

## SISTEM SMART TRAFFIC LIGHT MENGGUNAKAN ALGORITMA YOLOv8

Dewi Primasari<sup>1</sup>, Ghifari Ferdian R<sup>2</sup>, Zakiah Aulia R<sup>3</sup>, Ulfiya Tussyifaa<sup>4</sup>, Arrobi Rot Wiranto<sup>5</sup>

<sup>1, 2, 3, 4, 5</sup> Program Studi Teknik Informatika, Fakultas Teknik & Sains, Universitas Ibn Khaldun Bogor  
Email: <sup>1</sup>dewiprimasari9@gmail.com, <sup>2</sup>ghifaririzqullah0@gmail.com, <sup>3</sup>zakiahaulia01@gmail.com  
<sup>4</sup>ulfiyatussyifaa21@gmail.com, <sup>5</sup>kazerobi@gmail.com

### Abstrak

Kemacetan lalu lintas telah menjadi masalah serius di banyak negara, termasuk Indonesia. Manajemen lalu lintas merupakan tugas yang menantang, terutama di negara-negara berkembang. Salah satu penyebab kemacetan adalah barisan panjang kendaraan di persimpangan karena sinyal lampu lalu lintas yang menyala pada waktu yang sama tanpa memperhatikan jumlah kendaraan yang mengantri. Pendekatan *computer vision* dapat diterapkan sebagai solusi dengan meminimalkan waktu menunggu untuk pengemudi dengan menyesuaikan lamanya waktu sinyal lampu hijau menyala berdasarkan jumlah kendaraan di setiap jalur. Untuk melakukan pendekatan ini, sistem deteksi kendaraan dibangun menggunakan algoritma YOLOv8 (You Only Look Once versi 8), yang beroperasi secara *real-time*. Data yang digunakan bersumber dari CCTV yang dipasang pada setiap lampu lalu lintas di simpang empat. Algoritma YOLOv8 menunjukkan akurasi dan efisiensi tinggi dalam deteksi objek dalam berbagai penelitian. Untuk mengetahui sejauh mana model YOLO yang telah dibangun dapat memprediksi data dengan benar maka diperlukannya evaluasi model dengan menghitung akurasi, recall, dan skor mAP50. Penelitian ini bertujuan membangun sistem pengaturan lampu lalu lintas cerdas menggunakan YOLOv8, berdasarkan visi komputer dan perpustakaan OpenCV. Hasil yang diperoleh dari perhitungan sistem ini adalah nilai akurasi sebesar 0,689, nilai recall sebesar 0,578, dan skor mAP50 sebesar 0,65.

**Kata Kunci:** *You Only Look Once (YOLO), Computer Vision, OpenCV, Lampu Lalu Lintas*

### Abstract

Traffic jams have become a serious problem in many countries, including Indonesia. Traffic management is a challenging task, especially in developing countries. One of the causes of traffic jams is long lines of vehicles at intersections because the traffic light signals are on at the same time without paying attention to the number of vehicles queuing. The computer vision approach can be applied as a solution by minimizing waiting time for drivers by adjusting the length of time the green light signal is on based on the number of vehicles in each lane. To carry out this approach, a vehicle detection system was built using the YOLOv8 (You Only Look Once 8<sup>th</sup> version) algorithm, which operates in real-time. The data used comes from CCTV installed at every traffic light at intersection four. The YOLOv8 algorithm shows high accuracy and efficiency in object detection in various studies. To find out to what extent the YOLO model that has been built can predict the data correctly, it is necessary to evaluate the model by calculating accuracy, recall and mAP50 score. This research aims to build an intelligent traffic light management system using YOLOv8, based on computer vision and the OpenCV library. The results obtained from this system calculation are an accuracy value of 0.689, a recall value of 0.578, and an mAP50 score of 0.65.

**Keywords:** *You Only Look Once (YOLO), Computer Vision, OpenCV, Traffic Light*

## I. PENDAHULUAN

Kemacetan jalan di kota Bogor, Indonesia, kian meningkat setelah keadaan kembali normal sejak pandemi mereda. Kemacetan lalu lintas merupakan

masalah yang serius di sebuah kota dan negara, bahkan hal ini merupakan masalah global di dunia. Manajemen lalu lintas menjadi hal yang paling menantang dan sulit diatasi sebuah negara

(Amwin, 2021). Dari data tahun 2019, tercatat sekitar 133,6 juta kendaraan beredar di Indonesia. Pertumbuhan penggunaan transportasi setiap hari terus meningkat, yang menyebabkan peningkatan jumlah data kendaraan. Kenaikan jumlah kendaraan ini menjadi pemicu terjadinya kemacetan di beberapa wilayah Indonesia (Hayati et al., 2023:91).

Permasalahan kemacetan telah menarik perhatian sejumlah ahli teknisi dan peneliti yang berupaya menemukan solusi untuk mengatasi tantangan masalah tersebut. Pendekatan penyelesaian masalah melibatkan kegiatan pengkategorian dan pencatatan jumlah kendaraan sebagai langkah untuk memahami karakteristik lalu lintas, mengumpulkan data informasi lalu lintas, serta mengevaluasi kinerja sistem lalu lintas (Sunaryo dan Kusumawati, 2020:11). Banyak solusi lain yang diusulkan sebagai upaya menurunkan kemacetan lalu lintas seperti pelebaran jalan raya atau pembangunan jalan layang, namun hal itu tidak dapat diselesaikan dengan mengembangkan infrastruktur saja (E. Harahap et al., 2017; Fali Oklilas et al., 2023:799).

Kemacetan banyak terjadi di berbagai kawasan, terutama pada kawasan persimpangan jalan. Panjangnya antrian kendaraan di persimpangan sering terjadi karena kinerja lampu lalu lintas yang tidak efektif. Hal ini perlu menjadi perhatian dalam upaya mengembangkan manajemen lalu lintas, khususnya pada lampu lalu lintas di sebuah persimpangan yang sering sekali menyebabkan kemacetan. Salah satu caranya dengan mengupayakan meminimalkan waktu tunggu yang dihabiskan pengendara untuk mengantri mendapatkan giliran lampu hijau pada lampu lalu lintas (Dian Maniswari et al., 2015:2166). Lamanya waktu menyala lampu hijau, warna merah, dan warna kuning secara sama rata pada setiap jalur tidaklah efektif. Ketidakefisienan muncul karena ketika salah satu jalur yang hanya terdapat sedikit kendaraan atau tidak ada kendaraan mendapatkan waktu lampu hijau yang dibagi sama rata, maka waktu akan terbuang dan kendaraan di jalur lain tetap harus menunggu giliran mendapatkan lampu hijau. Maka diperlukan manajemen waktu lampu lalu lintas yang sesuai dengan kondisi jumlah kendaraan pada setiap jalur untuk meningkatkan efektifitas sehingga mengurangi kemacetan antrian lalu lintas di persimpangan.

Penerapan teknologi *computer vision* telah meluas ke berbagai sektor, termasuk keamanan, pekerjaan sosial, kesehatan, transportasi, dan berbagai bidang terkait lainnya. Dalam Sistem

Transportasi Cerdas (ITS), data lalu lintas memegang peranan penting sebagai elemen kunci dalam penelitian dan desain sistem (Amwin, 2021). Sistem pendeteksi jumlah kendaraan dapat membantu dalam mengelola *traffic light* pada sebuah persimpangan untuk mengurangi kemacetan (Rachmawati dan Widhyaestoeti, 2020:360). Salah satu metode sistem deteksi objek dengan memproses data secara *real-time* yang merupakan jaringan syaraf tiruan (JST) yaitu *You Only Look Once* (YOLO) (Redmon et al., 2016:799; Saputra et al., 2023:161). Algoritma YOLO adalah salah satu algoritma yang memiliki kemampuan mendeteksi yang akurat dan cepat bahkan dua kali lipat dibandingkan metode lain (M. Harahap et al., 2019). Algoritma YOLO dapat mendeteksi objek kendaraan dengan tingkat akurasi tinggi (Al Mudawi et al., 2023; Gibran et al., 2024; Hayati et al., 2023).

YOLO adalah sebuah algoritma yang dikembangkan untuk mendeteksi objek secara *real-time* dengan menggunakan *repurpose classifier* atau *localizer* (Cholissodin dan Soebroto, 2021:1). Model YOLO bekerja dengan langkah awal membagi gambar menjadi beberapa kotak pembatas untuk mendeteksi objek yang ingin diidentifikasi (Ardiansyah et al., 2022:52). Algoritma YOLO juga bekerja secara *real-time* sehingga cocok untuk memantau kendaraan pada lalu lintas dan YOLOv8 adalah model mutakhir dan terunggul *State of The Art* (SOTA) yang dibangun berdasarkan kesuksesan versi YOLO sebelumnya dan memperkenalkan fitur-fitur baru serta peningkatan untuk lebih meningkatkan kinerja dan fleksibilitas (Setianingsih dan Paryasto, 2022:5038). YOLO dapat melakukan pengenalan objek secara *real-time* melalui penerapan jaringan saraf konvolusional. Pada setiap jaringan, lapisan konvolusional digunakan untuk menjalani proses konvolusi, memungkinkan identifikasi objek dengan efektif (Hidayati, 2017). Pendekatan ini membedakan dirinya dengan kemampuan untuk melakukan deteksi objek dalam satu kali proses, dan memungkinkan respons waktu yang cepat. YOLO memecah gambar input menjadi grid, di mana setiap grid bertanggung jawab atas deteksi objek di dalamnya. Dengan demikian, YOLO menjadi solusi efisien dan cepat dalam mendeteksi objek secara kontinu dalam gambar atau video. Metode ini telah diterapkan secara luas dalam berbagai aplikasi, termasuk keamanan, kendaraan otonom, dan pemrosesan video *real-time*.

Berdasarkan penelitian-penelitian sebelumnya maka Sistem Smart Traffic Light ini akan dibangun menggunakan algoritma YOLO berbasis *computer*

*vision* dengan *library* OpenCV. Adapun versi YOLO yang digunakan yaitu YOLOv8, yaitu pendeteksi versi terbaru untuk memaksimalkan kinerja model (Reis et al., 2023). Sistem ini akan bekerja dengan cara mendeteksi jumlah kendaraan yang dilakukan dengan kamera di simpang empat. Data bersumber dari CCTV yang dipasang pada setiap lampu lalu lintas dan CCTV tersebut akan mendeteksi kendaraan yang ada di jalur persimpangan tersebut, kemudian input yang diberikan CCTV diolah dengan model YOLOv8 dan memberikan output waktu lampu hijau yang tepat untuk setiap jalur.

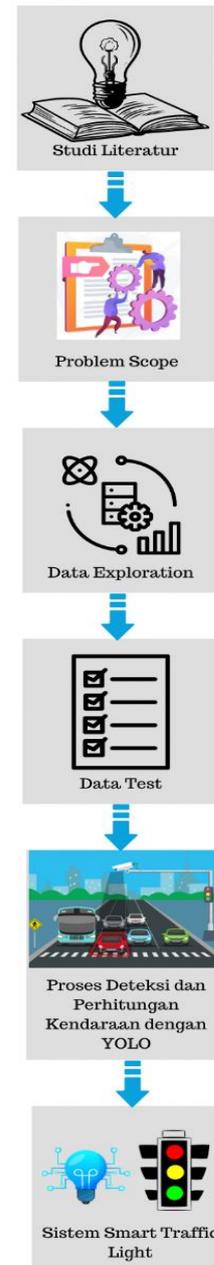
## II. METODE

Metode yang digunakan pada sistem pembuatan Smart Traffic Light ini adalah algoritma YOLOv8. Alur kerja perhitungan meliputi studi literatur, pencarian dataset, model YOLOv8, perhitungan hasil sistem lampu lalu lintas yang dapat mendeteksi kendaraan. Alur kerja sistem ini dapat di lihat pada Gambar 1.

Pada tahap studi literatur dilakukan pencarian referensi seperti e-book, artikel dan penelitian sebelumnya mengenai konsep dasar deteksi objek algoritma YOLO, pemahaman kelebihan dan kekurangan YOLO, dan perbandingan permasalahan kasus menggunakan metode lain.

Setelah mendapatkan kasus permasalahan selanjutnya mengkritisi masalah dengan pengembangan solusi atau penyelesaian masalah, bagaimana problem scope yang diambil ini dapat diselesaikan dengan sistem yang akan di bangun. Proses penggunaan 4W (*Who, What, Where, dan Why*) dalam menentukan lingkup masalah membantu dalam identifikasi dan penyelesaian masalah dengan lebih efisien (Hayati et al., 2023:91). Dengan pendekatan ini dapat lebih baik mengidentifikasi dan menyelesaikan masalah dengan upaya yang lebih efektif.

Data eksplorasi tahapan di mana data yang sudah dianotasi kemudian dibentuk ke dalam dataset dan akan di lakukan tahapan *training* dan *testing*. Proses ini melibatkan anotasi data citra, di mana objek dalam setiap gambar diberi kotak pembatas (*bounding box*) dan diberikan kelas yang sesuai sebagai penanda identifikasi (Hayati et al., 2023:91). Dataset berisi kumpulan gambar kendaraan mobil, motor dan truk yang kemudian dianotasi menggunakan roboflow. Proses anotasi ini memberikan kotak pembatas pada objek yang akan dideteksi seperti mobil, motor dan truk yang kemudian diberikan label pada setiap kendaraan.



Gambar 1. Flowchart

Setelah melakukan data eksplorasi, tahapan selanjutnya adalah melakukan training model menggunakan algoritma YOLO. melalui *training*, *validation* dan *testing data* kemudian dapat memprediksi pada model YOLO tersebut. Setelah tahap ini baru didapatkan model YOLO yang dapat mendeteksi kendaraan melalui video yang diambil dari Dinas Perhubungan (Dishub).

Data video merupakan hasil dari kamera secara *real-time* yang selanjutnya di inputkan kedalam model YOLO yang sudah dibuat. Data Dishub ini berupa 4 video berdurasi 10 menit. Setiap video menggambarkan keadaan jalan di sebuah persimpangan.

Setelah objek terdeteksi dengan YOLO selanjutnya kendaraan akan dihitung setelah melewati polyline dengan didukung oleh *Computer Vision*. Proses pembuatan polyline ini menggunakan *library* OpenCV yang merupakan teknologi komputer untuk mengolah *image/citra* yang ditangkap oleh alat perekam, lalu dikonversi dari analog ke digital dengan tujuan mengidentifikasi gambar salah satunya menghitung kendaraan pada sistem ini (Kandir, 2018; Lin et al., 2021). OpenCV adalah *library* perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara *real-time* (Kandir, 2018).

Aturan waktu lampu hijau yang di buat pada sistem ini dibagi menjadi 5 variabel yaitu lama, sedang, cepat, sangat cepat dan kosong. Berikut aturan waktu lampu hijau pada Sistem Smart Traffic Light yang tertera pada Tabel 1.

Tabel 1. Aturan Waktu Lampu Lalu Lintas

No	Variabel	Waktu	Banyak kendaraan
1	Lama	120 detik	>80
2	Sedang	80 detik	30-80
3	Cepat	50 detik	11-29
4	Sangat Cepat	20 detik	1-10
5	Kosong	0 detik	0

Setelah percobaan model YOLO dilakukan dan berhasil mendeteksi berbagai jenis kendaraan, sistem lalu menghitung jumlah kendaraan yang ada pada setiap jalur. Hasil perhitungan tersebut akan menentukan lamanya waktu lampu hijau menyala di setiap jalur.

Sistem pengelolaan lampu lalu lintas tersebut dibuat menggunakan bahasa pemrograman Python. *Python* adalah bahasa pemrograman tingkat tinggi berbasis teks yang menggunakan campuran kata-kata bahasa Inggris, karakter tanda baca, simbol, dan angka yang mudah dibaca, ditulis dan dipahami (Santoso, 2022:1). Python membantu visualisasi hasil identifikasi yaitu dengan memberi (1) kotak pembatas, yaitu garis persegi pembatas di sekitar objek, (2) nilai *confidence* yaitu ukuran seberapa yakin model bahwa kotak pembatas berisi objek yang diinginkan, dan (3) *class\_id* yaitu label dari objek yang berada di kotak pembatas.

Untuk mengetahui sejauh mana model YOLO yang telah dibangun dapat memprediksi data dengan benar maka diperlukannya evaluasi model. Mengukur evaluasi diperlukan pengujian *confusion matrix* yang terdiri dari akurasi, ketepatan (*precision*) dan keberhasilan (*recall*).

Dalam konteks *confusion matrix*, di mana *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN) dihitung berdasarkan perbandingan antara prediksi model dan keadaan sebenarnya (Hayati et al., 2023:91).

Akurasi merupakan ukuran evaluasi yang umum digunakan untuk mengukur performa suatu model, dimana untuk menentukan sejauh mana nilai aktual cocok dengan nilai hasil prediksi. Semakin tinggi tingkat akurasi, semakin baik kinerja metode yang digunakan. Rumus untuk menghitung akurasi sesuai dengan formula yang menentukan seberapa dekat nilai-nilai aktual dan hasil prediksi. Semakin tinggi nilai akurasi, semakin baik performa metode yang digunakan. Adapun rumus untuk mencari akurasi sebagai berikut (Hayati et al., 2023:91).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Presisi (*precision*) adalah ukuran tingkat keakuratan dalam membandingkan data yang diminta dengan data yang diberikan oleh model. Presisi mengukur sejauh mana data yang diprediksi sebagai positif oleh model sebenarnya merupakan nilai positif. Rumus untuk menghitung presisi adalah sebagai berikut (Hayati et al., 2023:91).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Dalam rumus ini, *True Positive* (TP) adalah jumlah data yang benar diprediksi sebagai positif oleh model, dan *False Positive* (FP) adalah jumlah data yang salah diprediksi sebagai positif oleh model. Presisi memberikan informasi tentang sejauh mana model dapat memberikan prediksi positif yang benar di antara semua prediksi positif yang dilakukannya. Semakin tinggi nilai presisi, semakin sedikit *false positive* yang dibuat oleh model.

*Recall* atau yang juga dikenal sebagai sensitivitas atau tingkat positif benar, mencerminkan sejauh mana sistem berhasil mendeteksi informasi yang relevan. Rumus untuk menghitung *recall* dapat dilihat seperti pada rumus berikut (Hayati et al., 2023:91).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Untuk menentukan Tingkat kepercayaan (*confidence*) dalam deteksi objek menggunakan algoritma YOLO diukur melalui nilai yang

dihasilkan oleh model, yang mencerminkan sejauh mana model yakin bahwa objek tertentu terdeteksi dengan benar dalam gambar (Y. E. K. Ignasius Widira Kristianto, 2022:275).

$$Confidence = \frac{Predicted (Object)}{Predicted} * IOU (Truth) \tag{4}$$

IoU adalah intersection antara kotak yang diprediksi dan ground truth. Jika tidak ada objek yang ada di dalam sel, skor *confidence* harus nol.

### III. HASIL DAN PEMBAHASAN

Data yang digunakan untuk pengujian sistem *traffic light* didapatkan dari video CCTV Dinas Perhubungan (Dishub). Terdapat 4 jalur yang

berbeda yang digunakan sebagai data untuk mendeteksi dan menghitung kendaraan yang lewat dengan durasi 10 menit tiap jalurnya.

Video CCTV yang diambil hanya video pada siang hari dengan kualitas video sangat baik karena tingkat pencahayaan yang tinggi. Untuk mengevaluasi kinerja sistem ini, diperlukan pengujian yang melibatkan pengambilan data dari beberapa sumber. Pengujian dilaksanakan dengan menggunakan video yang menampilkan kondisi jalan, dengan spesifikasi :

1. Durasi video 10 menit.
2. Frame rate 60 fps.
3. Resolusi video 2038 × 1080 piksel.

Beberapa contoh tangkapan layar dapat dilihat pada Gambar 2.



Gambar 2. Contoh tangkapan layar dari video CCTV

Tabel 2. *Performance Precision* Model YOLOv8

Class	Image	Instance	Precision
All	128	929	0,689
Car	128	46	0,84
Motorcycle	128	5	0,612
Bus	128	7	0,748
Truck	128	12	0,889

Sebelumnya, model YOLO yang digunakan pada sistem ini telah memiliki dataset yaitu *coco128.yaml* dan di-training sehingga dapat langsung mengenali objek salah satunya yaitu kendaraan motor atau mobil. Model YOLOv8 ini memiliki 168 layers dan 11156544 parameter dengan training 3 kali epoch. Dapat dilihat model yang digunakan pada tabel 2 dan 3.

Tabel 3. *Performance Recall* dan mAP Model YOLOv8

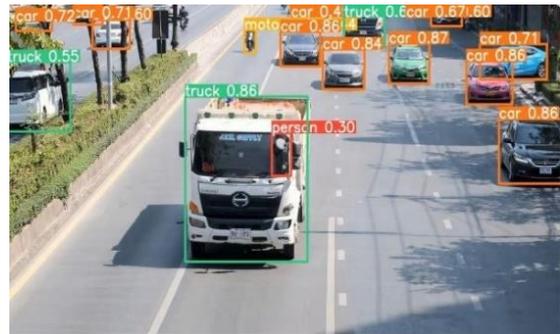
Class	Recall	mAP50	mAP50-95
All	0,578	0,65	0,486
Car	0,217	0,292	0,18
Motorcycle	0,8	0,872	0,709
Bus	0,714	0,721	0,675
Truck	0,5	0,529	0,342

Secara keseluruhan *training* model YOLOv8 ini mendapatkan nilai *precision* sebesar 0,689, *recall* 0,578 dan mAP50 sebesar 0,65. Selanjutnya model YOLO diuji dengan *data test* yang belum dilatih sebelumnya sehingga didapatkan hasil deteksi objek pada model YOLO pada gambar 2. Dari gambar di bawah dapat dilihat bahwa objek mobil (*car*), truk (*truck*), dan motor (*motorcycle*)

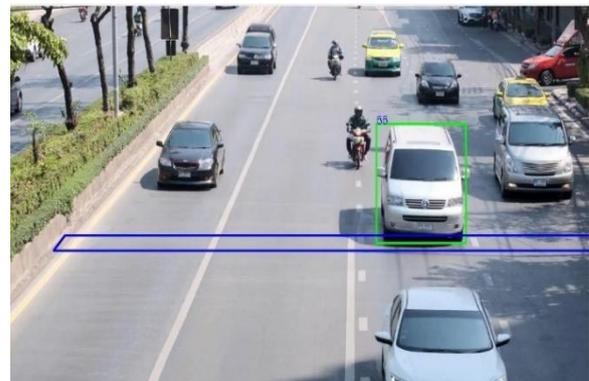
dapat dideteksi dengan baik. Pada *bounding box* di sebelah kanan nama kendaraan terdapat informasi nilai *confidence* yang diperoleh untuk setiap objek yang terdeteksi. Semakin tinggi nilai *confidence* yang didapatkan, semakin tinggi akurasi deteksi objek tersebut.

Seperti yang terlihat pada Gambar 3, model YOLO mendeteksi objek kendaraan truk dimana nilai *confidence* yang didapatkan yaitu 0,86. Hasil *confidence* ini menunjukkan bahwa akurasi deteksi objek tersebut mencapai 86% .

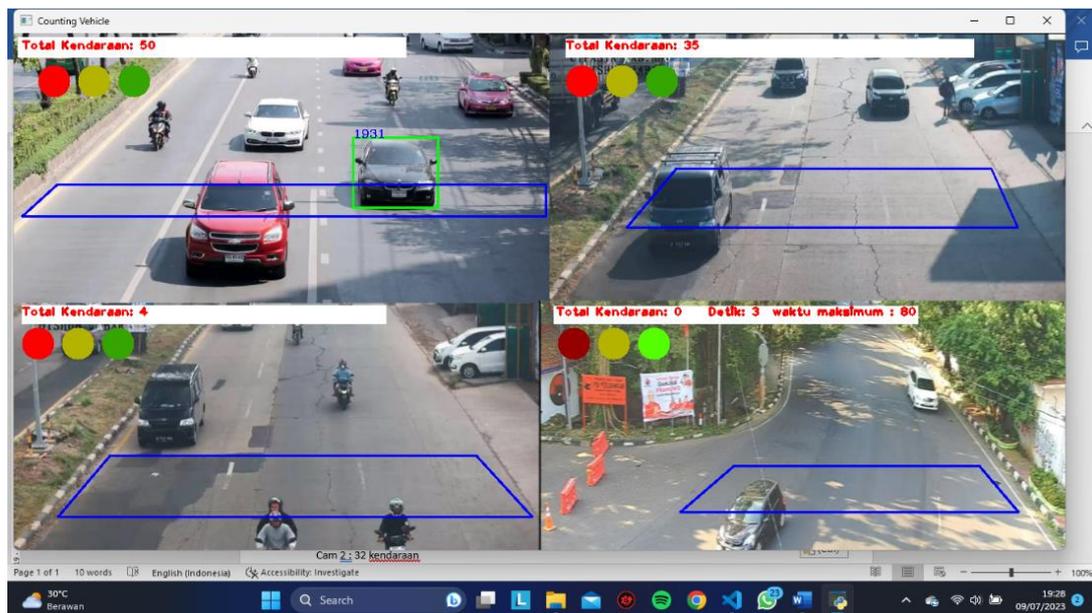
Selanjutnya YOLO yang sudah dapat mendeteksi kendaraan dihubungkan dengan pemrosesan dan analisis menggunakan *Computer Vision* yaitu *library* OpenCV. Tahap ini dibuat sebuah *polyline* pada jalan sehingga ketika kendaraan melewati *polyline* dipastikan kendaraan tersebut akan antri pada lampu lalu lintas di persimpangan (Gambar 4).



Gambar 3. Uji Datatest Menggunakan YOLO



Gambar 4. Pembuatan *Polyline* dengan OpenCV



Gambar 5. Sistem *Smart Traffic Light*

Kendaraan yang melewati batas *polyline* yang telah dibuat menggunakan OpenCV akan terdeteksi dan dihitung. Jumlah kendaraan tersebut akan menjadi input sistem untuk berapa lama waktu lampu hijau yang akan diberikan pada jalur tersebut (Gambar 5). Pemrograman Python digunakan untuk membuat logika pembagian

waktu dengan jumlah kendaraan yang telah terdeteksi oleh model YOLOv8, dimana pembagian waktu diatur pada setiap jalur sesuai dengan jumlah kendaraannya. Ketika jumlah kendaraan pada salah satu jalur di persimpangan melebihi dari 50 kendaraan yang mengantri di lampu merah, maka waktu lampu hijau yang

diberikan maksimal 80 detik untuk melewati persimpangan dengan pembagian lamanya waktu lampu hijau menyala sesuai dengan tabel 1.

Setelah dibuat logika mengenai perhitungan waktu lampu lalu lintas, didapatkan hasil Sistem Smart Traffic Light yang dibuat pada sebuah simulasi terhadap 4 video yang digabung menjadi 1 frame. Setiap video memiliki lampu lalu lintas yang dibuat menggunakan OpenCV yang menggambarkan sebuah persimpangan yang memiliki 4 jalur berbeda. Dapat dilihat pada gambar 4 untuk Sistem Smart Traffic Light pada sebuah persimpangan.

Tempat deteksi dan perhitungan kendaraan ditempatkan sedikit jauh sebelum lampu lalu lintas, hal ini dilakukan agar memaksimalkan deteksi kendaraan yang akan mengantri pada lampu lalu lintas. Sehingga ketika kendaraan lewat akan terdeteksi tanpa terhalang oleh kendaraan lain. Pada persimpangan terdapat 4 jalur, ketika jalur 1 mendapatkan kondisi lampu hijau maka jalur 2, 3 dan 4 mendapatkan kondisi lampu merah, Sistem Smart Traffic Light ini akan mendeteksi dan menghitung jumlah kendaraan yang mengantri pada lampu lalu lintas pada saat jalur mendapatkan lampu merah. Begitupun dengan jalur yang lain, ketika salah satu jalur dalam kondisi lampu hijau maka jalur lain akan melakukan pendeteksian dan perhitungan kendaraan. Misalnya pada jalur 2 mendapatkan lampu hijau hingga selesai, maka jalur 2 akan berubah mendapatkan lampu merah yang kemudian sistem ini akan melakukan perhitungan ulang dengan menghitung kendaraan mulai dari 0. Kemudian pada saat jalur lain mendapat giliran lampu hijau, misalkan jalur 3, maka sistem akan berhenti mendeteksi dan menghitung kendaraan pada jalur tersebut untuk diberikan waktu lampu hijau sesuai dengan berapa banyak kendaraan yang terdeteksi oleh model YOLOv8. Maka sistem ini akan menghentikan deteksi dan perhitungan jumlah kendaraan secara bergantian pada setiap jalur.

Sistem Smart Traffic Light ini berhasil melakukan deteksi dan perhitungan jumlah kendaraan untuk manajemen lampu lalu lintas pada sebuah persimpangan. Dapat dilihat pada tabel 4 untuk hasil uji deteksi Sistem Smart Traffic Light.

Tabel 4. Hasil Pengujian Sistem

Camera	Total Kendaraan	Waktu Lampu Hijau
Cam 2	32 kendaraan	80 Detik
Cam 3	48 Kendaraan	80 Detik
Cam 4	68 Kendaraan	80 Detik
Cam 1	75 Kendaraan	80 Detik
Cam 2	87 Kendaraan	120 Detik
Cam 3	83 Kendaraan	120 Detik
Cam 4	126 Kendaraan	120 Detik
Cam 1	110 Kendaraan	120 Detik
Cam 2	144 Kendaraan	120 Detik
Cam 3	156 Kendaraan	120 Detik
Cam 4	83 kendaraan	120 Detik
Cam 1	110 Kendaraan	120 Detik
Cam 2	116 Kendaraan	120 Detik
Cam 3	87 Kendaraan	120 Detik
Cam 4	134 Kendaraan	120 Detik
Cam 1	93 Kendaraan	120 Detik
Cam 2	120 Kendaraan	120 Detik
Cam 3	93 Kendaraan	120 Detik
Cam 4	116 Kendaraan	120 Detik
Cam 1	107 Kendaraan	120 Detik

Sistem yang menerapkan model YOLOv8 ini telah di-*training* dan memberikan hasil *precision* sebesar 0,689 dan nilai *recall* sebesar 0,578 secara keseluruhan (lihat tabel 2 dan 3). Hasil dari deteksi model YOLOv8 pada Sistem Smart Traffic Light memberikan prediksi yang benar secara akurat sebesar 68,9 % objek positif.

#### IV. PENUTUP

##### Kesimpulan

Sistem pengaturan lampu lalu lintas cerdas menggunakan YOLOv8, berdasarkan visi komputer dan perpustakaan OpenCV telah berhasil dibangun. Model YOLOv8 yang digunakan pada sistem ini memiliki 168 *layers* dan 11156544 parameter dengan *training* 3 kali *epoch*. Hasil perhitungan yang diperoleh adalah (1) nilai *precision* sebesar 0,689, (2) nilai *recall* sebesar 0,578, dan (3) nilai akurasi sebesar 68,9 % sebagai *True Positive*.

Hasil pendeteksian jumlah kendaraan pada setiap jalur diterapkan pada sistem pengaturan lampu lalu lintas di simpang empat. Jumlah kendaraan akan menentukan lamanya lampu hijau menyala secara bergantian pada setiap jalur. Ada 5 kriteria mulai dari 0 kendaraan sampai >80 kendaraan, dimana lampu hijau akan menyala antara 0 sampai 120 detik.

**Saran**

Secara keseluruhan model ini memiliki performa nilai *precision*, *recall* dan mAP50 yang cukup baik, namun nilai *precision* ini masih harus ditingkatkan. Maka disarankan agar pada penelitian lanjutan menggunakan data yang lebih banyak agar terbentuk model yang lebih baik.

**Ucapan Terima Kasih**

Terima kasih kami sampaikan pada Dinas Perhubungan yang telah memberikan data video lalu lintas.

**V. DAFTAR PUSTAKA**

- Al Mudawi, N., Qureshi, A.M., Abdelhaq, M., Alshahrani, A., Alazeb, A., Alonazi, M., & Algarni, A. (2023). Vehicle Detection and Classification via YOLOv8 and Deep Belief Network over Aerial Image Sequences. *Sustainability* 2023, 15, 14597. <https://doi.org/10.3390/su151914597>
- Amwin, A. (2021). Deteksi Dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (YOLO). *Universitas Islam Indonesia*. <https://Dspace.Uii.Ac.Id/Handle/123456789/34154>
- Ardiansyah, M. R., Supit, Y., & Said, M. S. (2022). Sistem Visi Komputer Untuk Kalkulasi Kepadatan Kendaraan Menggunakan Algoritma Yolo. *Simtek : Jurnal Sistem Informasi Dan Teknik Komputer*, 7(1), 52–59. <https://doi.org/10.51876/Simtek.V7i1.123>
- Cholissodin, I., & Soebroto, A. A. (2021). *Buku Ajar AI , Machine Learning & Deep Learning ( Teori & Implementasi )* (1.01, Issue July 2019). <https://www.researchgate.net/publication/348003841>
- Dian Maniswari, S., Rusdinar, A., & Purnama, B. (2015). Smart Traffic Light Using Image Processing And Fuzzy Logic Method. *E-Proceeding Of Engineering*, 2(2), 2166.
- Fali Oklilas, A., Dwinta, D., Shofi, G., Putri Mariza, N., Arum Kinanti, S., & Amanda Sari, Y. (2023). Akurasi Pengujian Model Hasil Training Menggunakan Yolov4 Untuk Pengenalan Kendaraan Di Jalan Raya. *Jurnal JUPITER*, 15(1), 799–806.
- Gibran, H., Purnama, B., & Kosala, G. (2023). Object Tracking Menggunakan Algoritma You Only Look Once (Yolo)v8 Untuk Menghitung Kendaraan. *Jurnal Komputa*, 12(2), 31–45.
- Harahap, E., Suryadi, A., Ridwan, R., Darmawan, D., & Ceha, R. (2017). Efektifitas Load Balancing Dalam Mengatasi Kemacetan Lalu Lintas. *Matematika*, 16(2), 1–7. <https://doi.org/10.29313/Jmtm.V16i2.3665>
- Harahap, M., Elfrida, J., Agusman, P., Rafael, M., Abram, R., Andrianto, K., Kunci-Visi Komputer, K., Arus Lalu Lintas, P., & Kendaraan, D. (2019). Sistem Cerdas Pemantauan Arus Lalu Lintas Dengan YOLO (You Only Look Once V3). In *Seminar Nasional APTIKOM*.
- Hayati, N. J., Singasatia, D., & Muttaqin, M. R. (2023). Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 Untuk Menghitung Kendaraan. *Komputa : Jurnal Ilmiah Komputer Dan Informatika*, 12(2), 91–99. <https://doi.org/10.34010/Komputa.V12i2.10654>
- Hidayati, Q. (2017). Kendali Lampu Lalu Lintas Dengan Deteksi Kendaraan Menggunakan Metode Blob Detection. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 6(2). <https://doi.org/10.22146/Jnteti.V6i2.318>
- Kandir, N. (2018). *Opencv Dengan Python*. <https://doi.org/10.1128/AAC.03728-14>
- Lin, C. J., Jeng, S. Y., & Lioa, H. W. (2021). A Real-Time Vehicle Counting, Speed Estimation, And Classification System Based On Virtual Detection Zone And YOLO. *Mathematical Problems In Engineering*, 2021. <https://doi.org/10.1155/2021/1577614>
- Rachmawati, F., & Widhyaestoeti, D. (2020). Deteksi Jumlah Kendaraan Di Jalur SSA Kota Bogor Menggunakan Algoritma Deep Learning YOLO. In *Prosiding LPPM UIKA Bogor*. <http://pkm.uika-bogor.ac.id/index.php/prosiding/index>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings Of The IEEE Computer Society Conference On Computer Vision And Pattern Recognition, 2016-Decem*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Reis, D., Kupec, J., Hong, J., & Daoudi, A. (2023). Real-Time Flying Object Detection With Yolov8. *Georgia Institute Of Technology*. <http://arxiv.org/abs/2305.09972>

- Santoso, J. T. (2022). *Proyek Coding Dengan Python* (M. K. Muhammad Sholikan (Ed.)). Yayasan Prima Agus Bekerja Sama Dengan universitas Sains & Teknologi Komputer (Universitas STEKOM).
- Saputra, B. M., Ilman, M. Z., Audina, M., & Jepri, M. (2023). Sistem Pengenalan Tanda Lalu Lintas Menggunakan Algoritma YOLO. *Jurnal Inovasi Dan Humaniora*, 1(1), 161–164.
- Setianingsih, C., & Paryasto, M. W. (2022). Sistem Deteksi Pelanggaran Zebra Cross Pada Kendaraan Sepeda Motor Menggunakan Algoritma Yolov4 Cross Zebra Violation Detection System On Motorcycle Vehicles Using The Yolov4 Algorithm. *E-Proceeding Of Engineering*, 9(5), 5038–5045.
- Sunaryo, & Kusumawati, Natalia Reza. (2020). Evaluasi Pembangunan Median Jalan. *Jurnal teknologi Transportasi Dan Logistik*, 1(1), 11–14.
- Y. E. K. Ignasius Widira Kristianto. (2022). Pendeteksian Sepeda Motor di Jalur Khusus Sepeda Menggunakan Algoritma Pendeteksi Objek YOLO. *Kalbisiana: Jurnal Mahasiswa Institut Teknologi Dan Bisnis Kalbis*, 8(1), 275–281.