

Perancangan Kontrol Adaptif Kepadatan Arus Lalu Lintas dengan Metode *Convolutional Neural Network*

Muhammad Ilham Bintang Adhisatria¹, Wahyudi²

^{1,2}Departemen Teknik Elektro, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Soedarto No.13, Kec. Tembalang, Kota Semarang, Jawa Tengah 50275
e-mail: adhisatria994@students.undip.ac.id

Abstrak— Kemacetan lalu lintas merupakan salah satu permasalahan dan tantangan yang memerlukan solusi berbasis teknologi cerdas. Perancangan ini membentuk sistem kontrol adaptif kepadatan arus lalu lintas menggunakan metode *Convolutional Neural Network* (CNN) dan pendekatan algoritme YOLOv8. Sistem ini mengintegrasikan teknologi *computer vision* dan *deep learning* untuk mendeteksi, menghitung, dan mengklasifikasikan kendaraan secara *real-time*, berbasis data citra digital kamera CCTV. Perancangan dilakukan dengan data *input* salah satu pertigaan Kota Malang pada dua waktu dan kondisi yang berbeda. Perhitungan matematis dan *training* data juga dilakukan menggunakan *software* komputasi MATLAB. Pada hasil perancangan, didapatkan nilai akurasi pada rentang 30 – 45%. Dengan sekuensial pertama, prioritas lalu lintas jalur 3, jalur 2, lalu jalur 1. Kemudian pada sekuensial kedua, prioritas lalu lintas jalur 3, jalur 1, dan jalur 2. Meskipun nilai akurasi terbatas, perancangan ini berpotensi bahwa teknologi *deep learning* dapat mengoptimalkan manajemen lalu lintas. Dengan faktor yang berpengaruh seperti kualitas data *input*, pemrograman sistem, kondisi cuaca, dan sudut pembacaan kamera.

Kata kunci: *Lalu Lintas, Deep learning, CNN, YOLOv8, Kontrol Adaptif.*

Abstract— *Traffic jams are one of the issues and challenges that requires an intelligent technology-based solution. This design creates an adaptive traffic density control system using the Convolutional Neural Network (CNN) method and YOLOv8 algorithm approach. This system integrates computer vision and deep learning technologies to detect, count, and classify vehicles in real-time, based on digital image data from CCTV cameras. The design was conducted with input data from an intersection in Malang City at two different times and conditions. Mathematical calculations and data training were also performed using MATLAB computational software. The design results showed an accuracy value around 30-45%. At the first sequential, traffic priority was lane 3, lane 2, then lane 1. In the second sequential, traffic priority changed to lane 3, lane 1, and lane 2. Despite limited accuracy, this design demonstrates the potential of deep learning technology to optimize traffic management. Influencing factors include input data quality, system programming, weather conditions, and camera reading angles.*

Keywords: *Traffic, Deep learning, CNN, YOLOv8, Adaptive Control.*

I. PENDAHULUAN

Jalan adalah sebuah lintasan yang dibuat untuk memudahkan transportasi dalam perjalanan dari satu tempat menuju tempat lainnya. Seiring dengan berjalannya waktu, jumlah volume kendaraan di Indonesia semakin meningkat dan menyebabkan kemacetan di berbagai lokasi ruas jalan. Akibat kemacetan, banyak waktu yang hilang dan berkurangnya kesempatan masyarakat untuk memperoleh pendapatan[1]. Perlu adanya sistem tracking lalu lintas untuk memantau kondisi kepadatan lalu lintas. *Area Traffic Control System* (ATCS) merupakan suatu sistem yang mengatur lalu lintas dengan sinyal yang memiliki koordinasi melingkupi suatu area terpusat[2].

Sistem lampu lalu lintas saat ini masih menerapkan penghitung waktu. Salah satu kekurangannya pada sistem penghitung waktu ini adalah kondisi waktu yang tidak sesuai berdasarkan kepadatan jumlah kendaraan pada persimpangan jalur yang ada[3]. Dalam perancangan ini, dilakukan pendekatan kontrol *neural network* adaptif untuk mencapai kontrol yang akurat dan kuat dari sistem non-linier dengan dinamika yang tidak diketahui, dimana pengontrol tidak lagi berisi operasi anti-perhitungan *input* pada metode berbasis model yang ada[4].

Adanya peran penting lalu lintas adalah untuk manusia dapat melakukan aktivitas mobilitas, seperti berkendara dengan lancar[5]. Untuk mempersiapkan dan menjawab kebutuhan, maka diperlukan suatu sistem yang mampu

memonitoring kendaraan pada ruas jalan yang secara otomatis mengklasifikasikan dan menghitung jumlah kendaraan untuk menggantikan proses manual pencatatan jumlah kendaraan yang digunakan sebagai data perencanaan kapasitas kepadatan arus lalu lintas melalui pengolahan citra digital kamera CCTV lampu lalu lintas dalam upaya mencegah kepadatan arus lalu lintas[6]. Maka dari itu *artificial intelligent* diperlukan untuk mendeteksi pelanggaran tersebut untuk menggantikan tenaga manusia[7].

Dalam beberapa kurun waktu terakhir, observasi tentang kecerdasan buatan, seperti pendeteksian objek, mampu memudahkan para peneliti dalam mengetahui dan memahami objek pada gambar[8]. Sehingga, dibutuhkan sebuah mekanisme lampu lalu lintas secara adaptif dan cerdas dalam mengatur pembagian jumlah waktu sinyal berdasarkan kondisi aktual lalu lintas[9].

Sebuah sistem yang dinamakan *intelligent transportation system*, merupakan perpaduan sistem antara jalan, manusia, dan kendaraan yang memanfaatkan *state of the art* teknologi informasi[10]. Dengan data jumlah kendaraan secara *real-time*, dapat diketahui untuk memprediksi kemacetan, mengkalkulasi kepadatan kendaraan, serta sebagai rujukan memperbaiki kondisi jalan, seperti dengan melebarkan jalanan, menambah jalur baru, peningkatan infrastruktur, dan menata kondisi petunjuk lalu lintas[11]. Kondisi lalu lintas mampu didapatkan solusi efektif melalui pra-kiraan volume kendaraan yang melewati persimpangan dengan rentang waktu tertentu[12].

Perancangan kontrol ini terdiri dari inisiasi data, perancangan, dan *training* data. Pada bagian *training* data, masa estimasi hasil keluaran diakibatkan oleh *training* data, nilai *neuron* yang tak tampak, serta nilai *epoch*, dengan semakin rendah tingkat kepentingan belajar, akan semakin panjang proses pembelajarannya[13]. Dalam merancang suatu jaringan sesuai pada keterkaitan dengan permasalahan, berpengaruh pada keberhasilan tujuan yang ditargetkan[14]. Dengan adanya kontrol adaptif menggunakan metode CNN melalui bahasa pemrograman *Python*, menghasilkan solusi aktual dalam mengatasi kepadatan lalu lintas. Sehingga, diperlukan sebuah sistem efektif dalam pengkalkulasian jumlah volume dan panjang antrian setiap jalur pada persimpangan secara *real-time*[15], sebagai bentuk rekayasa solusi dari permasalahan kondisi lalu lintas.

II. STUDI PUSTAKA

A. Deep Learning

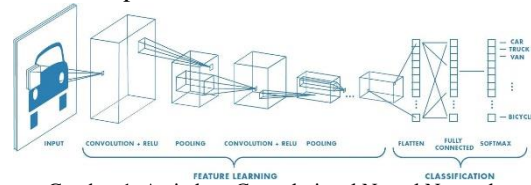
Deep learning adalah subwilayah pembelajaran mesin untuk menyelesaikan permasalahan algoritme lebih cepat dan sesuai kondisi. Dengan berkembangnya kemampuan komputasi, menjadi sebuah penyebab *deep learning* berkembang yang memudahkan pembuatan model *deep neural network* yang memiliki sejumlah tingkatan *hidden layer*[7].

B. Convolutional Neural Network (CNN)

CNN adalah pendekatan *deep learning* dalam mengolah data yang memiliki pola kisi-kisi/jaringan, misalkan suatu citra yang terbangun berdasarkan sejumlah piksel untuk dibaca perangkat, seperti komputer melalui kondisi rupa matriks. Algoritme CNN merupakan algoritme pengembangan *neural network* yang memiliki kompleksitas terhadap *layer*-nya. Termasuk dalam kategori *deep learning*

yang memanfaatkan *layer* konvolusi dalam matriks, baik 2 dimensi maupun 3 dimensi[6].

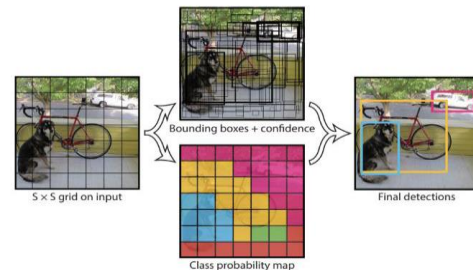
CNN dirancang bangun sedemikian rupa agar secara adaptif dan otomatis mampu mempelajari tingkatan spasial berdasarkan suatu fitur dari citra yang memiliki pola sederhana hingga ke pola yang lebih rumit. Lapisan ini memiliki fungsi dasar, seperti normalisasi, konvolusi, non-linier, pengumpulan, dan *layer* terhubung penuh[7]. Gambar 1 merupakan arsitektur dari CNN.



Gambar 1. Arsitektur Convolutional Neural Network

C. You Only Look Once (YOLO)

YOLO merupakan suatu jaringan syaraf cerdas dalam menerapkan pendeteksian objek secara *real-time*. YOLO melakukan penerapan jaringan syaraf tunggal untuk semua gambarnya. Cara kerjanya dengan membagi gambar menjadi beberapa area, lalu memprediksi kotak pembatas dan tingkat probabilitasnya[3]. Kaidah ini merupakan salah satu cara akurat dan cepat untuk deteksi objek hingga 2 kali lebih cepat daripada algoritme lain[8]. Ilustrasi algoritme diperlihatkan pada Gambar 2.



Gambar 2. Ilustrasi Algoritme YOLO (You Only Look Once)

D. OpenCV

OpenCV merupakan suatu pustaka dari *open source* pada pembelajaran mesin dan *computer vision*. OpenCV dirancang dengan infrastruktur yang mencadangkan pengaplikasian serta akselerasi mengenai pengolahan citra digital dengan sangat mudah[6].

E. MATLAB (Matrix Laboratory)

MATLAB merupakan *software* penerapan matriks dasar pada penggunaannya. Matriks pada MATLAB dapat dikatakan sederhana karena mudah untuk dipakai. MATLAB memperoleh dan mengolah data dalam bentuk citra CCTV, untuk masukan pengolahan citra digital, pada langkah sebelum pemrosesan MATLAB menerapkan kaidah deteksi kendaraan[5].

Perangkat lunak ini memanfaatkan bahasa pemrograman tingkat tinggi yang khusus dalam melakukan *technical computing*, *visualization*, dan komputasi matematis, *data analysis*, pengembangan bentuk algoritma, pensimulasian serta model grafik kalkulasi[5].

F. Confusion Matrix

Matriks kebingungan merupakan suatu tabel pada kinerja suatu model pembagian kelas untuk mendapatkan data *right answer* (*supervise*)[2]. Beberapa variabel pada *confusion matrix* seperti *precision*, *accuracy*, *f-score*, *recall*, serta lainnya, menyesuaikan bentuk data untuk diklasifikasikan. Tabel matriks kebingungan terdiri dari 4 ketentuan, yakni:

1. Positif Benar (TP), kelas model dibagi menjadi Benar, yang kelas aktualnya adalah Benar.
2. Negatif Benar (TN), kelas model dibagi menjadi Salah, yang kelas aktualnya adalah Salah.
3. Positif Salah (FP), kelas model dibagi menjadi Benar, yang kelas aktualnya adalah Salah.
4. Negatif Salah (FN), kelas model dibagi menjadi Salah, yang kelas aktualnya adalah Benar.

Penggunaan persamaan data matriks kebingungan yakni nilai *precision*, *accuracy*, *f-score*, dan *recall*[2].

- a. *Precision* adalah komparasi tingkat data positif dan dikategorikan *true* berdasarkan sistem serta seluruh data terkategori positif, persamaannya sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

- b. *Accuracy* adalah penaksiran seberapa benar sistem mengklasifikasi keseluruhan, persamaannya sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

- c. *F-score* yakni gabungan dari *recall* dan *precision*, persamaannya sebagai berikut:

$$f - Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (3)$$

- d. *Recall* adalah penaksiran data pembagian kelas positif benar pada sistem, persamaannya sebagai berikut:

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

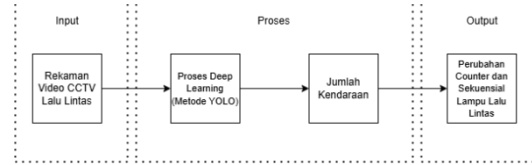
Dalam perancangan kontrol ini dilakukan melalui kajian kepustakaan dan simulasi sederhana menggunakan model arsitektur *neural network* dengan kombinasi adaptif melalui *software* komputasi MATLAB dan bahasa pemrograman *Python*.

III. METODE

Dalam perancangan kontrol ini dilakukan melalui kajian kepustakaan dan simulasi sederhana menggunakan model arsitektur *neural network* dengan kombinasi adaptif melalui *software* komputasi MATLAB dan bahasa pemrograman *Python*. Adapun seri MATLAB dan *Python* yang digunakan adalah MATLAB R2020a dan *Python* versi 3.12.2.

A. Diagram Blok Kinerja Sistem

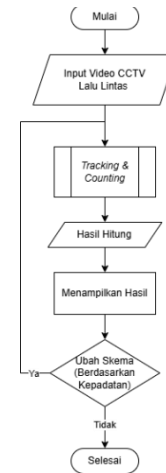
Perancangan dibagi 3 proses, yaitu Masukan, Pemrosesan, dan Keluaran. Pada perancangan ini, sistem lalu lintas dirancang mampu mengkondisikan tingkat volume kepadatan yang memanfaatkan teknologi melalui kaidah *deep learning* dan *computer vision* (AI)[2]. Perancangan ini berbasis menggunakan *software* secara penuh. Blok diagram kinerja sistem diperlihatkan pada Gambar 3.



Gambar 3. Blok Diagram Kinerja Sistem

B. Flowchart Perancangan Skema Sistem

Perkembangan dan kemajuan teknologi teraktual, mampu untuk menciptakan sistem pendeteksi objek dengan perhitungan kendaraan secara otomatis melalui *software*. Di antara yang berkembang pesat adalah *computer vision*, yang telah beredar luas di mancanegara dan sukses diterapkan dalam pendeteksian objek kendaraan, pendeteksi kemacetan, dan lainnya[8]. Hal ini sangat berguna dengan melihat kondisi kemacetan saat ini terjadi di berbagai tempat. Sehingga, hal ini merupakan sebuah solusi, dengan sebuah sistem efektif dalam pengkalkulasian jumlah volume dan panjang antrian setiap jalur pada persimpangan secara *real-time*[15]. Gambar 4 menunjukkan *flowchart* perancangan skema sistem.



Gambar 4. Flowchart Perancangan Skema Sistem

OpenCV dibutuhkan sebagai pemrograman dalam penerapan substraksi latar belakang yang berguna dalam mengambil citra objek berpindah tempat. Lalu, pelacakan dan perhitungan objek yang melalui langkah substraksi latar belakang. Pada tahap akhir, menampilkan perhitungan objek yang sudah dilacak serta dihitung[11].

C. Perhitungan *mean Average Precision* (mAP)

MAP adalah rata-rata nilai *Average Precision* (AP), yang dimanfaatkan untuk model parameter pelatihan dengan dataset yang telah ditentukan. Bentuk hasil dari AP adalah model zigzag dari plot presisi terhadap *recall* seperti pada persamaan (2) terhadap (3). Dengan dituliskan menjadi persamaan berikut.

$$P_{inter}(r_{n+1}) = \max p('r'); r' \geq r_{n+1} \quad (5)$$

AP diperoleh melalui perhitungan wilayah kurva dari hasil *recall*, saat hasil *precision* maksimal terjatuh. Maka wilayah terhitung adalah wilayah saat hasil zigzag diperhalus. Kemudian, AP diperoleh melalui persamaan (6).

$$AP = \frac{1}{11} \times (APr(0) + APr(0,1) + \dots + APr(1,0)) \quad (6)$$

Selanjutnya, dalam mencari hasil mAP, dilakukan perhitungan melalui persamaan (7).

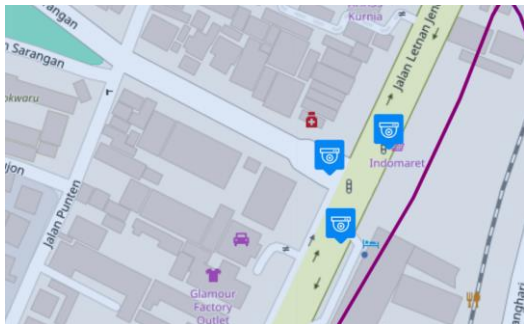
$$mAP = \sum_{i=1}^N \frac{AP_i}{N} \times 100\% \quad (7)$$

Dengan nilai n adalah banyak kelas dari model yang dilakukan *training*.

IV. HASIL DAN PEMBAHASAN

A. Komputasi menggunakan MATLAB

Pengujian perancangan menggunakan data lalu lintas yang diinput berdasarkan video rekaman CCTV lalu lintas kendaraan pada pertigaan Jalan raya Gempol-Malang dengan Jalan Sarangan yang berada di Kota Malang, diambil pada 2 waktu, yakni siang (dengan kondisi hujan) dan malam hari (kondisi setelah hujan). Lokasi perancangan yang digunakan ditunjukkan pada peta dengan Gambar 5.



Gambar 5. Lokasi Data Video CCTV Sebagai Input Perancangan Sistem

Dilakukan perancangan 2 sekuensial berdasarkan video input menggunakan *software* komputasi MATLAB, sehingga didapatkan model rekomendasi sekuensial ditunjukkan pada Tabel 1 dan model implementasi siklus sekuensial ditunjukkan oleh Tabel 2. Berdasarkan Tabel 1 dan Tabel 2, nilai rekomendasi adalah nilai yang disesuaikan dengan kepadatan lalu lintas yang terbaca berdasarkan masing-masing jalur pada pertigaan, di mana masing-masing kepadatannya dikonversi menjadi *digital counter* secara adaptif untuk penyesuaian kondisi lalu lintas.

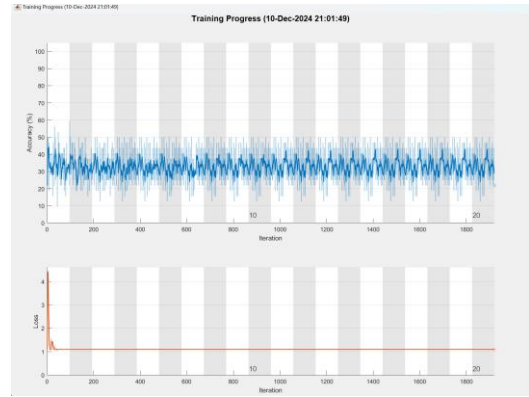
Tabel 1. Rekomendasi Sekuensial Lalu Lintas

| No. | Rekomendasi | |
|-----|-------------|----|
| | Sekuensial | |
| | 1 | 2 |
| J1 | 25 | 25 |
| J2 | 25 | 25 |
| J3 | 20 | 20 |

Tabel 2. Implementasi Siklus Sekuensial Lalu Lintas

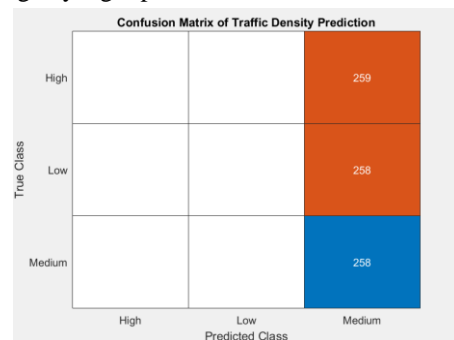
| No. | Implementasi | | | | | |
|--------|--------------|----|----|----|----|----|
| | Sekuensial | | | | | |
| | 1 | | | 2 | | |
| Siklus | 1 | 2 | 3 | 1 | 2 | 3 |
| J1 | 25 | 25 | 25 | 25 | 25 | 25 |
| J2 | 25 | 25 | 25 | 25 | 25 | 25 |
| J3 | 20 | 20 | 20 | 20 | 20 | 20 |

Bentuk nilai rekomendasi diubah dan diimplementasikan pada masing-masing jalur sesuai kebutuhan siklus sekuensial, dengan perhitungan matematis sistem pada MATLAB, didapatkan grafik progres *accuracy* serta *loss training data* pada Gambar 6.



Gambar 6. Progres Accuracy serta Loss Training Data

Gambar 6 menjelaskan tentang grafik akurasi (*accuracy*) dan kerugian (*loss*) model, dari data yang diinputkan selama masa pelatihan / *training* data pada setiap *epoch*. Sumbu x adalah jumlah iterasi dan sumbu y adalah tingkatan akurasi dalam persentase sesuai dengan yang tertera pada grafik. Berdasarkan fluktuasi akurasi yang ditampilkan pada grafik, rentang akurasi bernilai antara 30 – 45%. Sementara grafik yang bawah adalah fungsi kerugian (*loss*) seberapa baik model kinerja sistem. Dapat dilihat pada awal grafik terdapat lonjakan yang tinggi, hal ini disebabkan oleh penyesuaian data dengan input pada iterasi pertama, kemudian pada iterasi selanjutnya terjadi kestabilan sistem dan kinerja mencapai titik jenuh pada nilai $n = 1$. Kemudian, untuk memahami seberapa cocok pengklasifikasian data dalam kelas yang benar, dilakukan representasi matriks kebingungan yang diperlihatkan oleh Gambar 7.



Gambar 7. Matriks Kebingungan (Confusion Matrix)

Merujuk pada Gambar 7, adalah penjelasan dari *confusion matrix*, dimana pada pengklasifikasian kelas *low*, *medium*, dan *high*. Hal ini merepresentasikan bahwa model memprediksi sejumlah data dengan kelas tingkatan “*medium*” sebagai identifikasi data yang cocok. Karena, pada kelas *high* dan *low* kurang terjadi ketepatan data. Namun, jika diperhatikan kembali pada Gambar 6 tentang grafik akurasi data, penggunaan data baru pada rentang akurasi di 30 – 45% akurasi data, dengan rata-rata di 33.29%. Selanjutnya dilakukan pengujian kepadatan lalu lintas menggunakan metode CNN.

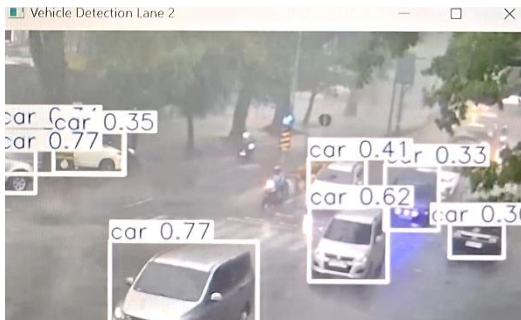
B. Pengujian Kepadatan Lalu Lintas menggunakan CNN Berbasis YOLOv8

Dalam perancangan kontrol adaptif untuk mengatasi kepadatan lalu lintas, diperlukan tahapan konvolusional dengan metode CNN, dalam hal ini menggunakan bahasa pemrograman *Python*.

Skema yang digunakan pada perancangan kontrol adaptif kepadatan lalu lintas kali ini adalah menggunakan 2 kali sekuensial waktu, yakni waktu siang dan malam. Adapun inisiasi data input rekaman video CCTV kendaraan lalu lintas yang digunakan diambil dari CCTV Diskominfo Kota Malang[16]. Gambar 8 – Gambar 10 adalah sekuensial pertama pada salah satu pertigaan Kota Malang, dengan jalur 1 adalah Jalan Sarangan, jalur 2 adalah Jalan Letjend. Sutoyo arah utara, dan jalur 3 adalah Jalan Letjend. Sutoyo arah selatan (Hotel Savana).



Gambar 8. Jalur 1 Sekuensial Pertama



Gambar 9. Jalur 2 Sekuensial Pertama



Gambar 10. Jalur 3 Sekuensial Pertama

Setelah inisiasi data input pada sekuensial pertama (waktu siang hari dan hujan), dilakukan *training* data melalui metode CNN dengan YOLOv8, didapatkan hasil *output*-nya yang ditunjukkan oleh Gambar 11.

```

Hasil Analisis Sekuensial Pertama:

Lane 1:
Total Kendaraan: 30
Tingkat Kepadatan: Rendah

Lane 2:
Total Kendaraan: 133
Tingkat Kepadatan: Sedang

Lane 3:
Total Kendaraan: 150
Tingkat Kepadatan: Sedang

Rekomendasi Urutan Lampu Hijau (Sekuensial Pertama):
Urutan 1: Lane 3
Urutan 2: Lane 2
Urutan 3: Lane 1

Simulasi Lampu Lalu Lintas (3 Perputaran):

Perputaran 1:
Lane 3 - Lampu Hijau: 25 detik (Kepadatan: Sedang)
Lane 2 - Lampu Hijau: 25 detik (Kepadatan: Sedang)
Lane 1 - Lampu Hijau: 20 detik (Kepadatan: Rendah)

Perputaran 2:
Lane 3 - Lampu Hijau: 25 detik (Kepadatan: Sedang)
Lane 2 - Lampu Hijau: 25 detik (Kepadatan: Sedang)
Lane 1 - Lampu Hijau: 20 detik (Kepadatan: Rendah)

Perputaran 3:
Lane 3 - Lampu Hijau: 25 detik (Kepadatan: Sedang)
Lane 2 - Lampu Hijau: 25 detik (Kepadatan: Sedang)
Lane 1 - Lampu Hijau: 20 detik (Kepadatan: Rendah)
    
```

Gambar 11. Hasil Analisis Adaptif Sekuensial 1

Jika diperhatikan secara pembacaan program melalui metode CNN menggunakan YOLOv8, Gambar 11 menunjukkan hasil pembacaan pada jalur 1 terhitung rendah, jalur 2 terhitung sedang, dan jalur 3 terhitung sedang. Meskipun sama memiliki karakter sedang, dengan jumlah kendaraan yang berbeda, jalur 3 lebih diprioritaskan karena objek yang dideteksi lebih banyak. Sehingga menghasilkan siklus sekuensial 1 dengan urutan jalur 3, jalur 2, kemudian jalur 1.

Perputaran pada analisis tersebut adalah siklus lampu merah yang terjadi selanjutnya. Apabila nilai input yang digunakan sama atau kondisi jalur tetap memiliki kepadatan yang tidak berubah signifikan, maka siklus lalu lintas akan tetap sama mengurai kepadatan yang terjadi pada jalur yang paling padat.

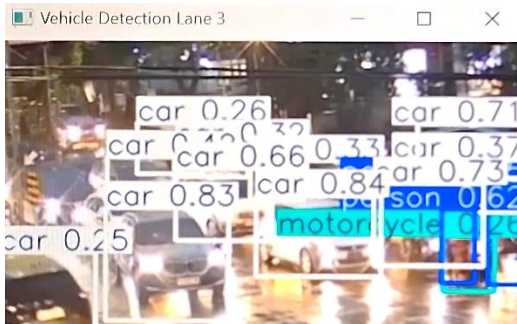
Namun, terdapat hal yang harus diperhatikan pada sekuensial pertama, yakni kondisi lalu lintas yang kurang jelas terbaca secara detail dan kondisi hujan. Hal ini sangat berpengaruh pada pembacaan yang tidak akurat secara total karena terdapat gangguan dari faktor eksternal. Selanjutnya dilakukan pendeteksian kepadatan lalu lintas pada sekuensial kedua yang ditunjukkan oleh Gambar 12 – Gambar 14, dengan jalur 1, 2, dan 3 masih sama namun dengan kondisi waktu dan cuaca berbeda.



Gambar 12. Jalur 1 Sekuensial Kedua



Gambar 13. Jalur 2 Sekuensial Kedua



Gambar 14. Jalur 3 Sekuensial Kedua

Dilakukan data *input* yang kedua pada sekuensial kedua, dengan waktu malam dan kondisi setelah hujan. Kemudian dilakukan *training* data melalui metode CNN dengan YOLOv8, didapatkan hasil *output*-nya yang ditunjukkan oleh Gambar 15.

```

Hasil Analisis Sekuensial Kedua:

Lane 1:
Total Kendaraan: 94
Tingkat Kepadatan: Rendah

Lane 2:
Total Kendaraan: 89
Tingkat Kepadatan: Rendah

Lane 3:
Total Kendaraan: 141
Tingkat Kepadatan: Sedang

Rekomendasi Urutan Lampu Hijau (Sekuensial Kedua):
Urutan 1: Lane 3
Urutan 2: Lane 1
Urutan 3: Lane 2

Simulasi Lampu Lalu Lintas (3 Perputaran):

Perputaran 1:
Lane 3 - Lampu Hijau: 25 detik (Kepadatan: Sedang)
Lane 1 - Lampu Hijau: 20 detik (Kepadatan: Rendah)
Lane 2 - Lampu Hijau: 20 detik (Kepadatan: Rendah)

Perputaran 2:
Lane 3 - Lampu Hijau: 25 detik (Kepadatan: Sedang)
Lane 1 - Lampu Hijau: 20 detik (Kepadatan: Rendah)
Lane 2 - Lampu Hijau: 20 detik (Kepadatan: Rendah)

Perputaran 3:
Lane 3 - Lampu Hijau: 25 detik (Kepadatan: Sedang)
Lane 1 - Lampu Hijau: 20 detik (Kepadatan: Rendah)
Lane 2 - Lampu Hijau: 20 detik (Kepadatan: Rendah)
    
```

Gambar 15. Hasil Analisis Adaptif Sekuensial Kedua

Gambar 15 menunjukkan hasil pembacaan pada jalur 1 terhitung rendah, jalur 2 terhitung rendah, dan jalur 3 terhitung sedang. Meskipun sama memiliki karakter rendah, dengan jumlah kendaraan yang berbeda, jalur 1 lebih diprioritaskan karena objek yang dideteksi lebih banyak dibanding jalur 2. Sehingga menghasilkan siklus sekuensial kedua dengan urutan jalur 3, jalur 1, kemudian jalur 2.

Siklus yang terjadi pada sekuensial kedua ini pun berbeda dari sekuensial pertama. Hal ini juga berdasarkan

kondisi pembacaan sistem yang berbeda, tingkat akurasi dan presisinya juga berbeda.

Pada sekuensial kedua ini juga perlu diperhatikan, karena pada sekuensial ini dilakukan pada waktu malam hari dengan kondisi lalu lintas setelah hujan. Hal ini juga dapat berpengaruh pada pembacaan CCTV dan karakter data yang dibutuhkan untuk mengenali kondisi lampu lalu lintas secara adaptif.

V. KESIMPULAN

Perancangan kontrol adaptif pada kepadatan lalu lintas dengan menggunakan metode CNN menghasilkan sebuah sistem kontrol adaptif berbasis teknologi *computer vision* (AI) menggunakan kaidah *deep learning*. Metode ini berfungsi dalam penyelesaian masalah kepadatan arus lalu lintas. Melalui pendekatan *Convolutional Neural Network* (CNN) dan algoritme YOLO, sistem mampu mendeteksi, menghitung, serta mengklasifikasikan kendaraan secara *real-time*.

Hasil perancangan dengan metode ini dapat mengatur kondisi siklus lampu lalu lintas setiap jalur yang dinamis dengan tingkat akurasi berada pada rentang 30 – 45%. Pada sekuensial pertama berdasarkan data *input*, didapat prioritas lalu lintas dari jalur 3, lalu jalur 2, dan jalur 1. Kemudian pada sekuensial kedua, prioritas lalu lintas dari jalur 3, lalu jalur 1, dan jalur 2.

Perancangan ini membuktikan potensi penggunaan teknologi *deep learning* metode CNN dengan algoritme YOLO untuk mengoptimalkan manajemen lalu lintas, walaupun dari tingkat akurasi masih sangat rendah untuk diimplementasikan. Namun, tingkat akurasi yang rendah juga terdapat terjadi karena adanya faktor internal seperti kualitas data *input* dan pemrograman sistem, maupun faktor eksternal seperti kondisi cuaca dan posisi sudut pembacaan kamera.

REFERENSI

- [1] G. P. R. Pingky, Mat Syai'in, Zindhu Maulana Ahmad Putra, Ahmad Putra, Ii Munadhif, and Imam Sutrisno, "Skema Koordinasi Persimpangan Untuk Kelancaran Arus Lalu Lintas Dengan Metode Neural Network," *J. Elektron. dan Otomasi Ind.*, vol. 11, no. 2, pp. 396–408, 2024, doi: 10.33795/elkolind.v11i2.5367.
- [2] M. I. Hadi, D. K. Silalahi, and P. D. Wibawa, "Pengaturan Lampu Lalu Lintas Berdasarkan Deteksi Volume Kendaraan Menggunakan Metode Yolov3 Traffic Light Setting Based On Vehicle Volume Detection Using The Yolov3 Method," *e-Proceeding Eng.*, vol. 9, no. 5, pp. 2133–2144, 2022, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/18470>
- [3] Irfan Hermawan M, Iwut Tritoasmoro I, and Ibrahim N, "Traffic Light Control Based on Vehicle Density Using the Yolo Method," *e-Proceeding Eng.*, vol. 8, no. 1, pp. 198–205, 2021.
- [4] L. Cheng, Z. Wang, F. Jiang, and J. Li, "Adaptive neural network control of nonlinear systems with unknown dynamics," *Adv. Sp. Res.*, vol. 67, no. 3, pp. 1114–1123, 2021, doi: 10.1016/j.asr.2020.10.052.
- [5] T. Pangemanan and A. Rondonuwu, "Perancangan Sistem Kontrol Lampu Lalulintas Cerdas Dengan Menggunakan Mikrokontroler dan Kamera," *J. MIPA*, vol. 8, no. 3, p. 200, 2019, doi: 10.35799/jmuo.8.3.2019.26198.

- [6] H. T. S. Abid Juliant Indraswara, Bagus Fathkurrozi, "PERANCANGAN SISTEM MACHINE LEARNING UNTUK KLASIFIKASI KENDARAAN PADA PERSIMPANGAN TRAFFIC LIGHT," 2022.
- [7] B. F. Darmanto, C. Setianingsih, and R. E. Saputra, "Deteksi Pelanggaran pada Bahu Jalan Tol Menggunakan Algoritma Cascade R-CNN," *e-Proceeding Eng.*, vol. 10, no. 5, pp. 4488–4497, 2023.
- [8] D. A. Abdurrafi, M. T. Alawiy, and B. M. Basuki, "Deteksi Klasifikasi Dan Menghitung Kendaraan Berbasis Algoritma You Only Look Once (YOLO) Menggunakan Kamera CCTV," *Sci. Electro*, no. 1, pp. 1–9, 2023, [Online]. Available: <https://jim.unisma.ac.id/index.php/jte/article/viewFile/21551/16069>
- [9] A. N. A. Yusuf, A. S. Arifin, and F. Y. Zulkifli, "Recent development of smart traffic lights," *IAES Int. J. Artif. Intell.*, vol. 10, no. 1, pp. 224–233, 2021, doi: 10.11591/ijai.v10.i1.pp224-233.
- [10] A. Hendriawan, M. I. M. Pradana, and R. Susetyoko, "Sistem Deteksi Lampu Lalu Lintas Sebagai Asisten Pengemudi Menggunakan Convolutional Neural Network," *INOVTEK Polbeng - Seri Inform.*, vol. 8, no. 1, p. 73, 2023, doi: 10.35314/isi.v8i1.3155.
- [11] V. P. Saputra and U. Latifa, "Simulasi Detection Counter Pada Objek Kendaraan Motor Dan Mobil Menggunakan Metode Convolutional Neural Network Berbasis Python," *J. Ilm. Wahana Pendidik.*, vol. 2023, no. 16, pp. 760–766, 2023, [Online]. Available: <https://doi.org/10.5281/zenodo.8265040>
- [12] I. Setiawan, "Mengatasi Kemacetan Di Lampu Merah Dengan Pendekatan Image Processing," *J. Innov. Futur. Technol.*, vol. 4, no. 2, pp. 9–18, 2022, doi: 10.47080/iftech.v4i2.2117.
- [13] Suci Wahyuni, "Jaringan Syaraf Tiruan Memprediksi Kendaraan Masuk Pada Pengujian Kir Menggunakan Metode Backpropagation (Studi Kasus : Dinas Perhubungan Kota Binjai)," *Semin. Nas. Inform.*, pp. 135–140, 2021.
- [14] I. Yani, F. F. Siregar, and D. S. Tua Sitorus, "Identifikasi Plat Mobil Dengan Menggunakan Metode Jaringan Syaraf Tiruan Kohonen Pada Sistem Parkir Cerdas," *Pros. Semin. Nas. Teknoka*, vol. 4, no. 2502, pp. I26–I31, 2019, doi: 10.22236/teknoka.v4i0.4205.
- [15] G. N. Laanaila, I. D. Irawati, and D. N. Ramadan, "Smart Traffic Monitoring & Control Dengan Pengolahan Citra Digital," *eProceedings Appl. Sci.*, vol. 9, no. 1, pp. 330–336, 2023.
- [16] CCTV Kota Malang. 2024. http://cctv.malangkota.go.id/cameras_dua