

## Prototipe lampu lalu lintas menggunakan PLC dan SCADA berbasis *computer vision* dengan raspberry pi 4B

Anton Firmansyah<sup>1</sup>, Andri Suyadi<sup>2</sup>, Alif Akram Khalish<sup>3</sup>, Al-Farick Zulhanudin<sup>4</sup>, Syafrudin<sup>5</sup>  
e-mail: [anton\\_firmansyah@polsri.ac.id](mailto:anton_firmansyah@polsri.ac.id), [andri\\_suyadi@polsri.ac.id](mailto:andri_suyadi@polsri.ac.id), [alifakramkhalish@gmail.com](mailto:alifakramkhalish@gmail.com),  
[alfarick54@gmail.com](mailto:alfarick54@gmail.com), [rudi.kta14@gmail.com](mailto:rudi.kta14@gmail.com)

<sup>1,2,3,4,5</sup>Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya, Indonesia

### Informasi Artikel

#### Riwayat Artikel

Diterima 13 Oktober 2024  
Direvisi 7 April 2025  
Diterbitkan 26 April 2025

#### Kata kunci:

*Machine learning*  
*Programmable logic controller*  
(PLC)  
Raspberry Pi 4B  
*Smart traffic light*  
*Supervisory control data and*  
*acquisition* (SCADA)

#### Keywords:

*Machine Learning*  
*Programmable logic controller*  
(PLC)  
Raspberry Pi 4B  
*Smart traffic light*  
*Supervisory control data and*  
*acquisition* (SCADA)

#### Penulis Korespondensi:

Alif Akram Khalish,  
Jurusan Teknik Elektro,  
Politeknik Negeri Sriwijaya,  
Jl. Sersan Aning gang Panus No. 39, RT.4/RW.7, Depok, Pancoran Mas, Jawa Barat, Id, 16436  
Email: [alifakramkhalish@gmail.com](mailto:alifakramkhalish@gmail.com)

### ABSTRAK

Pertumbuhan populasi dan peningkatan jumlah kendaraan di wilayah perkotaan telah menimbulkan tantangan serius dalam manajemen lalu lintas, terutama di persimpangan yang sering mengalami kemacetan. Sistem lampu lalu lintas konvensional yang tidak mampu merespons kondisi lalu lintas secara *real-time* menyebabkan pengaturan durasi lampu yang tidak efisien, memperburuk kemacetan, meningkatkan emisi karbon, serta menyebabkan pemborosan bahan bakar. Seiring dengan perkembangan teknologi, *machine learning* digunakan untuk mengoptimalkan pengaturan lalu lintas secara adaptif. Dalam penelitian ini, dikembangkan sistem lampu lalu lintas cerdas berbasis *Programmable Logic Controller* (PLC) yang dikombinasikan dengan Raspberry Pi 4B sebagai pusat pemrosesan data. Sistem ini juga terintegrasi dengan *Smart Traffic Light*, yang memungkinkan pengaturan durasi lampu berdasarkan analisis data lalu lintas secara *real-time*. Selain itu, penerapan *Supervisory Control and Data Acquisition* (SCADA) memungkinkan pemantauan dan pengendalian sistem secara efisien melalui antarmuka pengguna. Dengan memanfaatkan teknologi ini, sistem diharapkan dapat meningkatkan efisiensi lalu lintas dan mengurangi kemacetan dengan penyesuaian otomatis berbasis data.

### ABSTRACT

*The growth of population and the increasing number of vehicles in urban areas have posed significant challenges in traffic management, especially at intersections that frequently experience congestion. Conventional traffic light systems that cannot respond to real-time traffic conditions result in inefficient timing adjustments, exacerbating congestion, increasing carbon emissions, and causing fuel wastage. With technological advancements, machine learning is utilized to optimize traffic control adaptively. This study develops an intelligent traffic light system based on a Programmable Logic Controller (PLC), integrated with a Raspberry Pi 4B as the data processing center. The system is also incorporated with Smart Traffic Light technology, enabling adaptive light duration adjustments based on real-time traffic data analysis. Additionally, the implementation of Supervisory Control and Data Acquisition (SCADA) allows efficient system monitoring and control through a user interface. By leveraging these technologies, the system is expected to enhance traffic efficiency and reduce congestion through automated data-driven adjustments.*

## 1. PENDAHULUAN

Pertumbuhan pesat populasi dan jumlah kendaraan di daerah perkotaan telah memicu tantangan besar dalam manajemen lalu lintas, terutama pada persimpangan yang sering kali menjadi titik kemacetan. Sebagaimana yang dijelaskan oleh Duysembayeva dkk., peningkatan jumlah kendaraan menciptakan perlambatan arus lalu lintas akibat keterbatasan sistem manajemen lalu lintas yang ada. [1] Sistem lampu lalu lintas konvensional tidak mampu merespons kondisi lalu lintas secara real-time. Hal ini mengakibatkan pengaturan durasi lampu yang tidak efisien, sehingga memperburuk kemacetan pada jam-jam sibuk seperti di pagi hari ketika sebagian besar orang memulai aktivitas mereka atau di sore hari ketika orang pulang dari pekerjaan mereka.

Saat ini, sebagian besar sistem lampu lalu lintas masih dianggap monoton karena menggunakan timer statis dalam sebagai pengaturannya. Ketika volume kendaraan berubah atau kemacetan terjadi, sistem ini tidak mampu menyesuaikan waktu dengan cepat, sehingga menyebabkan kendaraan berhenti lebih lama dari yang diperlukan. Ketidakefisienan ini juga meningkatkan emisi karbon dan pemborosan bahan bakar akibat kendaraan yang terlalu lama dalam kondisi idle. Oleh karena itu, untuk mengatasi permasalahan ini, perlu ditingkatkan sistem manajemen lampu lalu lintas yang ada salah satunya dengan mengembangkan sistem waktu lampu lalu lintas yang didasarkan pada jumlah kendaraan yang melewati persimpangan.

Sudah terdapat beberapa studi yang telah dilakukan untuk membahas sistem lampu lalu lintas. Yang pertama adalah sistem lalu lintas yang mengatur durasi tiap lampu berdasarkan kepadatan kendaraan tiap simpang menggunakan sensor inductive load [2]. Kekurangan dari sistem ini adalah instalasi loop induktif memerlukan biaya dan waktu yang tinggi karena membutuhkan penggalian jalan. Selain itu, loop ini rentan terhadap kerusakan akibat cuaca ekstrem dan pekerjaan jalan, yang meningkatkan biaya pemeliharaan. Loop induktif juga mungkin tidak efektif mendeteksi kendaraan kecil dan dapat terganggu oleh gangguan elektromagnetik, mengurangi akurasi deteksi. Sistem ini juga kurang adaptif terhadap perubahan pola lalu lintas yang dinamis. Penelitian kedua adalah desain dan simulasi sistem lampu lalu lintas menggunakan proses citra digital yang memanfaatkan software MATLAB. Hasil didapatkan dalam proses citra yang digunakan sebagai referensi tingkat kepadatan jalan di sistem kontrol lampu lintas [3]. Studi lainnya membahas tentang desain lampu lalu lintas pintar menggunakan PIR sensor dan Raspberry Pi. Penelitian ini berfokus pada desain lampu lalu lintas menggunakan LED untuk menunjukkan waktu LED berjalan [4]. Sementara itu, studi lainnya menggunakan beberapa sensor infra merah atau photoelectric dan PLC yang pembacaan sensornya memerangui pengaturan LED [5].

PLC merupakan perangkat yang digunakan untuk mengotomatisasi proses kontrol. Dalam konteks lampu lalu lintas, PLC dapat diprogram untuk mengatur durasi lampu hijau, kuning, dan merah berdasarkan data yang diterima dari sensor lalu lintas. PLC menawarkan keandalan dan fleksibilitas yang tinggi dalam pengaturan logika kontrol [6].

Integrasi antara PLC dan sensor memungkinkan sistem lampu lalu lintas beroperasi secara adaptif. Misalnya, jika sensor mendeteksi antrean kendaraan yang panjang, PLC dapat memperpanjang durasi lampu hijau untuk mengurangi kemacetan. Sistem ini juga dapat diintegrasikan dengan jaringan komunikasi untuk berbagi data dengan pusat kendali lalu lintas. Beberapa studi kasus telah menunjukkan keberhasilan penerapan sistem ini dalam mengurangi waktu tunggu kendaraan dan meningkatkan efisiensi lalu lintas, terlebih menggunakan PLC industri [7].

Sistem lampu lalu lintas konvensional umumnya beroperasi dengan waktu yang telah ditentukan sebelumnya (*fixed-time*) [8]. Sistem ini menggunakan timer untuk mengubah sinyal pada interval yang telah ditetapkan, tanpa mempertimbangkan pergerakan lalu lintas aktual. Meskipun biaya awal dan pemeliharannya relatif rendah, sistem *fixed-time* dapat menyebabkan penundaan yang tidak perlu ketika tidak ada lalu lintas di salah satu jalur. Sistem lain yang disebut *actuated* menggunakan sensor seperti inductive loop, plat tekanan, atau sensor video untuk mendeteksi keberadaan kendaraan dan menyesuaikan waktu sinyal berdasarkan pergerakan lalu lintas. Sistem ini lebih responsif terhadap perubahan kondisi lalu lintas tetapi memiliki biaya awal dan pemeliharaan yang lebih tinggi. Beberapa penelitian juga mengeksplorasi penggunaan Raspberry Pi dan computer vision untuk analisis lalu lintas real-time, dengan kamera yang menangkap gambar untuk menghitung kendaraan dan menyesuaikan waktu lampu [9]. Selain itu, integrasi Programmable Logic Controller (PLC) dan Supervisory Control and Data Acquisition (SCADA) juga diteliti untuk kontrol yang lebih andal dan pemantauan sistem [10]. PLC sering digunakan untuk kontrol langsung lampu lalu lintas karena keandalannya, sementara SCADA menyediakan antarmuka untuk pemantauan, kontrol manual, dan akuisisi data untuk memungkinkan pengawasan dan pemantauan proses fisik, baik secara lokal maupun jarak jauh [11].

Berdasarkan penjelasan di atas, terlihat bahwa kebanyakan studi menggunakan sensor untuk mendeteksi jumlah kendaraan. Oleh karena itu, dengan kemajuan teknologi seperti Programmable Logic Controller (PLC) dan Supervisory Control and Data Acquisition (SCADA) menawarkan solusi yang lebih adaptif dan responsif terhadap perubahan arus lalu lintas. PLC memungkinkan logika kontrol yang lebih canggih dan reliable, sedangkan SCADA memungkinkan pemantauan dan pengendalian dari jarak jauh,

memberikan fleksibilitas yang lebih besar dalam manajemen lalu lintas. Selain itu, integrasi computer vision memungkinkan sistem ini untuk mengumpulkan data visual dari kondisi lalu lintas dan menyesuaikan durasi lampu lalu lintas secara real-time, sesuai dengan data yang diterima. Sistem ini memanfaatkan kamera untuk menangkap kondisi lalu lintas dan Raspberry Pi sebagai pusat pemrosesan untuk mengimplementasikan algoritma computer vision dalam mendeteksi dan menghitung kendaraan. Informasi ini kemudian digunakan untuk menyesuaikan waktu lampu lalu lintas secara dinamis dengan mengirim data jumlah kendaraan PLC yang kemudian PLC mengatur lampu lalu lintas secara dinamis berdasarkan tipe kepadatan.

Dalam penelitian ini, model Single Shot MultiBox Detector (SSD) [12] digunakan untuk mendeteksi dan menghitung jumlah kendaraan secara real-time dari data visual yang ditangkap oleh kamera. Model SSD dipilih karena kemampuannya dalam melakukan deteksi objek dalam satu tahap, sehingga lebih cepat dibandingkan metode deteksi dua tahap. Tujuan utama penggunaan model SSD adalah untuk memberikan informasi yang akurat dan cepat mengenai kepadatan lalu lintas kepada sistem PLC. SSD bekerja dengan menggunakan jaringan saraf konvolusional (CNN) untuk mengekstrak fitur dari gambar input, kemudian menggunakan serangkaian kotak pembatas default (anchor boxes) untuk memprediksi kotak pembatas objek dan label kelasnya. Teknik multi-scale feature maps dan hard negative mining yang digunakan oleh SSD memungkinkan deteksi objek dengan ukuran yang berbeda-beda dan peningkatan akurasi deteksi.

Penelitian ini berfokus pada pengembangan sistem lampu lalu lintas cerdas berbasis PLC dan SCADA, yang juga memanfaatkan teknologi computer vision untuk mengidentifikasi pola lalu lintas secara visual dan menggunakan Raspberry Pi 4B sebagai pusat komputasi yang memproses data dari kamera. Inovasi ini bertujuan untuk menghasilkan keputusan yang lebih akurat dan responsif dalam mengatur lampu lalu lintas, sehingga diharapkan mampu mengurangi kemacetan dan meningkatkan efisiensi pengaturan lalu lintas. Meskipun penelitian yang ada telah mengeksplorasi sistem lampu lalu lintas cerdas menggunakan Raspberry Pi dan computer vision [13], integrasi teknologi-teknologi ini dengan kemampuan kontrol yang kuat dari PLC dan fitur pemantauan yang komprehensif dari SCADA masih merupakan area yang kurang dieksplorasi. Penelitian tentang PLC dan SCADA dalam konteks kontrol lalu lintas seringkali menggunakan jenis sensor lain seperti inductive loop atau RFID [3].

Kebaruan penelitian ini terletak pada integrasi machine learning dan computer vision dengan sistem berbasis PLC dan SCADA. Hal ini bertujuan untuk mengisi gap penelitian dengan menggabungkan keunggulan deteksi visual real-time dari computer vision yang didukung oleh Raspberry Pi 4B dengan keandalan kontrol industri dari PLC dan kemampuan pemantauan serta supervisi dari SCADA. Pada studi ini, terdapat 2 kondisi, yaitu padat dan sepi. Padat merupakan kondisi dimana terdapat lebih dari 3 kendaraan di persimpangan dan sepi merupakan kondisi terdapat kurang dari 3 mobil di persimpangan. Integrasi ini diharapkan tidak hanya membuat sistem lebih responsif terhadap perubahan kondisi lalu lintas secara visual, tetapi juga memberikan platform yang kuat dan terpercaya untuk implementasi sistem lampu lalu lintas cerdas di persimpangan jalan. Dengan demikian, penelitian ini berupaya untuk mengatasi kebutuhan akan sistem lampu lalu lintas cerdas yang tidak hanya adaptif tetapi juga andal dan mudah dipantau, yang sangat penting untuk manajemen lalu lintas yang efektif di daerah perkotaan.

## 2. METODE PENELITIAN

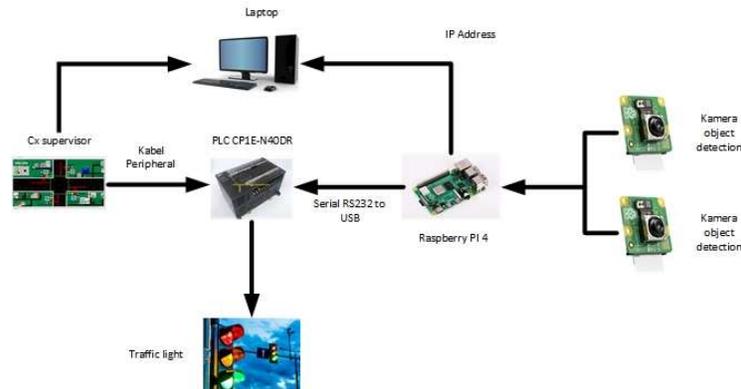
Penelitian ini dilakukan dengan metode kuantitatif, dimana *dataset* didapatkan dengan memfoto setiap mobil dan data lalu lintas diukur secara real-time menggunakan kamera dan perangkat keras yang dirancang. Sistem *Traffic Light* ini diuji dalam bentuk prototipe skala kecil, dimana mobil di jalan raya disimulasikan menggunakan mobil mainan untuk memvalidasi kinerja sistem. Dengan pengambilan *dataset* secara manual, simulasi *Traffic Light* dalam bentuk prototipe, objek mobil sebagai mobil mainan, dapat dilakukan pengembangan model SSD mobileNet V2 menggunakan Raspberry Pi 4B, pengontrolan lampu lalu lintas menggunakan PLC Omron CP1-E N40, dan pemantauan lalu lintas secara *realtime* menggunakan SCADA CX-Supervisor. Proses tersebut akan menghasilkan simulasi sistem *Smart Traffic Light* yang dapat mengatur kerja waktu lampu lalu lintas berdasarkan jumlah kendaraan yang ada di jalan.

### 2.1 Diagram Blok

Blok diagram dari Prototipe Lampu Lalu Lintas Menggunakan PLC dan SCADA berbasis Computer Vision dengan Raspberry Pi 4B diperlihatkan pada Gambar 1. PLC Omron CP1E-N40DR berfungsi sebagai slave dalam sistem ini, yang dikendalikan oleh dua master utama, yaitu *Supervisory Control and Data Acquisition* (SCADA) dan Raspberry Pi 4B. Tugas utama PLC adalah mengontrol pengaturan lampu lalu lintas, mengaktifkan dan menonaktifkan sinyal lampu hijau, kuning, dan merah berdasarkan data yang diterima dari Raspberry Pi 4B. Sistem ini menggunakan protokol komunikasi Modbus RTU melalui koneksi Serial RS232 to USB untuk pertukaran data antara PLC dan Raspberry Pi 4B.

Sistem SCADA yang dioperasikan melalui CX Supervisor bertindak sebagai master untuk memonitoring dan mengendalikan PLC. SCADA memfasilitasi pemantauan secara real-time terhadap kondisi

lampu lalu lintas di persimpangan, serta memberikan kemampuan kontrol manual jika diperlukan oleh operator. Komunikasi antara SCADA dan PLC dilakukan melalui kabel perifer, memungkinkan SCADA mengakses status sistem dan mengubah pengaturan lampu lalu lintas sesuai kebutuhan.



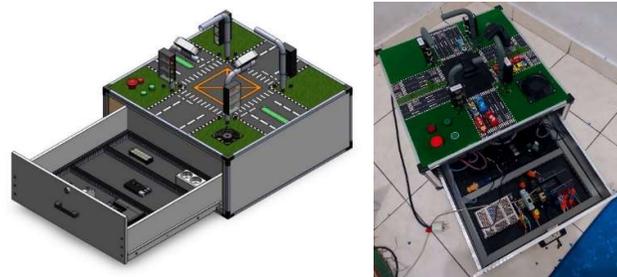
Gambar 1. Diagram Blok *Smart Traffic Light*

Raspberry Pi 4B bertindak sebagai master kedua, yang bertanggung jawab untuk pengolahan gambar dari dua kamera yang dipasang di simpang Timur dan Selatan. Kamera ini berfungsi untuk menangkap kondisi lalu lintas secara real-time protipe lalu lintas, dan mendeteksi objek kendaraan menggunakan algoritma SSD MobileNet V2. Algoritma ini mendeteksi jumlah kendaraan pada setiap frame gambar yang diambil oleh kamera, dan data tersebut kemudian dianalisis oleh Raspberry Pi. Hasil analisis dikirimkan ke PLC melalui komunikasi Modbus RTU.

Hasil deteksi kendaraan oleh Raspberry Pi 4B ditampilkan melalui web server sebagai bagian dari sistem pemantauan. Tampilan ini memungkinkan operator untuk melihat kondisi lalu lintas secara visual dan memastikan bahwa sistem bekerja sesuai dengan data real-time dari kamera. Implementasi web server ini juga mendukung pengambilan keputusan yang lebih cepat dan efisien dalam situasi kritis, seperti ketika terjadi penumpukan kendaraan di salah satu jalur.

**2.2 Perancangan Mekanik**

Hasil desain 3D dan hasil prototipe diperlihatkan pada Gambar 2.



(a) Desain 3D Prototipe *Smart Traffic Light*

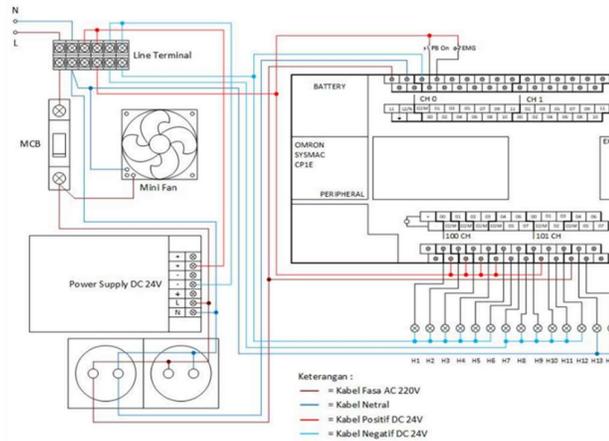
(b) Protipe *Smart Traffic Light*

Gambar 2. Hasil Perancangan Mekanik

**2.3 Perancangan Elektrikal**

Diagram pengkabelan mencakup skema koneksi untuk sistem PLC Omron CP1E-N40SDR-A. Skema ini mencakup komponen input *Programmable Logic Controller* (PLC) yang mengontrol fungsi lampu smart traffic light serta komponen output PLC yang diatur oleh input dari perangkat PLC.

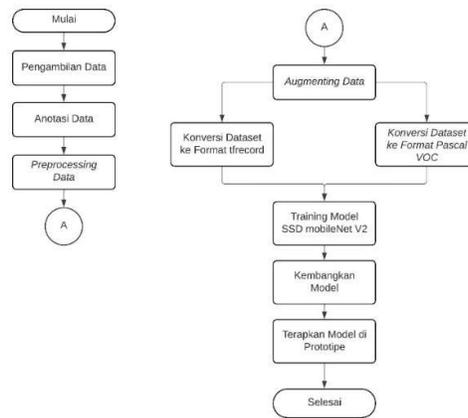
Gambar 3. meliputi jalur pengkabelan *Power Supply* DC 24B, *Mini Fan*, *Miniature Circuit Breaker* (MCB), dan *input* dan *output* yang terhubung ke PLC.



Gambar 3. Skematik PLC pada Lampu Lalu Lintas

**2.4 Proses Penerapan dan Pengembangan Model SSD MobileNet V2**

Pengembangan Model SSD MobileNet V2 harus melalui tahapan-tahapan tertentu terlebih dahulu. Tahapan tersebut digambarkan dalam diagram alir pada Gambar 4.



Gambar 4. Diagram Alir Penerapan dan Pengembangan Model SSD MobileNetV2

**2.4.1 Pengambilan Data**

Pengambilan data dilakukan secara mandiri sesuai dengan objek yang akan dideteksi, yaitu mobil mainan. Pengambilan foto untuk *dataset* akan sangat berpengaruh dalam akurasi dan performa model saat dijalankan nanti. Oleh karena itu, pengambilan foto *dataset* dilakukan sama seperti pengambilan *frame* pada kamera di *traffic light*.

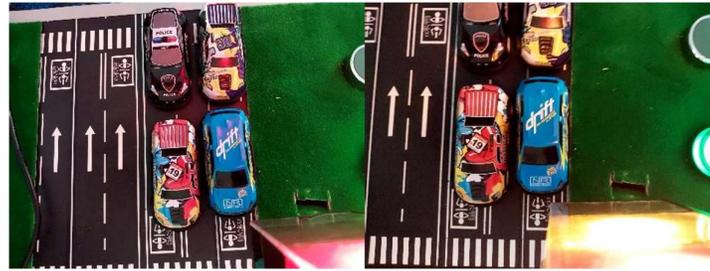
Porsi pengambilan gambar diawali dengan 1 mobil dengan 7 kondisi, yaitu kondisi lampu merah, lampu kuning, lampu hijau, minim cahaya, cahaya normal, diperbesar, dan tidak seluruh bagian mobil tampil. Masing-masing kondisi tersebut difoto sebanyak 20 kali. Selain itu terdapat 6 jenis motif mobil. Berikut merupakan Tabel pengambilan *dataset*.

Tabel 1. Jumlah Foto terhadap kendaraan Mobil

Mobil	Jumlah Foto
0	50
1	840
2	140
3	140
4	554

Pengambilan foto terbanyak ada saat jumlah mobil 1 dikarenakan model harus mempelajari motif dan kondisi tiap mobil. Jumlah terbanyak kedua adalah saat jumlah mobil 4 karena model harus mempelajari gambar saat mobil dalam kondisi berhimpitan. Foto dengan kondisi tanpa mobil juga diberikan agar model mempelajari bahwa gambar tersebut merupakan kondisi tanpa mobil atau hanya gambar latar belakang saja.

Gambar 6 merupakan contoh hasil pengambilan foto secara mandiri dengan menyesuaikan sudut kamera seperti sudut kamera yang dipasang pada tiang lalu lintas.



(a) 4 Buah Mobil saat Lampu Merah (b) 4 Buah mobil saat Lampu Kuning



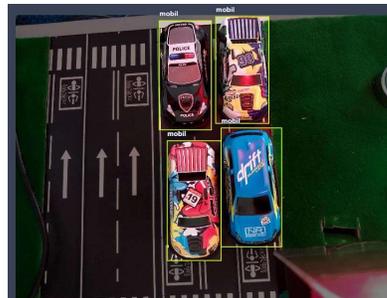
(c) 4 Buah Mobil saat Lampu Hijau (d) 4 Buah mobil saat Kondisi Diperbesar

Gambar 5. Contoh Dataset

Pengambilan foto *dataset* juga dilakukan berdasarkan kondisi pencahayaan seperti saat kondisi lampu merah, kuning, hijau, dan keadaan diperbesar.

**2.4.2 Anotasi Data**

Pada proses penganotasian data, semua foto yang sudah diambil harus dikumpulkan dan dianotasi menggunakan Roboflow.



Gambar 6. Contoh Anotasi Data

*Bounding boxes* digambar pada tiap mobil di dalam suatu foto. *Bounding Boxes* dibuat seketat mungkin dengan objek agar *ground truth* sesuai dengan objek atau pembuatan kotak nanti sesuai dengan posisi objek saat model dijalankan.

**2.4.3 Preprocessing Data**

Tahap preprocessing atau prapemrosesan dilakukan agar performa model meningkat dan mempersingkat waktu pelatihan model SSD MobileNetV2. Semua *dataset* diubah ukurannya menjadi 640x480.

**2.4.4 Augmenting Data**

Augmentasi data merupakan teknik penting dalam proses pelatihan model, terutama dalam konteks *deep learning*. Teknik ini digunakan untuk meningkatkan variasi data pelatihan dengan menghasilkan gambar-gambar baru yang berasal dari gambar asli. Dalam proyek ini, augmentasi data diterapkan untuk memperkaya *dataset* mobil mainan dan meningkatkan kemampuan generalisasi model SSD MobileNet V2.

Pada tahap ini, penambahan foto baru dari foto yang telah ada dengan membalikkan arah foto, merubah kecerahan, saturasi, dan tingkat blur foto yang telah ada, kemudian mengubahnya ke file yang baru.

Setelah melakukan augmentasi, total *dataset* yang didapatkan adalah 4480 foto. Dengan 4134 foto latihan, 296 foto validasi, dan 49 foto tes.

#### 2.4.5 Konversi Format *Dataset*

Terdapat 2 format *dataset* yang harus didapatkan dari hasil keseluruhan *raw photo*, yaitu format tfrecord dan format Pascal VOC. Dalam proses konversi *dataset* untuk deteksi objek, format Pascal VOC dan TFRecord memainkan peran yang berbeda. Pascal VOC adalah format anotasi yang menggunakan file XML untuk menyimpan informasi kotak pembatas dan label kelas objek dalam gambar. Tujuannya adalah untuk menyediakan format standar yang mudah dibaca dan digunakan untuk menganotasi *dataset*. Di sisi lain, TFRecord adalah format penyimpanan biner yang dioptimalkan untuk TensorFlow. Tujuannya adalah untuk menyimpan data secara efisien, terutama untuk kumpulan data besar, dan mempercepat proses pelatihan model. Biasanya, *dataset* yang dianotasi dalam format Pascal VOC akan dikonversi ke format TFRecord untuk digunakan dalam pelatihan model TensorFlow. Konversi ini memungkinkan pemrosesan data yang lebih cepat dan efisien.

#### 2.4.6 Training Model SSD MobileNet V2

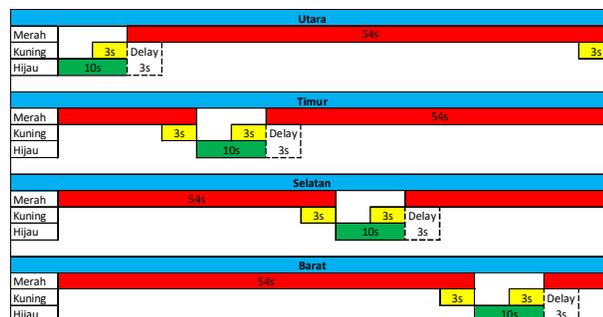
Dalam proses pelatihan model ini, penulis menggunakan model SSD MobileNetV2 dari tensorflow open source. Bobot data awal yang digunakan bukan data random, tetapi data pre-trained weights yang telah dilatih sebelumnya pada arsitektur SSD MobileNetV2 untuk suatu *dataset*. Untuk melakukan pendeteksian dan pengklasifikasian pada suatu *dataset* baru, diperlukan proses pelatihan *dataset* yang akan digunakan sehingga mendapatkan bobot baru yang sesuai dengan *dataset* yang dimiliki. Pelatihan *dataset* baru dengan menggunakan bobot awal pre-trained weights membuat proses pelatihan berjalan cepat.

#### 2.5 Protokol Komunikasi PLC dan Raspberry Pi 4B

PLC dapat mengirim dan menerima data dari PC (Personal Computer), laptop, atau Raspberry Pi dengan komunikasi *hostlink*, biasanya PLC menerima perintah untuk menulis atau membaca alamat di dalamnya. Terdapat aturan-aturan tersendiri dalam penulisan dan pembacaan data dari PLC OMRON CP1E-N40. Aturan-Aturan tersebut diatur dengan format *command block* dan *response block* atau biasa dikenal dengan C-Mode Command. *Command block* merupakan keadaan ketika PC menjadi host dan mengirimkan data ke PLC Format dari command block dan response block.

#### 2.6 Prinsip Kerja Sistem *Traffic Light*

Sistem Smart Traffic Light yang dikembangkan dalam penelitian ini mengintegrasikan algoritma Single Shot Detector (SSD) dan SCADA (CX-Supervisor) pada lampu lalu lintas yang dikendalikan oleh PLC (*Programmable Logic Controller*). Implementasi sistem ini difokuskan pada simulasi lalu lintas di persimpangan Rajawali. Dalam sistem ini, dua kamera dipasang pada jalur-jalur prioritas, yaitu jalur Timur-Selatan dan Selatan-Barat. Fungsi dari kamera ini adalah mendeteksi kepadatan lalu lintas di setiap jalur dan mengirimkan data jumlah kendaraan ke PLC melalui komunikasi serial RS232. Hasil deteksi jumlah kendaraan ini ditampilkan secara real-time pada sistem monitoring yang terhubung dengan laptop melalui tampilan video langsung. Terdapat *Timing Diagram* Lampu Lalu dalam keadaan normal pada Gambar 8.



Gambar 7. Timing Diagram

Sistem Smart Traffic Light ini menerapkan dua mode prioritas, yaitu:

- Prioritas 1: Jalur Timur-Selatan
- Prioritas 2: Jalur Selatan-Barat

Pengoperasian kedua prioritas ini tidak dapat dilakukan secara bersamaan karena sistem hanya menggunakan dua kamera. Oleh karena itu, mode prioritas hanya bisa diaktifkan secara bergantian. Mode prioritas pertama (Timur-Selatan) diaktifkan terlebih dahulu, di mana jalur Timur akan mendapatkan lampu hijau lebih lama ketika terdeteksi jumlah kendaraan melebihi 3 unit (padat), sementara jalur Selatan, yang lebih sepi (0-3 kendaraan), tetap pada kondisi lampu merah lebih lama. Setelah mode Timur-Selatan selesai, mode Prioritas 2 (Selatan-Barat) diaktifkan. Pada mode ini, jika jalur Selatan terdeteksi padat (lebih dari 3 kendaraan), durasi lampu hijau dipercepat, sementara jalur Barat yang juga padat akan memiliki lampu merah yang dipercepat.

Tabel 2. Kondisi Jumlah Kendaraan Terhadap Lampu Lalu lintas

Prioritas 1			
No	Timur (Hijau)	Selatan (Merah)	Hasil
1	Sepi	Sepi	Normal
2	Padat	Sepi	Hijau (T) diperlama dan Merah (S) diperlama
3	Padat	Padat	Normal
Prioritas 2			
No	Timur (Hijau)	Selatan (Merah)	Hasil
1	Sepi	Sepi	Normal
2	Sepi	Padat	Hijau (S) dipercepat dan Merah (B) dipercepat
3	Padat	Padat	Normal

Tabel 2. menggambarkan hasil perbandingan kondisi kendaraan di jalur prioritas dengan durasi lampu lalu lintas. Hasil ini menunjukkan bahwa pada situasi dengan kepadatan yang berbeda, sistem akan memperpanjang atau mempercepat durasi lampu lalu lintas untuk memaksimalkan kelancaran arus lalu lintas. Kondisi Operasional Berdasarkan Jumlah Kendaraan

- a. Padat: Jumlah kendaraan > 3
- b. Sepi: Jumlah kendaraan 0 - 3

Pada kasus ini, sistem mendeteksi kondisi kepadatan lalu lintas menggunakan kamera dan mengirimkan data tersebut secara real-time ke PLC untuk menentukan prioritas lampu lalu lintas. Dikarenakan simulasi ini dilakukan dalam kondisi alat yang terbatas, definisi padat dan sepi disederhanakan seperti yang telah dijelaskan di atas.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Pengujian Hasil Deteksi Pada Tiap Kamera Secara *Realtime*

##### 3.1.1 Pengujian Secara *Real-time* pada *Traffic Light* Prioritas 1 (Timur-Selatan)

Pada pengujian ini, penempatan mobil hanya akan dikondisikan timur 3 dan selatan 3. Kondisi tersebut sudah memenuhi syarat kondisi sepi dan padat.



(a) Simpang Selatan (b) Simpang Timur  
Gambar 8. Keadaan 3 Mobil

Gambar 9 menunjukkan hasil deteksi mobil mainan menggunakan model SSD MobileNet V2 pada prototipe lampu lalu lintas yang dikontrol oleh PLC. Pengujian dilakukan secara *real-time* di dua simpang berbeda: simpang selatan dan simpang timur. Hasil deteksi ini divisualisasikan dengan kotak pembatas (*bounding box*) yang menandai posisi mobil mainan yang terdeteksi serta persentase confidence level yang menunjukkan tingkat keyakinan model terhadap deteksi tersebut.

Tabel 3. PLC Menerima Data Jumlah Kendaraan

Address	Data Type / Format	Value	Value (Binary)	Comment
D42	CHANNEL (Hex, Channel)	0000 Hex	0000 0000 0000 0000	Selatan
D43	CHANNEL (Hex, Channel)	0000 Hex	0000 0000 0000 0000	Barat

Data jumlah kendaraan pada tiap simpang tersebut kemudian dikirim ke PLC Omron CP1-E N40 melalui komunikasi serial RS232. Data jumlah kendaraan pada simpang selatan dikirim ke memori D42 dan pada simpang barat dikirim ke memori D43. Semua data dikirim dalam format bilangan heksadesimal.

##### 3.1.2 Pengujian Secara *Real-time* pada *Traffic Light* Prioritas 2 (Selatan-Barat)



(a) Simpang Selatan (b) Simpang Barat

Gambar 9. Kondisi 3 Mobil di Selatan dan 3 Mobil di Barat

Pada Gambar 9 a yang diambil dari simpang selatan, terdapat tiga mobil mainan yang terdeteksi dengan *confidence level* masing-masing 78%, 59%, dan 73%. Total jumlah mobil yang terdeteksi di simpang ini adalah tiga mobil dengan kecepatan deteksi mencapai 5,58 FPS (*Frame Per Second*).

Pada Gambar 9 b yang diambil dari simpang barat, terdapat empat mobil mainan yang terdeteksi dengan *confidence level* masing-masing 50%, 46%, 75%, dan 67%. Total jumlah mobil yang terdeteksi di simpang ini adalah empat mobil dengan kecepatan deteksi mencapai 6,02 FPS. Meskipun ada variasi dalam tingkat keyakinan, sistem tetap mampu mendeteksi mobil mainan dengan baik.

Tabel 4. Menerima Data Jumlah Kendaraan

Address	Data Type / Format	Value	Value (Binary)	Comment
D42	CHANNEL (Hex, Channel)	0003 Hex	0000 0000 0000 0011	Selatan
D43	CHANNEL (Hex, Channel)	0004 Hex	0000 0000 0000 0100	Barat

Data jumlah kendaraan pada tiap simpang tersebut kemudian dikirim ke PLC Omron CP1-E N40 melalui komunikasi serial RS232 sama seperti sebelumnya. Nilai memori D43 menjadi 4 Hex dan nilai memori D42 menjadi 3 Hex.

### 3.2. Pengujian Komunikasi Raspberry Pi 4B ke PLC

Pengujian ini menggunakan software Thonny dengan program tersendiri. Program ini dibuat untuk mengetahui response block yang dikirim oleh PLC tepat setelah Raspberry Pi-4B mengirimkan *command block*. Raspberry Pi-4B mengirimkan data jumlah kendaraan dalam bentuk *command block*. Hasil pengujian selengkapnya dapat dilihat di Tabel 5.

Tabel 5. Hasil Pengujian Komunikasi Raspberry Pi 4 B ke PLC

Alamat PLC	Keterangan	Command Block	Response Block
-	Ubah PLC ke Monitor Mode	@00SC0252*	@00SC0050*
D0	Memori Jumlah Kendaraan Timur	@00WD000000053*	@00WD0053*
		@00WD000000152*	@00WD0053*
		@00WD0000000251*	@00WD0053*
		@00WD0000000350*	@00WD0053*
		@00WD0000000457*	@00WD0053*
D1	Memori Jumlah Kendaraan Selatan 1	@00WD0001000052*	@00WD0053*
		@00WD0001000153*	@00WD0053*
		@00WD0001000250*	@00WD0053*
		@00WD0001000351*	@00WD0053*
		@00WD0001000456*	@00WD0053*
D42	Memori Jumlah Kendaraan Selatan 2	@00WD0004200055*	@00WD0053*
		@00WD00042000154*	@00WD0053*
		@00WD00042000257*	@00WD0053*
		@00WD00042000356*	@00WD0053*
		@00WD00042000451*	@00WD0053*
D43	Memori Jumlah Kendaraan Barat	@00WD0004300054*	@00WD0053*
		@00WD00043000155*	@00WD0053*
		@00WD00043000256*	@00WD0053*
		@00WD00043000357*	@00WD0053*
		@00WD00043000450*	@00WD0053*

Dari hasil pengujian yang telah dilakukan didapatkan bahwa komunikasi serial bekerja dengan baik dan sudah dapat digunakan untuk keperluan sistem. Seperti pada *command block* @00WD0000000457\*, instruksi tersebut berarti PLC pertama atau dengan nomor host 00 dan menulis data(dengan simbol WD) di memori D0 dengan nilai 0004 Hexa. *Response block* yang didapatkan adalah @00WD0053\* yang berarti pengaksesan PLC pertama dan penulisan data berjalan lancar.

### 3.3. Pengujian Program PLC untuk Mengontrol Lampu Lalu Lintas

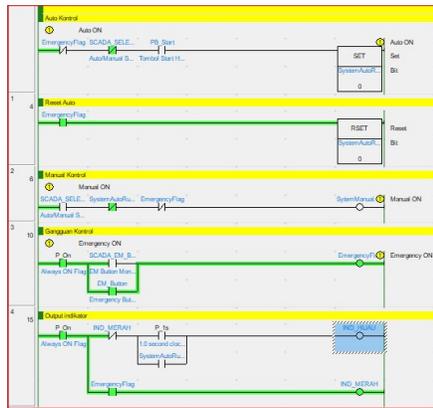
#### 3.3.1 Pengujian Rangkaian Input Kontrol Terhadap Hardware menggunakan CX-Supervisor

Tujuan dari pengujian ini adalah untuk memastikan bahwa semua sinyal dari sensor lalu lintas dapat diterima dengan baik oleh Raspberry Pi dan diproses dengan benar. Pengujian ini juga bertujuan memverifikasi bahwa data yang diterima dapat dikomunikasikan ke PLC melalui SCADA, sehingga sistem lampu lalu lintas dapat beroperasi secara otomatis dan sesuai dengan kondisi lalu lintas di lapangan.

Tabel 6. Kondisi perbandingan program dan alat simulasi

Address	Value	Comment	Description
0.00	0	Tombol Start Hardware	TrafficLightsSequence.PB_Start
0.01	1	Emergency Button Hardware	TrafficLightsSequence.EM_Button
101.05	1		TrafficLightsSequence.IND_MERAH
101.04	0		TrafficLightsSequence.IND_HIJAU

Pada pengujian ini, tombol dan lampu hardware merespons sesuai instruksi dari CX-Programmer. Lampu darurat menyala saat tombol darurat ditekan, sementara lampu siaga menandakan operasi normal. Tombol darurat juga berhasil mematikan sistem sesuai prosedur. Hasil ini menunjukkan integrasi perangkat lunak dan hardware berjalan dengan baik, memastikan seluruh fitur berfungsi sesuai spesifikasi.



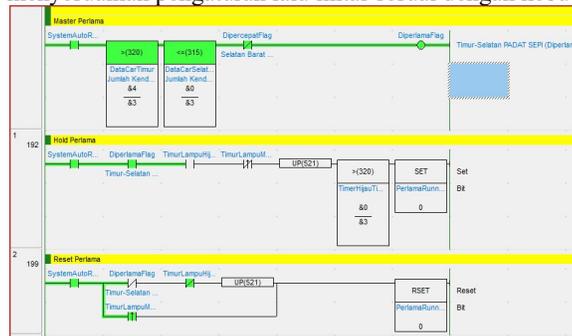
a. Program Rangkaian Kontrol

b. Kondisi pada alat simulasi

Gambar 10. Perbandingan antara program dan alat simulasi.

**3.3.2 Pengujian secara real-time pada traffic light Prioritas 1 (Timur-Selatan)**

Gambar 11 dan Tabel 7 di bawah ini menunjukkan sistem kontrol yang dirancang untuk mengelola prioritas 1 (arah timur-selatan), di mana data jumlah kendaraan dikirimkan ke memori D0 dan D1 pada PLC dari Raspberry Pi. Berdasarkan data input yang diterima oleh PLC, sistem akan mengeksekusi perintah dan mengaktifkan prioritas 1, menyesuaikan pengaturan lalu lintas sesuai dengan kebutuhan yang terdeteksi.



Gambar 11. Program PLC sistem Prioritas 1 (timur-selatan).

Tabel 7. Data memori PLC yang diperoleh dari Raspberry Pi.

Address	Value	Comment	Description
D0	&4	Jumlah Kendaraan Timur	TrafficLightsSequence.DataCarTimur
D1	&0	Jumlah Kendaraan Selatan (Prioritas 1)	TrafficLightsSequence.DataCarSelatan_1

Tabel 8 di bawah ini menunjukkan perbandingan durasi penyalan lampu hijau di persimpangan timur antara kondisi normal dan prioritas 1. Pengaktifan prioritas 1 adalah dengan terpenuhinya syarat diatas yaitu, simpang timur dalam keadaan padat (>3) dan simpang selatan dalam keadaan sepi (≤ 3). Dalam kondisi normal, lampu hijau menyala selama 10 detik. Namun, ketika sistem berada dalam mode prioritas 1, durasi penyalan lampu hijau di persimpangan tersebut diperpanjang sebesar 5 detik, sehingga total durasinya menjadi 15 detik. Hal ini menunjukkan adanya penyesuaian waktu siklus lampu lalu lintas untuk memberikan preferensi pada arus kendaraan di persimpangan tertentu sesuai dengan kebutuhan prioritas lalu lintas.

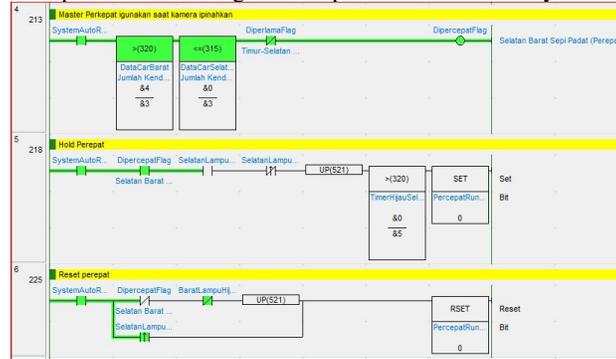
Tabel 8. Perbandingan durasi antara keadaan normal dan keadaan Prioritas 1.

Address	Value	Comment
D12	&10	Memori Timer Hijau Timur
D13	&54	Memori Timer Merah Timur

Address	Value	Comment
D12	&15	Memori Timer Hijau Timur
D13	&54	Memori Timer Merah Timur

**3.3.3 Pengujian secara real-time pada traffic light Prioritas 2 (Selatan-Barat)**

Gambar 12 dan Tabel 9 di bawah menunjukkan sistem kontrol untuk prioritas 2 (selatan-barat), di mana data jumlah kendaraan dikirim ke memori D42 dan D43 pada PLC dari Raspberry Pi. Berdasarkan input data, PLC akan menjalankan perintah dan mengaktifkan prioritas 1 untuk menyesuaikan pengaturan lalu lintas.



Gambar 12. Program PLC sistem Prioritas 2 (selatan-barat).

Tabel 9. Data memori PLC yang diperoleh dari Raspberry Pi.

Address	Value	Comment	Description
D42	&0	Jumlah Kendaraan Selatan (Prioritas 2)	TrafficLightsSequence.DataCarSelatan_2
D43	&4	Jumlah Kendaraan Barat	TrafficLightsSequence.DataCarBarat

Tabel 10 dibawah ini membandingkan durasi lampu hijau di persimpangan selatan antara kondisi normal dan prioritas 2. Pengaktifan prioritas 2 adalah dengan terpenuhinya syarat diatas yaitu, simpang selatan dalam keadaan sepi ( $\leq 3$ ) dan simpang barat dalam keadaan padat ( $> 3$ ). Dalam kondisi normal, lampu hijau menyala selama 10 detik, tetapi pada mode prioritas 2, durasinya berkurang 5 detik menjadi 5 detik. Ini menunjukkan penyesuaian waktu lampu lalu lintas untuk memprioritaskan arus kendaraan di persimpangan sesuai kebutuhan.

Tabel 10. Perbandingan durasi antara keadaan normal dan keadaan Prioritas 2.

Address	Value	Comment
D14	&10	Memori Timer Hijau Selatan
D15	&15	Memori Timer Merah Selatan

Address	Value	Comment
D14	&4	Memori Timer Hijau Selatan
D15	&54	Memori Timer Merah Selatan

**3.4. Pengujian Hasil CX-Supervisor untuk Pengontrolan dan Monitoring Traffic Light secara real-time.**

**3.4.1 Pengujian control SCADA menggunakan CX-Supervisor**

Pengujian ini bertujuan untuk memastikan bahwa kontrol pada sistem Smart Traffic Light berfungsi dengan optimal. Pengujian ini mencakup pengujian berbagai fitur kontrol, seperti Emergency Lamp, Emergency Button, Push Button ON, dan lampu indikator operasi (ON/Standby Lamp). Setiap komponen diuji secara menyeluruh untuk memastikan keandalan sistem dalam kondisi darurat dan operasi normal.



(a) Kondisi pada CX-Supervisor (b) Kondisi pada Alat Simulasi

Gambar 13. Kondisi CX-Supervisor terhadap alat simulasi

Tabel 11. Kondisi pada CX-Programmer

Address	Value	Comment
0.00	0	Tombol Start
0.01	1	Tombol Stop
101.04	1	On/Standby
101.05	0	Emergency Lamp

Pada proses pengujian, tombol dan lampu perangkat keras memberikan respons yang sesuai dengan instruksi yang diberikan melalui perangkat lunak kontrol *CX-Supervisor*. Lampu darurat menyala ketika tombol darurat ditekan, sementara lampu siaga menyala untuk menandakan kondisi operasi normal. Tombol darurat juga berhasil memutus aliran sistem sesuai prosedur. Hasil ini menunjukkan bahwa integrasi antara perangkat lunak dan perangkat keras berjalan dengan baik, memastikan seluruh fitur sistem bekerja sesuai spesifikasi. Pengujian ini membuktikan bahwa komunikasi antara sistem simulasi dan panel kontrol berjalan efektif, mencerminkan keakuratan dan kehandalan implementasi SCADA dalam proyek ini.

**3.4.2 Pengujian secara *real-time* pada *traffic light* Prioritas 1 (Timur-Selatan)**

Pada kondisi ini, lampu lalu lintas di simpang Timur telah memenuhi kriteria pengaktifan di perlamanya simpang, yaitu jumlah kendaraan yang terdeteksi lebih dari tiga, pengujian ini juga melibatkan penempatan kendaraan secara manual pada alat simulasi untuk simpang Timur. Data mengenai jumlah kendaraan ini kemudian dikirimkan ke *CX-Supervisor* melalui Programmable Logic Controller (PLC), yang mengolah informasi tersebut dan menyimpannya dalam memori dengan alamat "D0" dan "D1". Proses ini menunjukkan kemampuan sistem untuk secara otomatis memonitor dan mengendalikan arus lalu lintas berdasarkan data yang diterima dari sensor dan perangkat simulasi.

Dengan demikian, hasil pengujian membuktikan bahwa sistem integrasi antara PLC dan *CX-Supervisor* berfungsi dengan baik, memastikan pengaturan lalu lintas berbasis deteksi kendaraan berjalan sesuai rencana



(a) Kondisi pada *CX-Supervisor*

(b) Kondisi pada Alat Simulasi

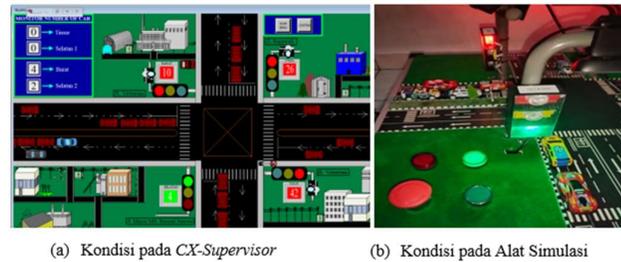
Gambar 14. Kondisi diperlama terhadap Timur (padat) dan Selatan (sepi)

Tabel 12. Kondisi pada CX-Programmer

Address	Value	Value (Binary)	Comment
D0	&4	0000 0000 0000 0100	Timur
D1	&2	0000 0000 0000 0010	Selatan

**3.4.3 Pengujian secara *real-time* pada *traffic light* Prioritas 2 (Selatan-Barat)**

Pada skenario ini, parameter untuk mempercepat waktu siklus lampu lalu lintas pada simpang Selatan telah terpenuhi, dengan timer yang dikurangi menjadi 5 detik. Hal ini terjadi karena jumlah kendaraan pada simpang Barat mencapai lebih dari tiga unit, yang sesuai dengan kondisi padat sebagaimana diatur dalam logika pengendalian. Proses pengujian dilakukan dengan penempatan kendaraan secara manual pada kedua simpang, Selatan dan Barat, untuk mensimulasikan skenario lalu lintas nyata. Kondisi padat ini memicu respons otomatis yang mengakselerasi durasi lampu hijau dzzi simpang Selatan, sehingga mengurangi waktu tunggu. Nilai pada register memori "D42" dan "D43" di dalam Programmable Logic Controller (PLC) dipantau secara khusus, di mana nilai untuk simpang Barat (D43) melebihi jumlah yang ditentukan, menandakan kondisi lalu lintas yang padat.



Gambar 15. Kondisi dipercepat terhadap Selatan (sepi) dan Barat (padat)

Tabel 13. Kondisi pada CX-Programmer

Address	Value	Value (Binary)	Comment
D42	&2	0000 0000 0000 0010	Selatan
D43	&4	0000 0000 0000 0100	Barat

Sistem *CX-Supervisor* berfungsi sebagai antarmuka monitoring yang memudahkan pengawasan kondisi ini secara *real-time*. Dengan perangkat lunak ini, operator dapat melihat perubahan nilai memori yang menggambarkan kepadatan lalu lintas pada tiap simpang. Selain itu, *CX-Supervisor* memungkinkan operator untuk menilai performa dari logika pengendalian yang diimplementasikan pada PLC. Setiap kali kondisi padat terdeteksi pada simpang Barat, *CX-Supervisor* akan menampilkan informasi yang relevan dan menyorot perubahan status pada timer lampu lalu lintas simpang Selatan. Hal ini meningkatkan efisiensi operasional sistem, sekaligus memfasilitasi analisis data untuk keperluan optimasi pengendalian lalu lintas di masa mendatang.

#### 4. KESIMPULAN

Sistem pengiriman data jumlah kendaraan ke PLC bekerja efisien tanpa mengganggu proses deteksi objek pada Raspberry Pi 4B. Penggunaan library threading di Python memungkinkan eksekusi paralel antara pengiriman data dan deteksi objek, sehingga keduanya dapat berjalan secara independen. Sistem *traffic light* yang menggunakan PLC dan Raspberry Pi-4B dapat beroperasi secara otomatis dan adaptif berdasarkan data lalu lintas yang dianalisis melalui SCADA. Optimalisasi pewaktuan lampu lalu lintas dengan algoritma adaptif ini berhasil mengurangi waktu tunggu secara signifikan dengan menyesuaikan durasi lampu hijau, kuning, dan merah sesuai volume kendaraan. Smart traffic light dirancang untuk mengurangi kemacetan dan memudahkan interface (SCADA) untuk para operator secara real-time.

#### 5. UCAPAN TERIMA KASIH

Kami ingin menyampaikan rasa terima kasih yang mendalam kepada Politeknik Negeri Sriwijaya atas dukungan penuh dan fasilitas yang diberikan untuk penelitian ini. Ucapan terima kasih juga kami sampaikan kepada Ketua Jurusan Elektro Bapak Ir. Iskandar Lutfi, M.T, Kepala Bengkel Bapak Mutiar, S.T., M.T dan Bersiap Ginting, S.T., M.T serta kedua dosen pembimbing atas waktu, dan dedikasi yang luar biasa. Tanpa dukungan dan arahan mereka, penelitian ini tidak akan berjalan dengan baik. Kami sangat menghargai semua kontribusi yang telah mereka berikan selama proses penelitian ini.

#### 6. DAFTAR PUSTAKA

- [1] L. Dyusembayeva, B. Belgibaev, M. Mansurova, S. Abdrakhim, "Neural computer visualization of smart programs in megacities of the country," Certificate of Authorship of the Republic of Kazakhstan No. 39772 dated 19 October 2023
- [2] L. Bhaskar, A. Sahai, D. Sinha, G. Varshney, and T. Jain, "Intelligent traffic light controller using inductive loops for vehicle detection," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Sep. 2015, doi: <https://doi.org/10.1109/ngct.2015.7375173>.
- [3] Riansa E.P. Polah, Rizal Sengkey, and Yaulie D.Y. Rindengan, Jurnal Teknik Elektro dan Komputer, vol. 4, no. 4, pp. 35–45, 2015, doi: <https://doi.org/10.35793/jtek.v4i4.8971>.
- [4] N. Diaz, J. Guerra, and J. Nicola, "Smart Traffic Light ControlSystem," 2018 IEEE 3rd Ecuador Tech. Chapters Meet. ETCM 2018, 2018, doi: 10.1109/ETCM.2018.8580282.
- [5] A. Firdous, Indu, and V. Niranjan, "Smart Density Based TrafficLight System," ICRITO 2020 - IEEE 8th Int. Conf. Reliab. InfocomTechnol. Optim. (Trends Futur. Dir., pp. 497–500, 2020, doi:10.1109/ICRITO48877.2020.9197940.
- [6] J. Smith, "Programmable logic controllers in traffic systems," Journal of Traffic Management, vol. 12, no. 3, pp. 45–58, 2020.
- [7] A. Toroman and E. Mujčić, "Application of industrial PLC for controlling intelligent traffic lights," 2017 25th Telecommunication Forum (TELFOR), Belgrade, Serbia, 2017, pp. 1-4, doi: 10.1109/TELFOR.2017.8249411.

- [8] S. Chepure, “Smart Traffic Signal Control System: Design And Implementation,” vol. 16, no. 1, p. 256, 2019, Available: [https://www.webology.org/data-cms/articles/20220912110108pmwebology%2016%20\(1\)%20-%201.pdf](https://www.webology.org/data-cms/articles/20220912110108pmwebology%2016%20(1)%20-%201.pdf)
- [9] M. Vidhyia and S. Elayaraja, “Traffic Light Control System Using Raspberry-PI,” Asian Journal of Electrical Sciences, vol. 5, no. 1, pp. 8–12, May 2016, doi: <https://doi.org/10.51983/ajes-2016.5.1.1970>.
- [10] Pravin Sonwane, H. Kaushik, Mansi Bardawat, and A. Gupta, “Controlling a Smart Traffic Light Using Programmable Logic Controller (PLC),” Mar. 20, 2024. [https://www.researchgate.net/publication/379082584\\_Controlling\\_a\\_Smart\\_Traffic\\_Light\\_Using\\_Programmable\\_Logic\\_Controller\\_PLC](https://www.researchgate.net/publication/379082584_Controlling_a_Smart_Traffic_Light_Using_Programmable_Logic_Controller_PLC)
- [11] Y. Cherdantseva, P. Burnap, S. Nadjm-Tehrani, and K. Jones, “A Configurable Dependency Model of a SCADA System for Goal-Oriented Risk Assessment,” Applied Sciences, vol. 12, no. 10, p. 4880, May 2022, doi: <https://doi.org/10.3390/app12104880>.
- [12] W. Liu et al., “SSD: Single Shot MultiBox Detector,” Computer Vision – ECCV 2016, vol. 9905, pp. 21–37, 2016, doi: [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2).
- [13] Helfy Susilawati, P. Rahman, Ade Rukmana, M. Matin, None Sarbini, and N. Ismail, “Smart Traffic Light Using Raspberry Pi and Digital Image Processing,” pp. 1–6, Jul. 2023, doi: <https://doi.org/10.1109/icwt58823.2023.10335398>.