapat dilihat S.1 dalam keadaan normal terhubung (mendapat tegangan) sedangkan pada S.2 dalam keadaan tidak mendapat tegangan (tidak terhubung) atau sudah berubah menjadi Kontak NO.

#### 2.4.4 Output

Output berfungsi menuliskan (*write*) suatu nilai logika (*true / false*) pada sumber data yang merupakan bit tujuan. Bit tujuan harus berupa variabel dengan notasi R dan B. Karena keadaan jalur output hanya dipengaruhi oleh keadaan diagram garis (*ladder*) dan bukan oleh logika bit tujuan, output dapat dihubungkan secara seri tanpa bergantung pada logika komponen sebelumnya.

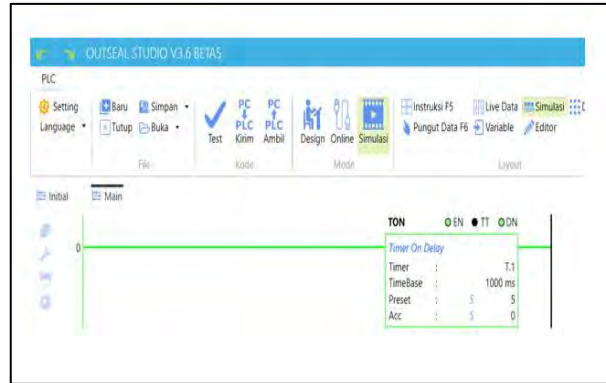


Gambar 2.7 Instruksi output outseal studio

#### 2.4.5 Timer TON (Timer On Delay)

TON adalah instruksi yang digunakan untuk menunda (*delay*) suatu perubahan logika dari *true* ke *false* dengan waktu *delay* yang dapat disesuaikan. Durasi penundaan dapat disesuaikan dengan opsi interval dan kolom *preset*. Interval adalah ukuran basis waktu (*timer base*) dan *preset* adalah jumlah hitungan waktu yang diinginkan.





Gambar 2.8 Parameter *timer* TON

Tabel 2.1 Keterangan status pada *timer* TON

| Singkatan | Status              | Keterangan                                  |
|-----------|---------------------|---|
| EN        | <i>Enable</i>       | Menandakan aktif atau tidak                 |
| TT        | <i>Timing</i>       | Menandakan sedang menghitung atau tidak     |
| DN        | <i>Done</i>         | Menandakan sudah mencapai target atau belum |
| PRE       | <i>Preset</i>       | Nilai target yang diinginkan                |
| ACC       | <i>Accumulation</i> | Nilai akumulasi perhitungan pada timer      |

#### 2.4.6 EQU (*EQUAL*)

EQU adalah sebuah instruksi yang digunakan untuk menguji dua nilai apakah sama atau tidak. Instruksi ini mempunyai dua masukan nilai untuk dibandingkan yakni A dan B. Keduanya dapat diisi variabel atau konstanta.

Tabel 2.2 EQU

| Jenis | Kondisi    | Sifat                      | Sumber               |
|-------|------------|----------------------------|----------------------|
| EQU   | $A = B$    | Menghantarkan energi       | Variable dan konstan |
|       | $A \neq B$ | Tidak menghantarkan energi |                      |

### 2.4.7 GEQ (*Greater than or equal to*)

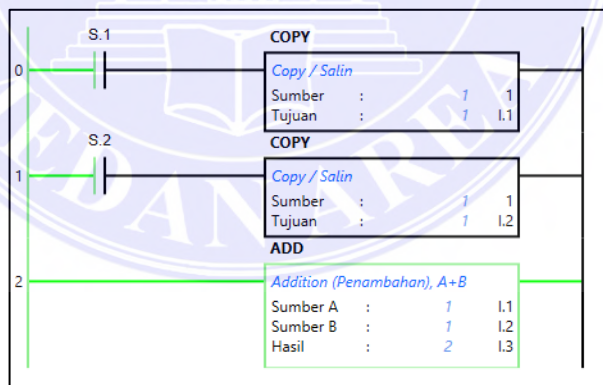
GEQ adalah sebuah instruksi yang digunakan untuk menguji dua nilai apakah nilai pertama lebih besar atau sama. Instruksi ini mempunyai dua masukan nilai untuk dibandingkan yakni A dan B. Keduanya dapat diisi variabel atau konstanta. Apabila kondisi jalur masuk instruksi ini dalam kondisi berenergi dan nilai A lebih besar atau sama dengan B, maka energi tersebut akan dihantarkan ke jalur keluaran.

Tabel 2.3 GEQ

| Jenis | Kondisi | Sifat                      | Sumber               |
|-------|---------|----------------------------|----------------------|
| GEQ   | $A > B$ | Menghantarkan energi       | Variable dan konstan |
|       | $A < B$ | Tidak menghantarkan energi |                      |

### 2.4.8 ADD (*Addition*)

ADD adalah sebuah instruksi yang digunakan untuk menambahkan dua nilai dan hasilnya diletakkan pada sebuah variabel. Instruksi ini mempunyai 2 masukan nilai untuk diproses yakni A dan B dan satu keluaran untuk menampung hasil. Keduanya masukannya dapat diisi variabel atau konstanta. Kondisi jalur keluar dari instruksi ini selalu mengikuti kondisi jalur masuknya.



Gambar 2.9 Penggunaan ADD

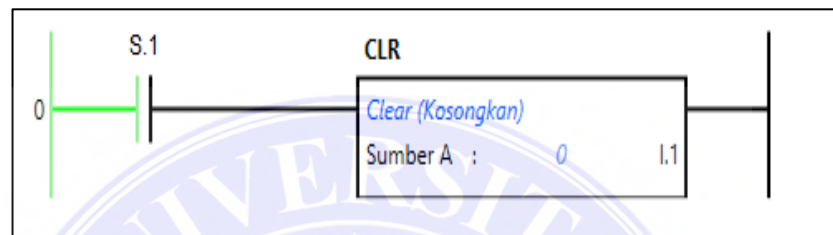
### 2.4.9 COPY

COPY adalah sebuah komponen yang digunakan untuk menyalin nilai suatu variabel. Komponen ini mempunyai 1 masukan nilai yang tidak boleh dalam bentuk

konstanta melainkan dari suatu variabel dan mempunyai satu keluaran untuk menampung hasil.

#### 2.4.10 CLR (Clear)

CLR adalah sebuah instruksi yang digunakan untuk mengubah nilai suatu variabel agar bernilai 0. Instruksi ini hanya membutuhkan satu variabel sebagai inputan dan sekaligus sebagai keluaran.



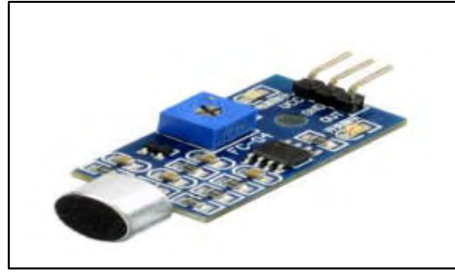
Gambar 2.10 CLR (Clear)

### 2.5 Sensor

Sensor adalah perangkat yang digunakan untuk mendeteksi perubahan besaran fisik seperti tekanan, gaya, besaran listrik, cahaya, gerakan, kelembaban, suhu, kecepatan dan fenomena-fenomena lingkungan lainnya.

#### 2.5.1 Sensor Suara

Sensor suara adalah sebuah alat yang mampu mengubah gelombang sinusioda suara menjadi gelombang sinus energi listrik (*Alternating Sinusioda Electric Current*). Sensor suara bekerja berdasarkan besar kecilnya kekuatan gelombang suara yang mengenai membran sensor yang menyebabkan Bergeraknya membran sensor yang juga terdapat sebuah kumparan kecil di balik membran tersebut naik dan turun. Kecepatan gerak kumparan tersebut menentukan kuat lemahnya gelombang listrik yang dihasilkannya.



Gambar 2.11 Sensor suara

Sumber : (Michael dan Gustina, 2019)

## 2.6 Program OpenCV-Python

*Open Capture Vision* adalah pustaka antarmuka pemrograman aplikasi yang memungkinkan komputer mengenali objek, seperti orang dan benda, sehingga komputer dapat membuat keputusan, mengambil tindakan, dan mengenali objek mana yang menjadi objek target (Warnilah dkk., 2022).

### 2.6.1 Sejarah OpenCV

OpenCV pertama kali diperkenalkan secara resmi pada tahun 1999 oleh *Inter Research* sebagai bagian dari proyek tindak lanjut yang menyediakan aplikasi intensif CPU, pelacakan sinar waktu nyata, dan dinding visual 3D. Tujuan utama dari proyek OpenCV yakni :

1. Riset sensor gambar digital dilakukan tidak hanya dengan kode terbuka, tetapi juga dengan kode yang dioptimalkan untuk infrastruktur sensor gambar.
2. Sebarkan ilmu menggambar dan buat kode Anda lebih mudah dibaca dan portabel dengan menyediakan infrastruktur bersama yang dapat digunakan oleh pengembang.
3. Bangun aplikasi komersial berbasis gambar di mana kode yang dioptimalkan tersedia secara bebas di bawah lisensi terbuka dan program tidak harus terbuka atau gratis.

### 2.6.2 Fitur pada OpenCV

OpenCV ini memiliki banyak fitur yang dapat dimanfaatkan. Fitur utama dari OpenCV yaitu :

1. *Image* dan video I/O
2. *Computer vision* umum untuk pencitraan digital digunakan untuk API tingkat rendah dan menengah.
3. Modul *computer vision* masuk kedalam jenis level tinggi.
4. Sampling gambar dan transformasi

### 2.6.3 OpenCV dan Python

OpenCV adalah *library* (perpustakaan) untuk mengolah gambar dan video. Kata open pada OpenCV berarti *open source* dapat didownload dimana saja secara gratis. Kata OpenCV berasal dari singkatan dari *Computer Vision*, yang menggunakan komputer yang ditangkap untuk memproses gambar yang diambil oleh perangkat penangkap seperti kamera atau webcam, mengubahnya dari analog menjadi digital, dan kemudian mengubahnya menjadi proses. Pemrosesan citra memiliki banyak tujuan dalam pemrosesan citra, banyak di antaranya adalah peningkatan kualitas citra atau identifikasi citra.

Python adalah bahasa pemrograman yang banyak digunakan di dunia komputasi. Kemampuan Python untuk memproses gambar sangat terbatas, jadi kita perlu mengimpor pustaka dari OpenCV.

OpenCV dan Python memproses gambar atau video sesuai dengan tujuannya masing-masing. Kamera menangkap gambar dan komputer memprosesnya. Sebenarnya OpenCV bisa digunakan di bahasa C dan bahasa Java selain Python, tapi Python adalah bahasa yang paling sederhana diantara ketiga bahasa tersebut.

## BAB III METODOLOGI PENELITIAN

### 3.1 Waktu Dan Tempat Penelitian

#### 3.1.1 Tempat Penelitian

Penelitian ini akan dilaksanakan di CV. Angkasa Mobie Tech yang beralamat di Jalan Sultan Serdang, Gg. Ikhlas No.5.

#### 3.1.2 Waktu Penelitian

Adapun jadwal penelitian yang akan dilaksanakan adalah sebagai berikut:

Tabel 3.1 Jadwal Penelitian

| No | Kegiatan Penelitian   | Bulan Ke |   |   |   |    |   |   |   |     |   |   |   |  |  |  |  |
|----|---|----------|---|---|---|----|---|---|---|-----|---|---|---|--|--|--|--|
|    |   | I        |   |   |   | II |   |   |   | III |   |   |   |  |  |  |  |
|    |   | 1        | 2 | 3 | 4 | 1  | 2 | 3 | 4 | 1   | 2 | 3 | 4 |  |  |  |  |
| 1  | Studi Literatur   |          |   |   |   |    |   |   |   |     |   |   |   |  |  |  |  |
| 2  | Mengenal Permasalahan   |          |   |   |   |    |   |   |   |     |   |   |   |  |  |  |  |
| 3  | Perancangan Desain Alat   |          |   |   |   |    |   |   |   |     |   |   |   |  |  |  |  |
| 4  | Mempersiapkan Alat dan Bahan                                      |          |   |   |   |    |   |   |   |     |   |   |   |  |  |  |  |
| 5  | Pembuatan Alat  |          |   |   |   |    |   |   |   |     |   |   |   |  |  |  |  |
| 6  | Pembuatan serta Simulasi Program Ladder Diagram dan OpenCV-Python |          |   |   |   |    |   |   |   |     |   |   |   |  |  |  |  |
| 7  | Pengujian Alat  |          |   |   |   |    |   |   |   |     |   |   |   |  |  |  |  |
| 8  | Pengambilan Data  |          |   |   |   |    |   |   |   |     |   |   |   |  |  |  |  |
| 9  | Penyusunan Laporan  |          |   |   |   |    |   |   |   |     |   |   |   |  |  |  |  |

### 3.2 Tahapan dan Metodologi Penelitian

Dalam melaksanakan penelitian ini, maka diperlukan metode dan tahapan-tahapan yang terstruktur supaya tujuan daripada penelitian dapat berjalan dengan lancar dan tercapai.

## 1. Studi Literatur

Studi literatur dilakukan dengan mengumpulkan informasi dari berbagai literatur baik melalui buku-buku, jurnal artikel, website dan laporan yang berkaitan dengan *traffic light*, outseal PLC, dan OpenCV-Python.

## 2. Mengenali Permasalahan

Pada bagian ini merupakan bagian yang sangat penting karena merupakan dasar mengapa peneliti melakukan penelitian tentang *smart traffic light*. Penelitian ini didasarkan pada keadaan yang di alami oleh semua pengguna jalan raya yang sering mengalami kemacetan akibat *traffic light* yang hidup secara bergantian terlebih pada pagi dan sore hari yang merupakan puncak arus lalu lintas. Sehingga, arus lalu lintas mengalami pemadatan yang sangat panjang.

## 3. Perancangan Desain Alat

Perancangan alat meliputi desain dan cara kerja alat agar dapat beroperasi sesuai dengan yang diharapkan.

## 4. Mempersiapkan Bahan dan alat

Pada tahapan ini langkah mempersiapkan peralatan dan bahan merupakan hal yang harus dilakukan supaya alat yang dibuat dapat terselesaikan dengan baik dan tepat waktu serta tidak mengalami kendala ketika melakukan pembuatan alat.

## 5. Pembuatan Alat

Pada tahap ini saya melakukan proses pembuatan alat berdasarkan desain yang telah ditetapkan melalui konsep yang terdapat pada gambar desain alat sehingga hasil yang didapatkan benar atau akurat. Sistem yang dibuat adalah:

- a. Rangkaian adaptor tegangan 12 Volt
- b. Miniatur *traffic light* 3 simpang
- c. Rangkaian kamera webcam dengan outseal PLC

- d. Rangkaian sensor suara dengan outseal PLC
  - e. Rangkaian modul *traffic light* dengan outseal PLC
  - f. Rangkaian Modbus Atmega 32 dengan outseal PLC
6. Pembuatan serta simulasi program OpenCV-Python dan ladder diagram outseal PLC

Pada tahap ini saya melakukan pembuatan program OpenCV-Python menggunakan aplikasi visual studio code untuk dapat memproses gambar video yang ditangkap secara langsung oleh kamera webcam sehingga mampu menghitung jumlah kendaraan dan mengirim perintah ke outseal PLC untuk dapat di proses berdasarkan keadaan arus lalu lintas pada persimpangan. Pada bagian ini juga saya membuat program pada outseal PLC dengan bahasa pemrograman ladder diagram menggunakan aplikasi outseal studio dimana program ini mampu mengendalikan *traffic light* berdasarkan jumlah kepadatan lalu lintas serta keadaan darurat. Pada tahapan ini juga dilakukannya simulasi pada setiap program agar tidak terjadi nya kesalahan dan meminimalisir kerusakan pada outseal PLC dan komponen lainnya jika program tidak sesuai dengan yang diinginkan.

7. Pengujian Alat

Pada tahapan pengujian alat ini bertujuan untuk mengetahui apakah alat bekerja sesuai dengan yang diinginkan dengan berbagai parameter pengujian. Pada bagian ini alat diuji apakah mampu mengendalikan *traffic light* untuk dapat memprioritaskan simpang yang mengalami pematatan arus lalu lintas dan pada saat adanya kendaraan darurat. Pada tahapan ini juga akan dilakukannya pengujian terhadap sensor apakah bekerja dengan baik sehingga serta mengukur tingkat sensitivitas sensor. Dalam pengujian ini agar dapat terlaksana dengan baik, berikut kegiatan percobaan yang akan dilaksanakan antara lain :

- a. Pengujian alat dalam memprioritaskan simpang yang mengalami pematatan arus lalu lintas terlebih dahulu.



b. Pengujian sensor suara dalam mendeteksi suara sirine pada kendaraan darurat seperti mobil ambulance dan kendaraan khusus negara.

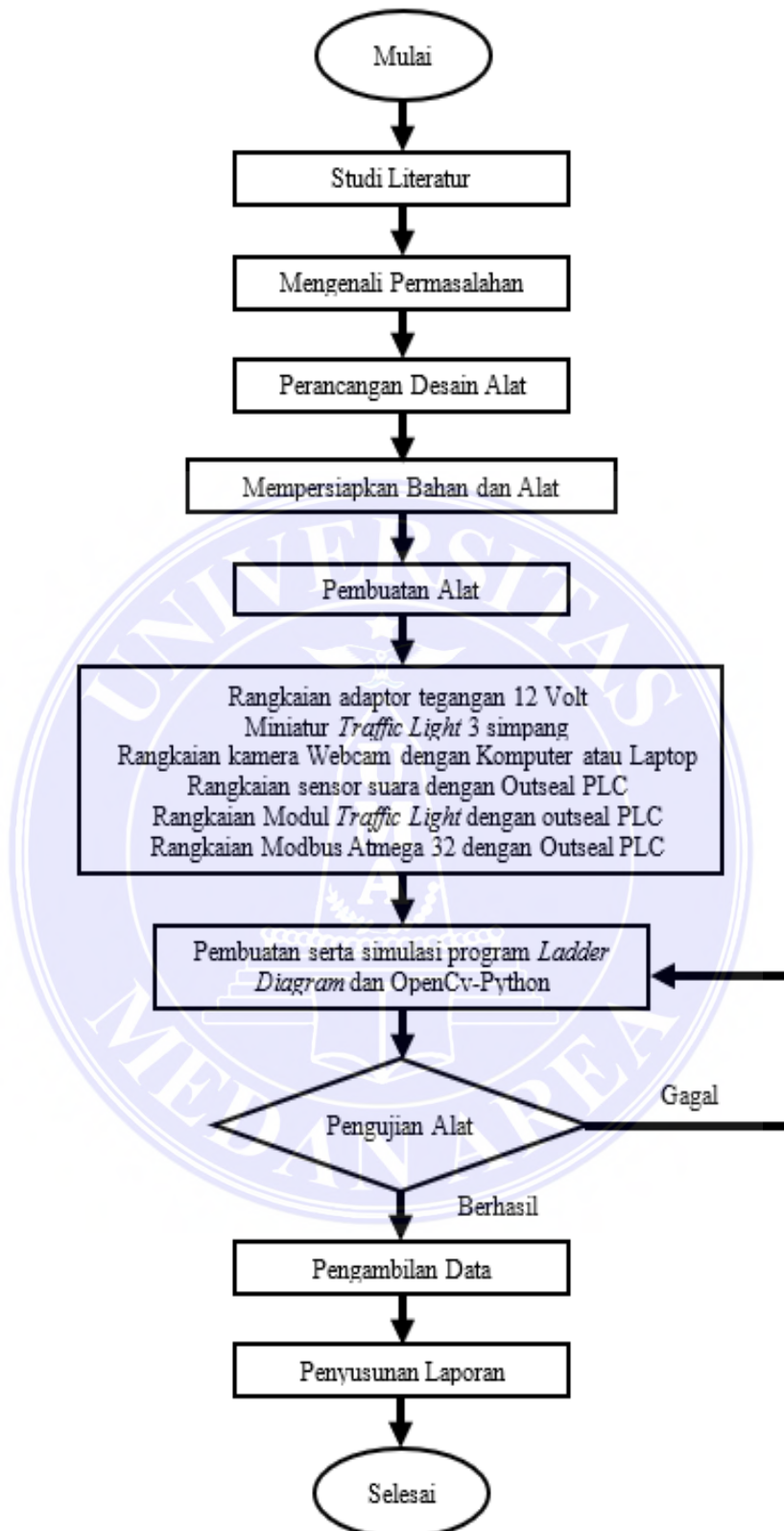
#### 8. Pengambilan Data

Pada bagian ini saya melakukan pengambilan data pada setiap uji coba berupa jumlah kendaraan yang terhitung oleh kamera webcam dan status *traffic light* pada setiap keadaan padat dan tidak padat serta pada keadaan ketika suara sirine kendaraan khusus melewati persimpangan.

#### 9. Penyusunan Laporan

Penulisan laporan merupakan tahap akhir dari rangkaian program penelitian ini, yang menyajikan hasil dan gagasan penelitian ini dalam bentuk skripsi dengan memperhatikan format penulisan yang benar sesuai kaidah penulisan yang berlaku di Fakultas Teknik.

Kerangka pemikiran diagram alir untuk melakukan penelitian ini ditunjukkan pada diagram di bawah ini. Diagram alir penelitian ini sebagai langkah-langkah yang peneliti lakukan untuk melakukan proses penelitian yang telah dijelaskan di atas.



Gambar 3.1 Diagram alir penelitian

### 3.3 Alat dan Bahan

Tentunya untuk mewujudkan salah satu tujuan dari penelitian ini yaitu merancang atau membuat lampu lalu lintas pintar berbasis outseal PLC diperlukan perangkat keras maupun perangkat lunak. Alat dan bahan yang digunakan pada tahap ini adalah gergaji, obeng, bor listrik, penggaris dan alat tulis. Alat yang digunakan untuk pengujian kelistrikan adalah multimeter digital. Selain alat-alat tersebut di atas, unsur yang paling penting tentunya bahan atau komponen elektronik yang digunakan dalam pelaksanaan rencana penelitian ini. Adapun bahan (komponen) yang akan digunakan adalah sebagai berikut :

Tabel 3.2 Tabel Alat dan Bahan dan Spesifikasinya

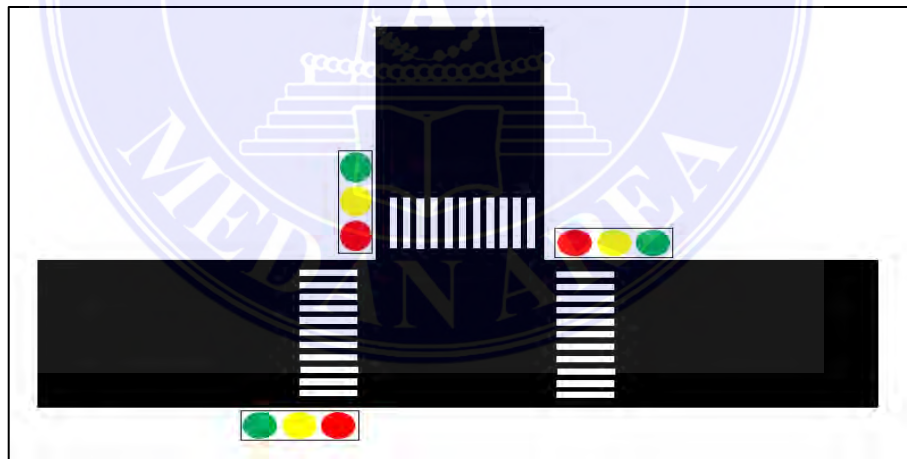
| No | Nama komponen           | Spesifikasi   | Jumlah |
|----|-------------------------|---|--------|
| 1  | Outseal PLC mega<br>V.3 | <ul style="list-style-type: none"> <li>- 16 digital input</li> <li>- 16 digital output</li> <li>- Analog Input, 2 jalur ( 0-5V/ 0-20mA)</li> <li>- MODBUS RTU protocol:               <ul style="list-style-type: none"> <li>- Onboard RS485, 2 jalur</li> <li>- Bluetooth, external module HC05/HC06</li> </ul> </li> <li>- WiFi, external module DT06</li> <li>- 1 Jalur komunikasi TWI/I2C</li> <li>- 1 Jalur komunikasi SPI</li> <li>- Flash Memory: 128kB</li> <li>- Voltase, 10-24V DC</li> </ul> | 1      |
| 2  | Webcam                  | <ul style="list-style-type: none"> <li>- Resolusi: 1920x1080</li> <li>- Tampilan sudut 150°</li> <li>- Jenis Antarmuka: USB</li> <li>- Sensor Gambar: 1080P CMOS, presisi tinggi dan tidak ada gambar yang terdistorsi.</li> </ul>  | 3      |

| No | Nama komponen                                | Spesifikasi  | Jumlah |
|----|--|--|--------|
|    |  | <ul style="list-style-type: none"> <li>- Keseimbangan putih otomatis, koreksi warna otomatis</li> <li>- Kecepatan bingkai: 30FPS</li> <li>- Mendukung Windows 2000 / XP / win7 / win8 / win10 Vista 32bit.</li> </ul>  |        |
| 3  | Sensor KY-037                                | <ul style="list-style-type: none"> <li>- Output : Analog (AO) dan Digital (DO)</li> <li>- Digital Output adalah output yang berupa sinyal digital yaitu LOW atau HIGH</li> <li>- Analog Output adalah output yang berupa sinyal analog yaitu angka 0 - 1000</li> <li>- Input : dihubungkan ke tegangan 3V - 5V</li> <li>- GND : dihubungkan ke Ground</li> </ul> | 3      |
| 4  | Sitem minimum AVR Atmega 32 (sebagai ModBus) | <ul style="list-style-type: none"> <li>- Konektivitas : I<sup>2</sup>C, SPI, UART /USART</li> <li>- Ukuran Inti :8-Bit</li> <li>- EEPROM : 1K x 8</li> <li>- Memori Program :32KB (16K x 16)</li> <li>- RAM Size :2K x 8</li> <li>- Kecepatan: 16MHz</li> </ul>  | 1      |
| 5  | Modul <i>traffic light</i> simulasi          | <ul style="list-style-type: none"> <li>- Tegangan kerja 5 Vdc</li> <li>- Terdiri dari 3 warna : merah, kuning dan hijau</li> </ul>   | 3      |
| 6  | Modul relay 8 channel                        | <ul style="list-style-type: none"> <li>- Tegangan kerja 5V</li> </ul>  | 2      |

| No | Nama komponen | Spesifikasi  | Jumlah |
|----|---------------|--|--------|
|    |               | <ul style="list-style-type: none"> <li>- 8-Channel Relay interface board, arus sink 15 mA - bisa langsung dari pin mikrokontroler.</li> <li>- Kapasitas relay, AC250V 10A ; DC30V 10A.</li> <li>- Rangkaian proteksi (isolasi, arus kickback) sudah termasuk di dalamnya - aman dan siap digunakan</li> <li>- LED indikator untuk menandakan channel yang aktif</li> </ul> |        |

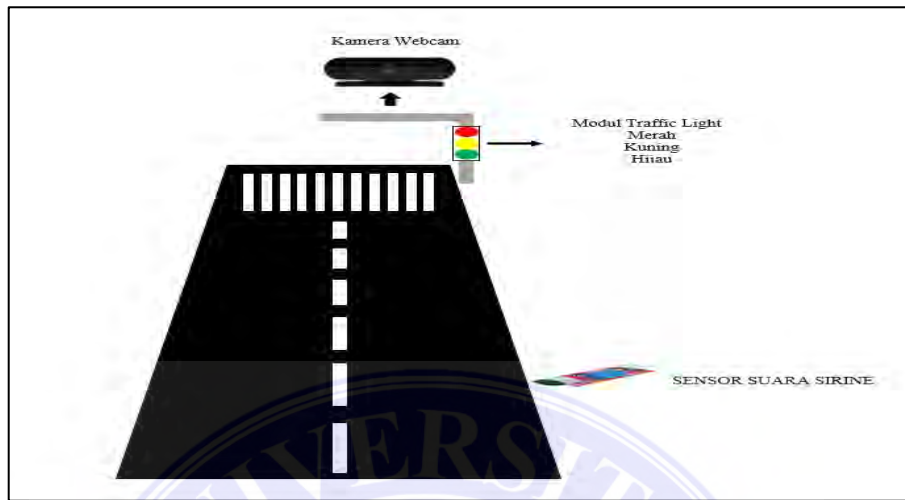
### 3.4 Desain Alat Miniatur *Traffic Light* dan Tata Letak Komponen

Berikut ini merupakan gambar desain miniatur *traffic light* 3 simpang dan tata letak lampu lalu lintas.



Gambar 3.2 Desain Miniatur *traffic light* 3 simpang tampak atas

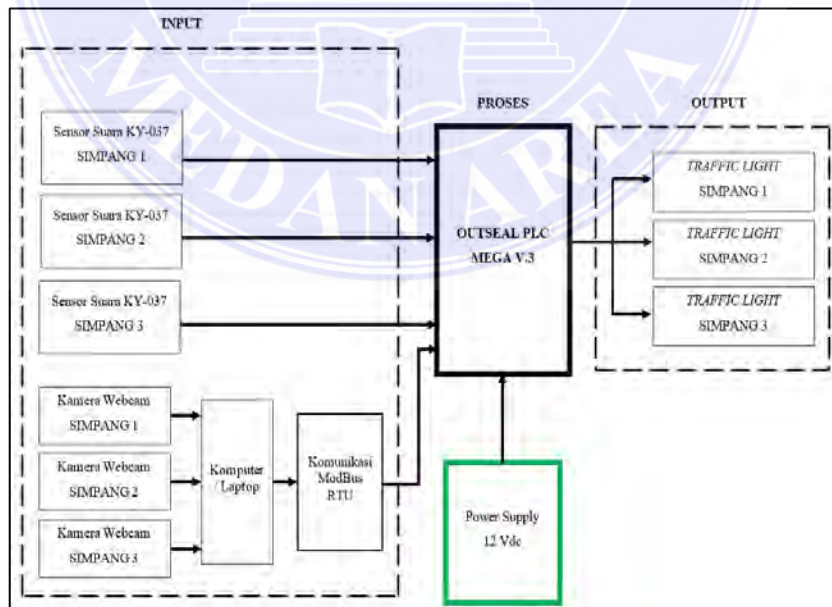
Berikut ini merupakan gambar desain maket *traffic light* 3 simpang tampak depan beserta tata letak sensor yang digunakan.



Gambar 3.3 Desain tampak depan *smart traffic light* beserta tata letak sensor

### 3.5 Diagram Blok Alat

Untuk memudahkan pemahaman tentang koneksi dan mekanisme, interaksi antara sistem yang dirancang digambarkan dalam bentuk diagram blok alat sebagai berikut :



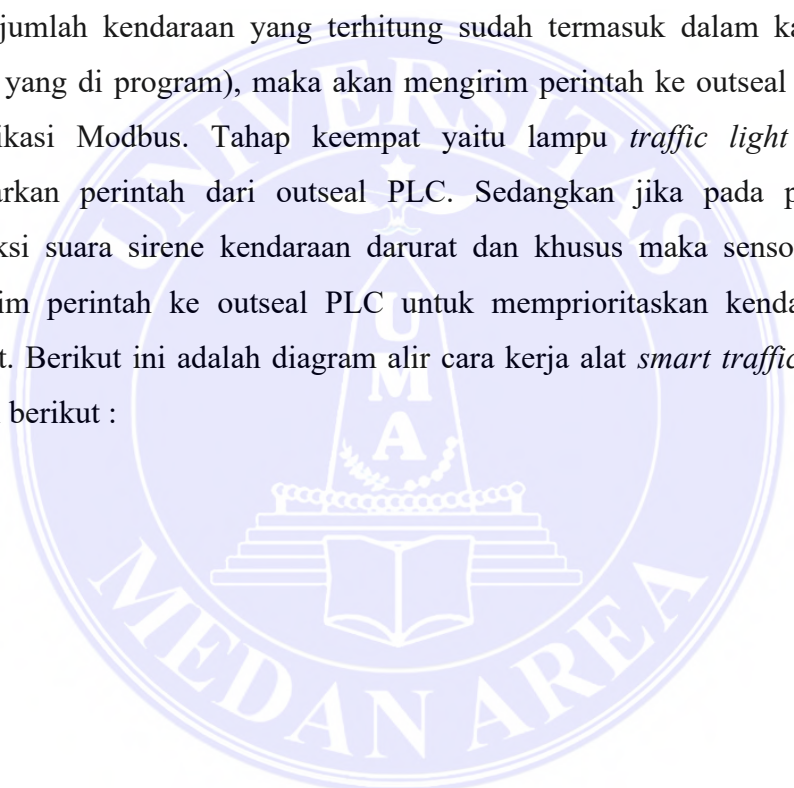
Gambar 3.4 Diagram blok alat

Dari diagram di atas dapat dijelaskan bagaimana kerja masing-masing sistem dikoordinasikan dengan yang lainnya menjadi satu kesatuan yang bekerja sama menjadi *smart traffic light* berbasis outseal PLC adalah sebagai berikut :

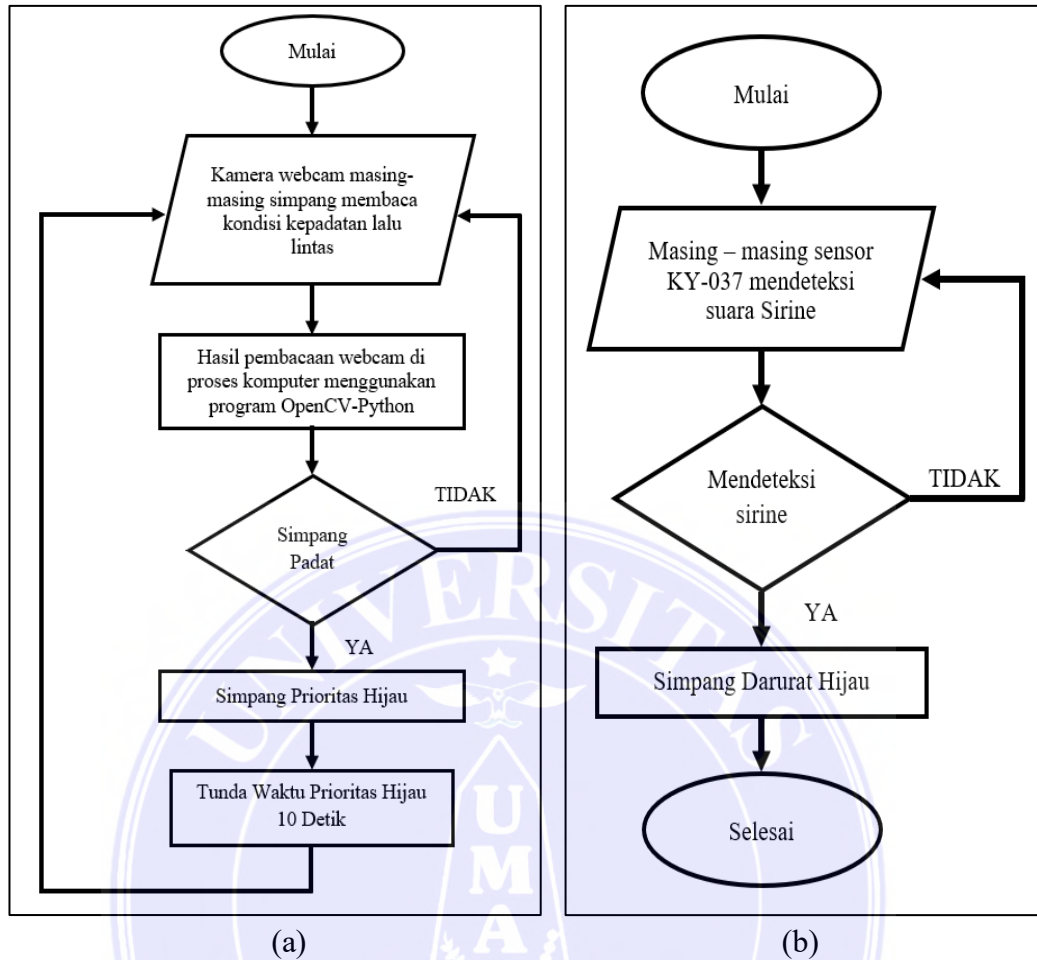
1. Sensor suara KY-037 berfungsi untuk mendeteksi suara sirine kendaraan darurat dan kendaraan khusus negara. Cara kerja sensor suara yaitu merubah besaran suara menjadi besaran listrik. Sinyal yang masuk akan di olah sehingga akan menghasilkan satu kondisi yaitu kondisi 1 atau 0. Sensor suara pada masing – masing simpang akan mendeteksi suara kendaraan yang menggunakan sirine dimana kendaraan tersebut berada.
2. Kamera webcam berfungsi untuk menangkap gambar video secara langsung kemudian di proses oleh komputer atau laptop untuk dapat menghitung jumlah kendaraan pada persimpangan tersebut.
3. Laptop atau komputer berfungsi untuk memproses gambar video yang di rekam oleh kamera webcam menggunakan program OpenCV-Python.
4. Modul *traffic light* warna merah, kuning, dan hijau berfungsi sebagai rambu isyarat lampu lalu lintas dimana pada setiap keadaan warna lampu mempunyai arti sebagai berikut :
  - a. Merah menandakan berhenti.
  - b. Kuning menandakan interval waktu sebagai sinyal untuk memulai berjalan atau mulai berhenti.
  - c. Hijau menandakan boleh mulai berjalan.
5. Outseal PLC berfungsi untuk mengatur atau mengontrol nyala (ON) atau tidak (OFF) perangkat lain (kontrol logika) yang tersambung dengan perangkat tersebut dan logika pengaturan tersebut dapat diubah-ubah (diprogram).
6. Komunikasi Modbus RTU berfungsi sebagai komunikasi antara PC dengan outseal. Modbus ini menggunakan mikrokontroller sistem minimum Atmega 32.

### 3.6 Diagram Alir Sistem Cara Kerja Alat

Dalam memudahkan pemahaman cara kerja alat, maka diagram alir ini dibuat untuk mempermudah memahami cara kerja alat secara sistematis. Pada gambar dibawah ini merupakan diagram alir cara kerja alat yaitu pada tahap pertama diawali dengan pengambilan gambar rekaman secara langsung menggunakan kamera webcam kemudian gambar tersebut dikirim ke komputer atau laptop menggunakan komunikasi kabel USB webcam itu sendiri. Kemudian tahap kedua hasil perekaman gambar video tersebut di proses oleh komputer atau laptop menggunakan program OpenCV-Python kemudian jumlah kendaraan dapat terhitung. Tahap ketiga yaitu ketika jumlah kendaraan yang terhitung sudah termasuk dalam kategori padat (sesuai yang di program), maka akan mengirim perintah ke outseal PLC melalui komunikasi Modbus. Tahap keempat yaitu lampu *traffic light* akan hidup berdasarkan perintah dari outseal PLC. Sedangkan jika pada persimpangan terdeteksi suara sirene kendaraan darurat dan khusus maka sensor suara akan mengirim perintah ke outseal PLC untuk memprioritaskan kendaraan khusus tersebut. Berikut ini adalah diagram alir cara kerja alat *smart traffic light* adalah sebagai berikut :







Gambar 3.5 Diagram alir cara kerja *smart traffic light*

(a). Diagram alir mendeteksi kepadatan kendaraan

(b). Diagram alir sensor KY-037 mendeteksi sirine

### 3.7 Pembuatan Perangkat Keras

Tahapan ini merupakan proses pembuatan alat perangkat keras. Sistem yang dibuat ada seperti yang dijelaskan dalam diagram alir penelitian di atas. Kemudian, saya akan menjelaskan cara membuat masing-masing sistem di bawah ini.

#### a. Rangkaian adaptor tegangan 12 Volt

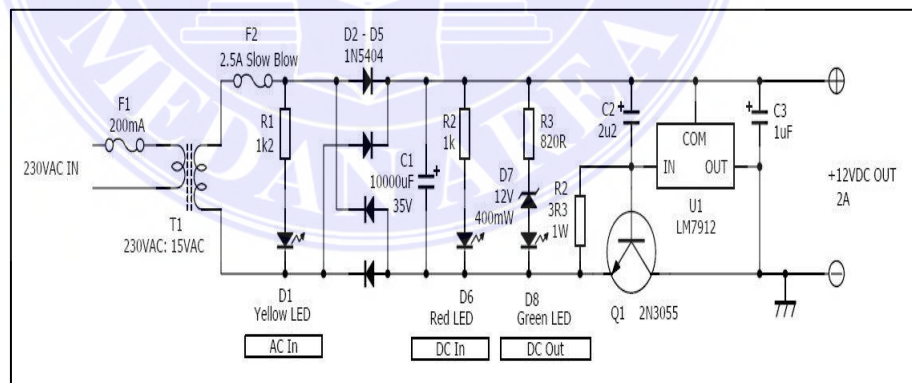
Adaptor adalah perangkat berupa rangkaian elektronik yang mengubah tegangan besar menjadi tegangan yang lebih kecil, atau rangkaian yang

mengubah arus bolak-balik menjadi arus searah. Berikut ini merupakan bentuk fisik adaptor 12 Vdc.



Gambar 3.6 Bentuk fisik adaptor tegangan 12 Vdc

Pada gambar diatas merupakan bentuk fisik adaptor tegangan 12 Vdc, dimana adaptor tersebut digunakan sebagai sumber listrik untuk mengoperasikan sistem *smart traffic light*. Agar mudah dalam memahami rangkaian dalam adaptor tersebut berikut merupakan skema daripada adaptor tegangan 12 Vdc.



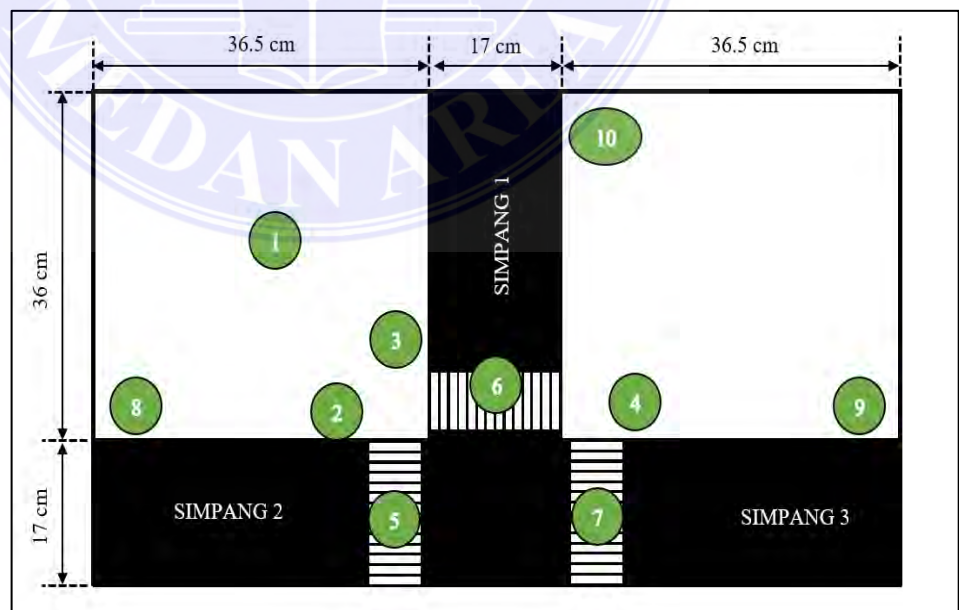
Gambar 3.7 Rangkaian skematik adaptor 12 Vdc

b. Miniatur *traffic light* 3 simpang

Miniatur *traffic light* terbuat dari papan triplek dengan ketebalam 8 mm yang dipotong dan diukur berdasarkan desain seperti pada gambar

dibawah ini. Berikut ini, penjelasan daripada gambar tersebut adalah sebagai berikut :

1. Tata letak outseal PLC sebagai kendali *traffic light*.
2. Tata letak modul *traffic* simpang 2 sebagai isyarat rambu lalu lintas.
3. Tata letak modul *traffic* simpang 1 sebagai isyarat rambu lalu lintas.
4. Tata letak modul *traffic* simpang 3 sebagai isyarat rambu lalu lintas.
5. Tata letak kamera webcam simpang 2 sebagai penghitung jumlah kendaraan.
6. Tata letak kamera webcam simpang 1 sebagai penghitung jumlah kendaraan.
7. Tata letak kamera webcam simpang 3 sebagai penghitung jumlah kendaraan.
8. Tata letak sensor suara KY-037 simpang 2 sebagai alat pendeteksi suara sirine kendaraan darurat dan khusus.
9. Tata letak sensor suara KY-037 simpang 3 sebagai alat pendeteksi suara sirine kendaraan darurat dan khusus.
10. Tata letak sensor suara KY-037 simpang 1 sebagai alat pendeteksi suara sirine kendaraan darurat dan khusus.



Gambar 3.8 Tata letak komponen

c. Rangkaian kamera webcam dengan komputer atau laptop

Kamera webcam berfungsi untuk merekam secara langsung kendaraan yang berada di persimpangan. Kemudian, hasil gambar yang di peroleh di kirim ke komputer atau laptop untuk di proses menggunakan program OpenCV-Python untuk mengetahui jumlah kendaraan. Agar dapat memahaminya, berikut hubungan rangkaian instalasi kamera webcam dengan komputer atau laptop, yaitu :



Gambar 3.9. Rangkaian instalasi kamera webcam dengan komputer atau laptop

d. Rangkaian sensor suara dengan outseal PLC

Dalam mendeteksi suara sirine kendaraan darurat dan khusus maka diperlukan sensor suara. Sensor suara yang digunakan dalam penelitian ini adalah sensor KY-037. Dalam penggunaannya, diperlukan komponen tambahan berupa modul relay 5 V dengan tujuan supaya dapat memberikan input 12 Vdc ke modul outseal PLC. Pada input modul outseal PLC hanya dapat membaca input bertegangan 12 Vdc.

1. Sensor suara KY-037 simpang 1

- Pin “DO“ di hubungkan ke pin “in1” modul relay
- Output relay “k1“ ke input “S4“ Outseal
- Pin “VCC“ di hubungkan ke sumber +5Vdc
- Pin “GND“ di hubungkan ke pin GND

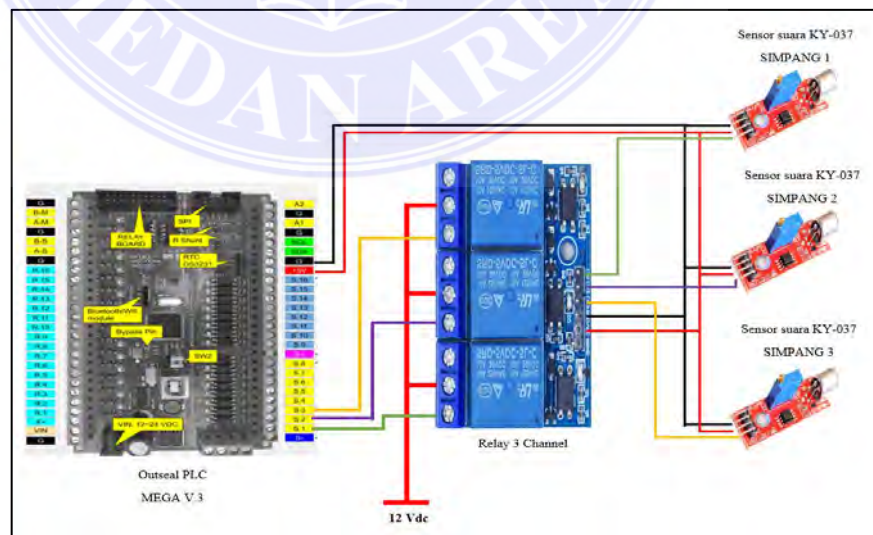
2. Sensor suara KY-037 simpang 2

- Pin “DO“ di hubungkan ke pin “in2” modul relay
- Output relay “k2“ ke input “S5“ Outseal
- Pin “VCC“ di hubungkan ke sumber +5Vdc
- Pin “GND“ di hubungkan ke pin GND

3. Sensor suara KY-037 simpang 3

- Pin “DO“ di hubungkan ke pin “in3” modul relay
- Output relay “k3“ ke input “S6“ Outseal
- Pin “VCC“ di hubungkan ke sumber +5Vdc
- Pin “GND“ di hubungkan ke pin GND

Berikut ini merupakan instalasi hubungan sensor suara dengan Outseal PLC, yaitu :



Gambar 3.10 Rangkaian sensor suara dengan Outseal PLC

e. Rangkaian modul *Traffic Light* dengan outseal PLC

Untuk rambu lalu lintas traffic light disini menggunakan modul led 3 warna yaitu merah, kuning, dan hijau dengan input tegangan 5 Vdc. Berikut ini intalasi hubungan modul traffic light dengan Outseal PLC.

1. Modul *traffic light* simpang 1

- Pin “R8“ Outseal ke “in1“ Relay lalu output “k1“ relay ke pin “R“ modul *traffic light*.
- Pin “R9“ Outseal ke “in2“ Relay lalu output “k2“ relay ke pin “Y“ modul *traffic light*.
- Pin “R10“ Outseal ke “in3“ Relay lalu output “k3“ relay ke pin “G“ modul *traffic light*.
- Pin “GND“ modul traffic light dan modul relay dihubungkan ke “GND“ Outseal.

2. Modul *traffic light* simpang 2

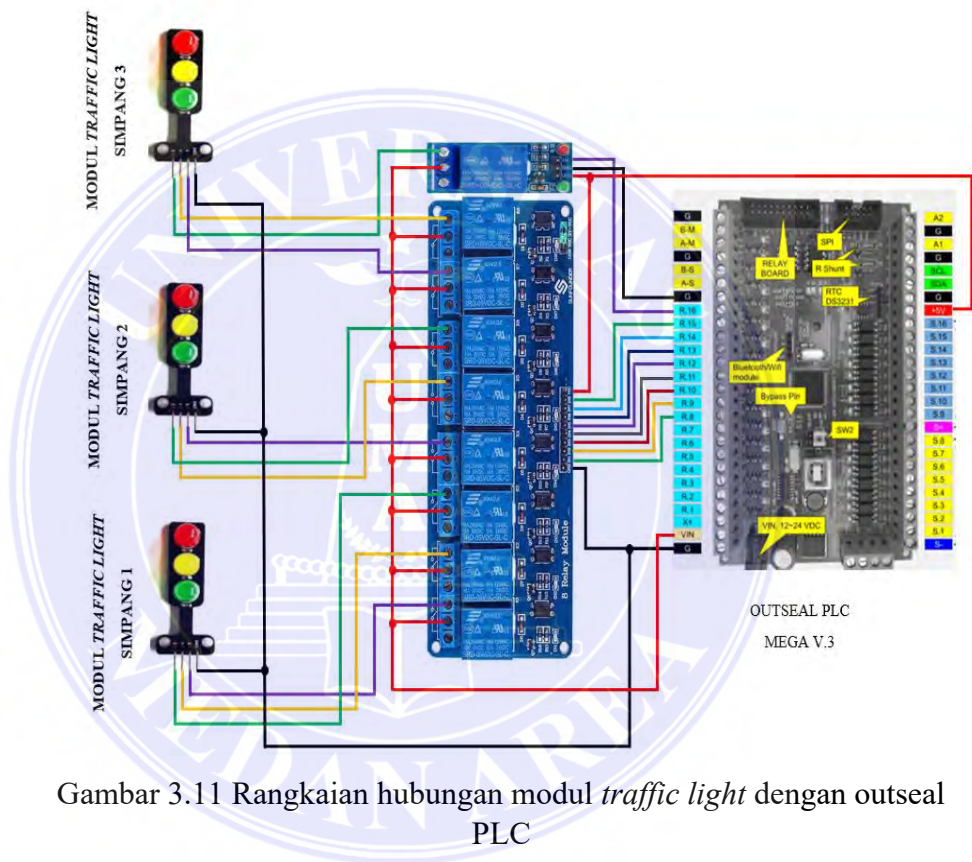
- Pin “R11“ Outseal ke “in4“ Relay lalu output “k4“ relay ke pin “R“ modul *traffic light*.
- Pin “R12“ Outseal ke “in5“ Relay lalu output “k5“ relay ke pin “Y“ modul *traffic light*.
- Pin “R13“ Outseal ke “in6“ Relay lalu output “k6“ relay ke pin “G“ modul *traffic light*.
- Pin “GND“ modul traffic light dan modul relay dihubungkan ke “GND“ Outseal.

3. Modul *traffic light* simpang 3

- Pin “R4“ Outseal ke “in7“ Relay lalu output “k7“ relay ke pin “R“ modul *traffic light*.
- Pin “R15“ Outseal ke “in8“ Relay lalu output “k8“ relay ke pin “Y“ modul *traffic light*.

- Pin “R16“ Outseal ke “in9“ Relay lalu output “k9“ relay ke pin “G“ modul *traffic light*.
- Pin “GND“ modul *traffic light* dan modul relay dihubungkan ke “GND“ Outseal.

Berikut ini merupakan gambar rangkaian hubungan dari ketiga modul *traffic light* tersebut, yaitu :

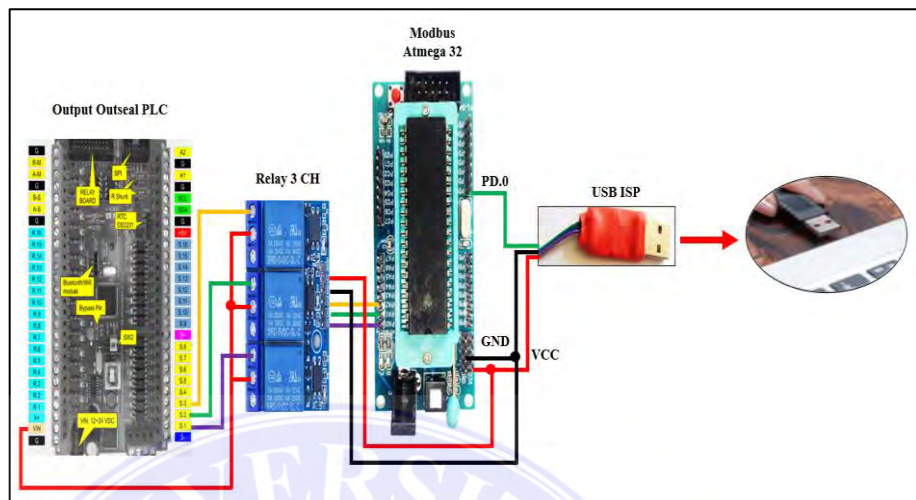


Gambar 3.11 Rangkaian hubungan modul *traffic light* dengan outseal PLC

#### 4. Rangkaian modbus atmega 32 dengan outseal PLC

Modbus atmega 32 ini berfungsi sebagai alat komunikasi PC dengan outseal PLC. Untuk input outseal PLC sendiri hanya bisa di proses apabila input “S1“ diberi tegangan 12Vdc, maka dari itu dibutuhkan relay yang mampu menjadi saklar input 12 Vdc untuk outseal PLC.

Berikut rangkaian komunikasi antara modbus atmega 32 dengan outseal PLC.



Gambar 3.12 Rangkaian komunikasi antara modbus atmega 32 dengan outseal PLC

### 3.8 Pembuatan Perangkat Lunak

Tahapan pembuatan perangkat lunak merupakan hal yang sangat penting untuk dilakukan supaya sistem *smart traffic light* dapat bekerja sesuai dengan yang di inginkan dan berjalan dengan baik tanpa ada kegagalan operasi. Dalam sistem *smart traffic light* ini menggunakan 2 bahasa pemrograman yakni bahasa pemrograman OpenCV-Python untuk memproses kendaraan miniatur supaya dapat menghitung jumlah kendaraan yang terekam oleh kamera secara langsung menggunakan aplikasi Visual Studio Code. Kemudian bahasa pemrograman ladder diagram untuk sistem kontrol outseal PLC menggunakan aplikasi outseal Studio V.3.6 yang berfungsi sebagai aplikasi edit program dan *compiler*.

#### 1. Program OpenCv-Python

```
# Python code for Multiple Color Detection
import numpy as np
import cv2
import serial

#outseal = serial.Serial('COM21', 9600)

batas_A = 21
```



```

batas_B = 21
batas_C = 21

webcam1 = cv2.VideoCapture(1)
webcam2 = cv2.VideoCapture(2)
webcam3 = cv2.VideoCapture(3)
# Start a while loop
while True:

    jumlah_merah_2=0
    jumlah_merah_3=0
    jumlah_merah=0
    jumlah_hijau_2=0
    jumlah_hijau_3=0
    jumlah_hijau=0
    jumlah_biru_2=0
    jumlah_biru_3=0
    jumlah_biru=0
    total_kendaraan_A=0
    total_kendaraan_B=0
    total_kendaraan_C=0

    ret, imageFrame = webcam1.read()
    ret, imageFrame2 = webcam2.read()
    ret, imageFrame3 = webcam3.read()
    img1 = cv2.resize(imageFrame,(560,480))
    img2 = cv2.resize(imageFrame2,(560,480))
    img3 = cv2.resize(imageFrame3,(560,480))
    hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)
    hsvFrame2 = cv2.cvtColor(imageFrame2, cv2.COLOR_BGR2HSV)
    hsvFrame3 = cv2.cvtColor(imageFrame3, cv2.COLOR_BGR2HSV)
    # Set range for red color and
    # define mask
    red_lower = np.array([136, 87, 111], np.uint8)
    red_upper = np.array([180, 255, 255], np.uint8)
    red_mask = cv2.inRange(hsvFrame, red_lower, red_upper)
    red_mask2 = cv2.inRange(hsvFrame2, red_lower, red_upper)
    red_mask3 = cv2.inRange(hsvFrame3, red_lower, red_upper)

    # Set range for green color and
    # define mask
    green_lower = np.array([25, 52, 72], np.uint8)
    green_upper = np.array([102, 255, 255], np.uint8)
    green_mask = cv2.inRange(hsvFrame, green_lower, green_upper)
    green_mask2 = cv2.inRange(hsvFrame2, green_lower, green_upper)
    green_mask3 = cv2.inRange(hsvFrame3, green_lower, green_upper)

```

```

# Set range for blue color and
# define mask
blue_lower = np.array([94, 80, 2], np.uint8)
blue_upper = np.array([120, 255, 255], np.uint8)
blue_mask = cv2.inRange(hsvFrame, blue_lower, blue_upper)
blue_mask2 = cv2.inRange(hsvFrame2, blue_lower, blue_upper)
blue_mask3 = cv2.inRange(hsvFrame3, blue_lower, blue_upper)

# Morphological Transform, Dilation
# for each color and bitwise_and operator
# between imageFrame and mask determines
# to detect only that particular color
kernel = np.ones((5, 5), "uint8")

# For red color
red_mask = cv2.dilate(red_mask, kernel)
red_mask2 = cv2.dilate(red_mask2, kernel)
red_mask3 = cv2.dilate(red_mask3, kernel)
res_red = cv2.bitwise_and(imageFrame, imageFrame,
                           mask = red_mask)
res_red2 = cv2.bitwise_and(imageFrame2, imageFrame2,
                            mask = red_mask2)
res_red3 = cv2.bitwise_and(imageFrame3, imageFrame3,
                            mask = red_mask3)

# For green color
green_mask = cv2.dilate(green_mask, kernel)
green_mask2 = cv2.dilate(green_mask2, kernel)
green_mask3 = cv2.dilate(green_mask3, kernel)
res_green = cv2.bitwise_and(imageFrame, imageFrame,
                             mask = green_mask)
res_green2 = cv2.bitwise_and(imageFrame2, imageFrame2,
                              mask = green_mask2)
res_green3 = cv2.bitwise_and(imageFrame3, imageFrame3,
                              mask = green_mask3)

# For blue color
blue_mask = cv2.dilate(blue_mask, kernel)
blue_mask2 = cv2.dilate(blue_mask2, kernel)
blue_mask3 = cv2.dilate(blue_mask3, kernel)
res_blue = cv2.bitwise_and(imageFrame, imageFrame,
                            mask = blue_mask)
res_blue2 = cv2.bitwise_and(imageFrame2, imageFrame2,
                             mask = blue_mask2)
res_blue3 = cv2.bitwise_and(imageFrame3, imageFrame3,
                             mask = blue_mask3)

```

```

mask = blue_mask3)

# Creating contour to track red color
contours, hierarchy = cv2.findContours(red_mask,
                                       cv2.RETR_TREE,
                                       cv2.CHAIN_APPROX_SIMPLE)
contours2, hierarchy2 = cv2.findContours(red_mask2,
                                       cv2.RETR_TREE,
                                       cv2.CHAIN_APPROX_SIMPLE)
contours3, hierarchy3 = cv2.findContours(red_mask3,
                                       cv2.RETR_TREE,
                                       cv2.CHAIN_APPROX_SIMPLE)
for pic, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if(area > 300):
        x, y, w, h = cv2.boundingRect(contour)
        img1 = cv2.rectangle(img1, (x, y),
                              (x + w, y + h),
                              (0, 0, 255), 2)
        jumlah_merah = jumlah_merah+1
        cv2.putText(img1, "Red Car", (x, y),
                    cv2.FONT_HERSHEY_SIMPLEX, 1.0,
                    (0, 0, 255))

for pic2, contour2 in enumerate(contours2):
    area2 = cv2.contourArea(contour2)
    if(area2 > 300):
        x2, y2, w2, h2 = cv2.boundingRect(contour2)
        img2 = cv2.rectangle(img2, (x2, y2),
                              (x2 + w2, y2 + h2),
                              (0, 0, 255), 2)
        jumlah_merah_2 = jumlah_merah_2+1
        cv2.putText(img2, "Red Car", (x2, y2),
                    cv2.FONT_HERSHEY_SIMPLEX, 1.0,
                    (0, 0, 255))

for pic3, contour3 in enumerate(contours3):
    area3 = cv2.contourArea(contour3)
    if(area3 > 300):
        x3, y3, w3, h3 = cv2.boundingRect(contour3)
        img3 = cv2.rectangle(img3, (x3, y3),
                              (x3 + w3, y3 + h3),
                              (0, 0, 255), 2)
        jumlah_merah_3 = jumlah_merah_3+1
        cv2.putText(img3, "Red Car", (x3, y3),

```

```
cv2.FONT_HERSHEY_SIMPLEX, 1.0,
(0, 0, 255))

# Creating contour to track green color
contours, hierarchy = cv2.findContours(green_mask,
                                       cv2.RETR_TREE,
                                       cv2.CHAIN_APPROX_SIMPLE)
contours2, hierarchy2 = cv2.findContours(green_mask2,
                                       cv2.RETR_TREE,
                                       cv2.CHAIN_APPROX_SIMPLE)
contours3, hierarchy3 = cv2.findContours(green_mask3,
                                       cv2.RETR_TREE,
                                       cv2.CHAIN_APPROX_SIMPLE)

for pic, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if(area > 300):
        x, y, w, h = cv2.boundingRect(contour)
        img1 = cv2.rectangle(img1, (x, y),
                              (x + w, y + h),
                              (0, 255, 0), 2)
        jumlah_hijau = jumlah_hijau+1
        cv2.putText(img1, "Green Car", (x, y),
                    cv2.FONT_HERSHEY_SIMPLEX,
                    1.0, (0, 255, 0))

for pic2, contour2 in enumerate(contours2):
    area2 = cv2.contourArea(contour2)
    if(area2 > 300):
        x2, y2, w2, h2 = cv2.boundingRect(contour2)
        img2 = cv2.rectangle(img2, (x2, y2),
                              (x2 + w2, y2 + h2),
                              (0, 255, 0), 2)
        jumlah_hijau_2 = jumlah_hijau_2+1
        cv2.putText(img2, "Green Car", (x2, y2),
                    cv2.FONT_HERSHEY_SIMPLEX,
                    1.0, (0, 255, 0))

for pic3, contour3 in enumerate(contours3):
    area3 = cv2.contourArea(contour3)
    if(area3 > 300):
        x3, y3, w3, h3 = cv2.boundingRect(contour3)
        img3 = cv2.rectangle(img3, (x3, y3),
                              (x3 + w3, y3 + h3),
                              (0, 255, 0), 2)
```

```

jumlah_hijau_3 = jumlah_hijau_3+1
cv2.putText(img3, "Green Car", (x3, y3),
            cv2.FONT_HERSHEY_SIMPLEX,
            1.0, (0, 255, 0))

# Creating contour to track blue color
contours, hierarchy = cv2.findContours(blue_mask,
                                       cv2.RETR_TREE,
                                       cv2.CHAIN_APPROX_SIMPLE)
contours2, hierarchy2 = cv2.findContours(blue_mask2,
                                       cv2.RETR_TREE,
                                       cv2.CHAIN_APPROX_SIMPLE)
contours3, hierarchy3 = cv2.findContours(blue_mask3,
                                       cv2.RETR_TREE,
                                       cv2.CHAIN_APPROX_SIMPLE)

for pic, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if(area > 300):
        x, y, w, h = cv2.boundingRect(contour)
        img1 = cv2.rectangle(img1, (x, y),
                              (x + w, y + h),
                              (255, 0, 0), 2)
        jumlah_biru = jumlah_biru+1
        cv2.putText(img1, "Blue Car", (x, y),
                    cv2.FONT_HERSHEY_SIMPLEX,
                    1.0, (255, 0, 0))

for pic2, contour2 in enumerate(contours2):
    area2 = cv2.contourArea(contour2)
    if(area2 > 300):
        x2, y2, w2, h2 = cv2.boundingRect(contour2)
        img2 = cv2.rectangle(img2, (x2, y2),
                              (x2 + w2, y2 + h2),
                              (255, 0, 0), 2)
        jumlah_biru_2 = jumlah_biru_2+1
        cv2.putText(img2, "Blue Car", (x2, y2),
                    cv2.FONT_HERSHEY_SIMPLEX,
                    1.0, (255, 0, 0))

for pic3, contour3 in enumerate(contours3):
    area3 = cv2.contourArea(contour3)
    if(area3 > 300):
        x3, y3, w3, h3 = cv2.boundingRect(contour3)
        img3 = cv2.rectangle(img3, (x3, y3),
                              (x3 + w3, y3 + h3),

```

```
(255, 0, 0), 2)
jumlah_biru_3 = jumlah_biru_3+1
cv2.putText(img3, "Blue Car", (x3, y3),
            cv2.FONT_HERSHEY_SIMPLEX,
            1.0, (255, 0, 0))

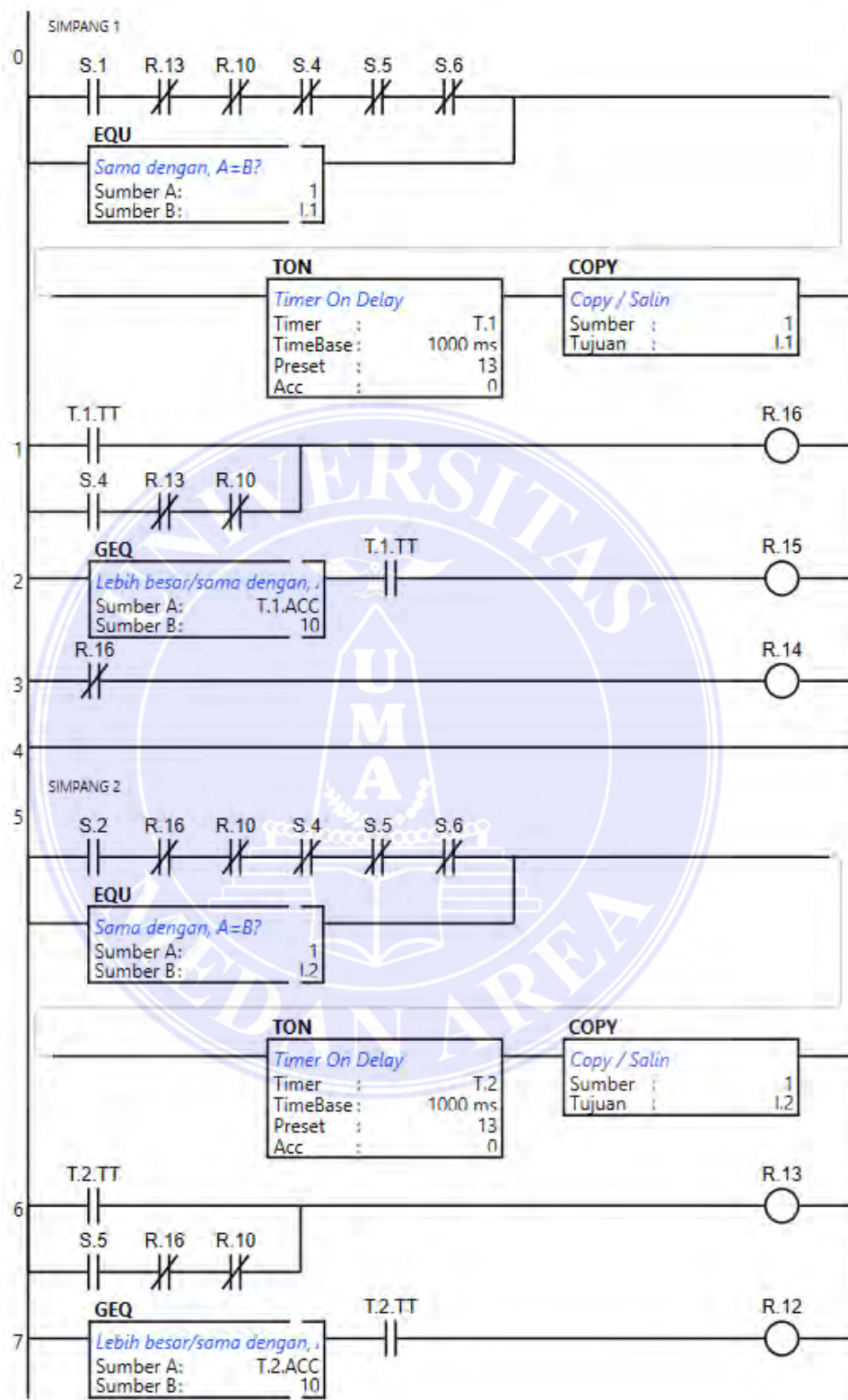
total_kendaraan_A = jumlah_merah+jumlah_biru+jumlah_hijau
total_kendaraan_B = jumlah_merah_2+jumlah_biru_2+jumlah_hijau_2
total_kendaraan_C = jumlah_merah_3+jumlah_biru_3+jumlah_hijau_3

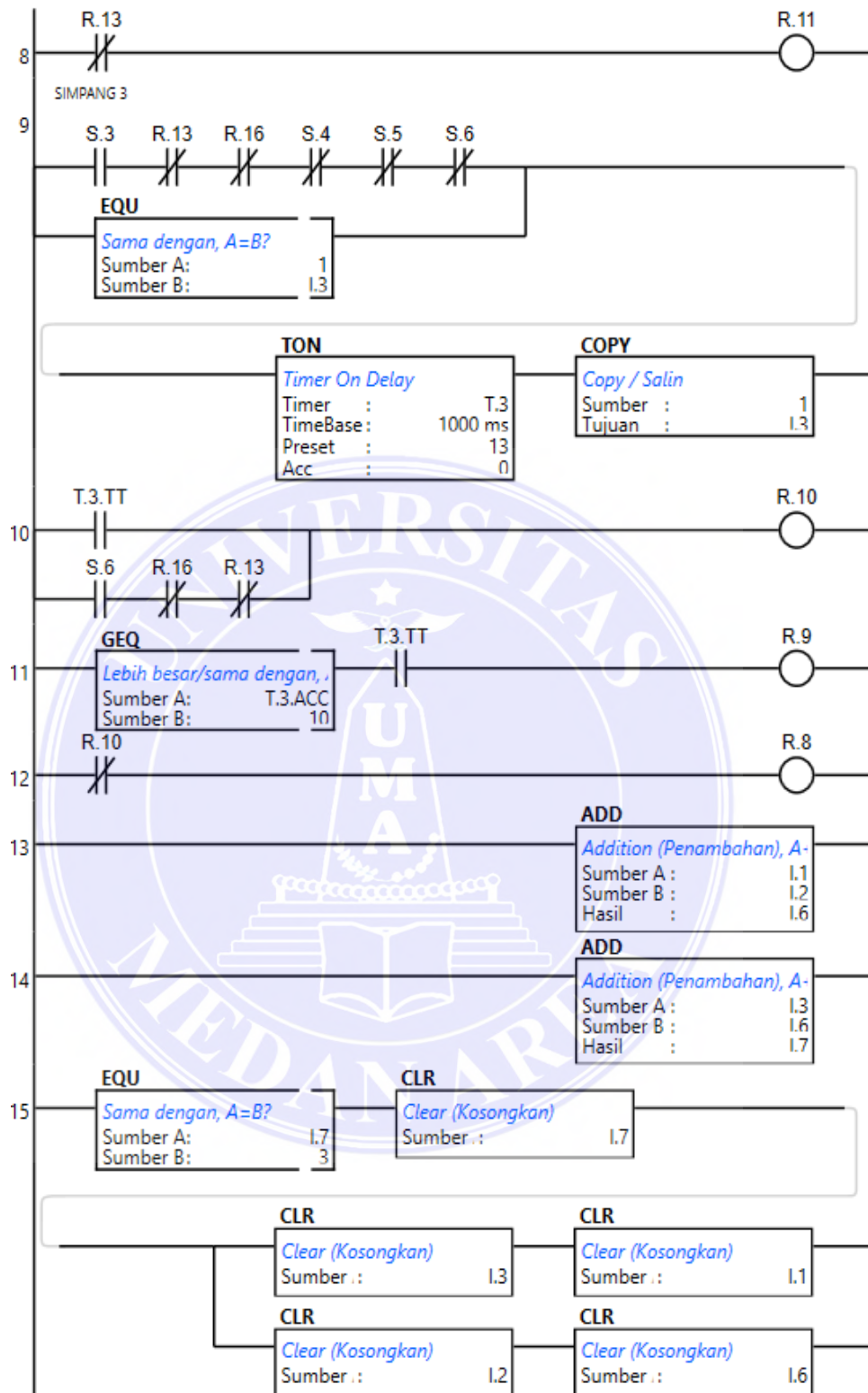
if(total_kendaraan_A > batas_A):
    #outseal.write(bytes("1", 'utf-8'))
    print("Simpang A Jalan")
if(total_kendaraan_B > batas_B):
    #outseal.write(bytes("2", 'utf-8'))
    print("Simpang B Jalan")
if(total_kendaraan_C > batas_C):
    # outseal.write(bytes("3", 'utf-8'))
    print("Simpang C Jalan")

print("Simpang B = "+str(total_kendaraan_A))
print("Simpang B = "+str(total_kendaraan_B))
print("Simpang B = "+str(total_kendaraan_C))

cv2.imshow("Simpang A", img1)
cv2.imshow("Simpang B", img2)
cv2.imshow("Simpang C", img3)
if cv2.waitKey(10) & 0xFF == ord('q'):
    cv2.destroyAllWindows()
    break
# Program Termination
```

## 2. Program Ladder diagram Outseal Studio







3. Program sistem minimum atmega 32 sebagai modbus

```
$regfile = "m32def.dat"
```

```
$crystal = 12000000
```

```
$baud = 9600
```

```
Config Porta.0 = Output
```

```
Config Porta.1 = Output
```

```
Config Porta.2 = Output
```

```
Dim Masuk As String * 1
```

```
Porta.0 = 1
```

```
Porta.1 = 1
```

```
Porta.2 = 1
```

```
Do
```

```
  Masuk = Inkey()
```

```
  If Masuk = "1" Then
```

```
    Porta.0 = 0
```

```
    Porta.1 = 1
```

```
    Porta.2 = 1
```

```
  End If
```

```
  If Masuk = "2" Then
```

```
    Porta.0 = 1
```

```
    Porta.1 = 0
```

```
    Porta.2 = 1
```

```
  End If
```

```
  If Masuk = "3" Then
```

```
    Porta.0 = 1
```

```
    Porta.1 = 1
```

```
    Porta.2 = 0
```

```
  End If
```

```
Loop
```

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

*Traffic light* adalah alat pemberi isyarat lalu lintas atau APILL yang berfungsi mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyeberangan pejalan kaki (*zebra cross*), dan tempat arus lalu lintas lainnya.

Pada penelitian ini membuat sebuah alat *smart traffic light* yang mampu memprioritaskan kendaraan yang mengalami pemadatan arus lalu lintas dan kendaraan darurat.

Pada akhir penelitian, terdapat beberapa kesimpulan yang penulis rangkum sebagai berikut :

1. Sistem *Smart traffic light* mampu mendahulukan simpang yang mengalami pemadatan kendaraan (kendaraan miniatur) dengan melakukan percobaan alat sebanyak 4 kali dan didapatkan hasil ketika kendaraan terhitung 3 maka persimpangan tersebut di prioritaskan.
2. Sistem *Smart Traffic Light* mampu memprioritaskan kendaraan darurat melalui sensor KY-037 dengan cara mendeteksi suara sirine kendaraan dengan melakukan percobaan sebanyak 6 kali dan didapatkan hasil persimpangan yang terdeteksi oleh sensor suara mampu memprioritaskan simpang tersebut meskipun persimpangan yang lain sedang jalan (lampu hijau).
3. Sistem *Smart Traffic Light* mampu melakukan monitoring kendaraan melalui kamera webcam yang terpasang pada masing-masing persimpangan.

#### 5.2 Saran

Dalam penelitian ini juga penulis mengharapkan adanya peningkatan tentang mengatasi kemacetan pada persimpangan lalu lintas. Berikut ini merupakan saran yang penulis inginkan antara lain :

1. Untuk pengembangan lebih lanjut mengenai kendaraan darurat agar dapat di prioritaskan dapat menggunakan GPS pada kendaraan darurat sehingga ketika mendekati persimpangan langsung dapat melewatinya.
2. Untuk pengaplikasian secara langsung harus melakukan uji coba kamera yang memiliki kemampuan yang bisa menangkap gambar pada malam hari.



## DAFTAR PUSTAKA

- Anas, F.H., Sumaryo, S. dan Estananto, E. (2019) “Desain Dan Perancangan Prototype Lampu Lalu Lintas Untuk Mengatasi Kepadatan Dan Keadaan Darurat,” *eProceedings of Engineering*, 6(2), hal. 2668–2676.
- Bakhtiar, A. (2019) *Panduan Dasar Outseal PLC*, Agung Bakhtiar.
- Fariz Elazar Ahmad dan Endah Fitriani (2020) “Penggunaan Sistem Outseal PLC Pada Pemilah Otomatis Dan Penghitung Otomatis,” *Bina Darma Conferenceon Engineering Science*, 2(2), hal. 27–39.
- Michael, D. dan Gustina, D. (2019) “Rancang Bangun Prototype Monitoring Kapasitas Air Pada Kolam Ikan Secara Otomatis Dengan Menggunakan Mikrokontroler Arduino,” *IKRA-ITH Informatika*, 3(2), hal. 59–66.
- Muhammad, A.A., Munadi, R. dan Mayasari, R. (2019) “Perancangan Lampu Lalu Lintas Pintar Untuk Smart City Menggunakan Wireless Sensor Network,” *eProceedings*, 6(1), hal. 1010–1017.
- Nursalim (2022) “Desain Sistem Kontrol Penanggulangan Kemacetan Lalu Lintas Adaptif Berbasis Plc,” *Jurnal Media Elektro*, XI(1), hal. 70–75.
- Rosyady, P.A., Ikhsan, Z.A. dan Feter, M.R. (2022) “Prototype Lampu Lalu Lintas Adaptif Berdasarkan Panjang Antrian Kendaraan Berbasis Arduino Uno,” *CIRCUIT: Jurnal Ilmiah Pendidikan Teknik Elektro*, 6(2), hal. 173–186.
- Setiawan, I. (2022) “Mengatasi Kemacetan Di Lampu Merah Dengan Pendekatan Image Processing,” *Journal of Innovation And Future Technology (IFTECH)*, 4(2), hal. 9–18.
- Usman dkk., (2020) “Implementasi Pengenalan Jenis Pola Tapak Ban (Tread) Menggunakan Metode Local Binary Patterns,” *Jurnal Sistem Komputer dan Informatika (JSON)*, 1(3), hal. 189.
- Warnilah, A.I. dkk., (2022) “Program Aplikasi Pendeteksi Masker Dengan Menggunakan Algoritma Haarcascade,” *EVOLUSI: Jurnal Sains dan Manajemen*, 10(1), hal. 89–98.