

Deteksi Klasifikasi Kendaraan dan Menghitung Jumlah Pelanggaran Melawan Arah menggunakan Model Yolov5

Ondang Fajrul Falach¹, Fawaidul Badri², Bambang Minto Basuki³

^{1,2,3} Prodi Teknik Elektro, Fakultas Teknik, Universitas Islam Malang

JL. MT. HARYONO 193 MALANG

21901053018@email.com, fawaidulbadri@unisma.ac.id, bambang.minto@unisma.ac.id

Abstract

The rapid growth of vehicles in major Indonesian cities is not matched by adequate road infrastructure, leading to congestion, air pollution, and an increase in traffic accidents. An accidents number more higher are caused by violations rider on the traffic, including driving against the flow. With the video input and YOLOv5 neural model will be manage automatically by the system program on this research and giving result a data of traffic violation are being focus and goals of this research. The system leverages computer vision to detect, classify, and count vehicles in real time.

Due to limited access to real-world violation footage, simulated traffic scenarios were used for system testing. The detection model was trained on four vehicle classes: motorcycles, cars, buses, and trucks. Experimental results show high performance, with 94% precision, 97% recall, and a 96% F1-score. Vehicle counting accuracy reached 93%.

This system demonstrates strong potential to support traffic analysis and assist authorities in making more informed decisions to reduce congestion and traffic accidents.

Keywords— Yolov5, vehicle detection, vehicle classification, Yolov5 SORT, vehicle counting

Abstraksi

Melonjaknya angka penggunaan kendaraan di kota-kota besar tidak diimbangi dengan infrastruktur jalan yang memadai, memicu kemacetan, polusi udara, dan peningkatan angka kecelakaan lalu lintas. Sebagian besar kecelakaan disebabkan oleh pelanggaran, seperti melawan arus. Penelitian ini mengusulkan sistem deteksi otomatis pelanggaran lalu lintas berbasis video menggunakan algoritma YOLOv5. Sistem ini memanfaatkan teknologi **computer vision** agar mampu mengenali dalam bentuk deteksi dan mengklasifikasikan bentuk dari kelas serta menghitung jumlah pelanggaran melawan arah secara real-time oleh pengguna jalan. Karena keterbatasan akses terhadap data video nyata, digunakan simulasi pelanggaran lalu lintas sebagai data uji. Sistem diuji dengan empat kategori kendaraan: motor, mobil, bus, dan truk. Hasil pengujian menunjukkan performa tinggi dengan precision sebesar 94%, recall 97%, dan F1-score 96%. Akurasi perhitungan kendaraan mencapai 93%. Sistem ini menunjukkan potensi besar dalam mendukung analisis lalu lintas serta membantu otoritas dalam pengambilan keputusan yang lebih tepat untuk mengurangi kemacetan dan kecelakaan.

Kata Kunci— Yolov5, deteksi kendaraan, deteksi klasifikasi kendaraan, Yolov5 SORT, kalkulasi kendaraan

I. PENDAHULUAN

Di kota-kota besar pada negara kita, bertumbuhnya jumlah kendaraan pribadi mengalami peningkatan signifikan. Fenomena ini tidak selaras dengan pengembangan fasilitas jalan yang memadai, sehingga menimbulkan berbagai permasalahan seperti kemacetan, peningkatan polusi udara, dan tingginya angka kecelakaan lalu lintas[1][2]. Dari data Badan Pusat Statistik menandakan jumlah kecelakaan terus meningkat setiap tahun, dengan sebagian besar insiden dihasilkan oleh ketidaktaatan pengguna jalan terhadap aturan, seperti menerobos lampu apil dan melawan arus [3][4]. Pelanggaran ini umumnya dilakukan oleh pengendara roda dua, yang mendominasi populasi kendaraan di Indonesia[9].

Sebagai bentuk pengawasan, pemerintah telah menerapkan sistem pemantauan lalu lintas menggunakan CCTV. Meskipun efektif dalam mengumpulkan data, proses analisis dari rekaman tersebut masih bersifat manual, sehingga kurang optimal [10] dalam mendukung pengambilan keputusan secara cepat dan akurat. Seiring perkembangan teknologi,

pendekatan berbasis kecerdasan buatan (Artificial Intelligence) mulai banyak dikembangkan, khususnya dalam bidang deteksi objek (object detection).[5]

Model jaringan YOLOv5 merupakan salah satu algoritma model jaringan kompleks yang terkategori pada ranah keilmuan deep learning, Dimana ditujukan untuk melakukan pendeteksian dan identifikasi objek di dalam sebuah citra. secara langsung seiring berjalannya waktu[6]. Penelitian ini mengimplementasikan YOLOv5 untuk membangun sistem otomatis yang mampu mendeteksi, mengklasifikasikan, dan menghitung kendaraan, serta mendeteksi pelanggaran melawan arus lalu lintas menggunakan data video.

Kendala utama dalam penelitian ini adalah terbatasnya data nyata pelanggaran lalu lintas akibat keterbatasan akses CCTV dan pertimbangan etika. Oleh karena itu, digunakan video simulasi sebagai data uji. Meskipun berbasis simulasi, sistem tetap dapat dievaluasi secara fungsional untuk menguji efektivitas deteksi pelanggaran.

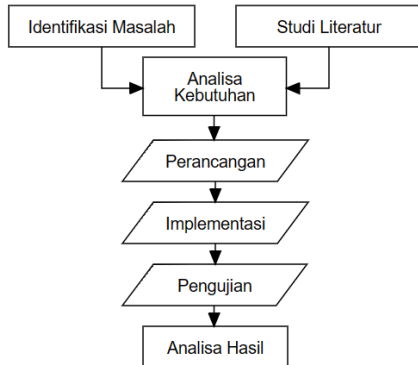
Melalui pendekatan ini, diharapkan sistem dapat menjadikan kontribusi dalam mendukung analisis lalu lintas secara otomatis dan akurat, serta menjadi dasar

pengambilan kebijakan dalam manajemen transportasi dan penegakan hukum lalu lintas.

II. METODE PENELITIAN

A. Metode Penelitian

Sistematika atau alur dari penulisan yang menjadi metode berfikir dalam penelitian otomatisasi perhitungan ini dapat kita lihat gambar 1.



Gambar 1. Diagram alir metode penelitian.

B. Identifikasi Masalah

Penelitian ini membahas permasalahan kemacetan lalu lintas yang dipicu oleh berbagai faktor, seperti minimnya infrastruktur pendukung serta tingginya angka pelanggaran lalu lintas yang berpotensi menimbulkan kecelakaan. Sebagai solusi, penulis mengusulkan pengembangan sistem pemantauan lalu lintas secara otomatis dengan memanfaatkan model dari jaringan YOLOv5. Sistem yang digunakan memiliki rancangan yang bertujuan agar mampu untuk melakukan deteksi dan klasifikasi kendaraan, serta mengidentifikasi pelanggaran melawan arus secara real-time[8].

C. Studi Literatur

Adapun beberapa literatur yang digunakan sebagai referensi adalah literatur yang memenuhi kriteria penulis, yakni :

- Literatur berisi penelitian terhadap pendeteksian objek.
- Literatur berisi penelitian terhadap pendeteksian kendaraan.
- Literatur berisi penelitian terhadap klasifikasi kendaraan.
- Literatur berisi penelitian terhadap pendeteksian pelanggaran.
- Literatur berisi penelitian terhadap kalkulasi pelanggaran
- Literatur berisi penelitian terhadap sistem deteksi kendaraan.
- Literatur menggunakan masukan data berupa gambar dan video.
- Literatur merupakan publikasi ilmiah.

D. Analisa Kebutuhan

Hal-hal yang dibutuhkan sebagai bentuk penunjang kelancaran dalam merancang dan mengimplementasikan serta menguji sistem ialah

menentukan dan memilih standar minimum hardware dan software yang mana saja nanti dipergunakan.

Tabel 1. Spesifikasi perangkat keras.

Perangkat Keras =	RAM 16 GB
	AMD Reyzen 5 75735SHS Processor
	Nvidia GeForce RTX 2050

Tabel 2. Spesifikasi perangkat lunak.

Perangkat Lunak =	Windows 11 - 64 bit
	Python
	Anaconda Navigator
	Anaconda Commandprompt
	Numpy
	Matplotlib
	Sort
	Times
	Adobe Priemere Pro

E. Diagram Sistem



Gambar 2. Blok diagram system deteksi pelanggaran.

a) **Input Image** Merupakan tahap awal di mana sistem menerima data berupa gambar atau video yang akan dianalisis.

b) **Datasets** Proses pencocokan data input dengan data latih (training data) yang telah disiapkan sebelumnya, untuk mendukung proses identifikasi objek.

c) **Object Detection** Tahapan pendeteksian objek pada data input. Jika objek memiliki kesesuaian dengan dataset, maka akan dikenali sebagai objek valid.

d) **Feature Extraction** Proses ekstraksi ciri atau karakteristik dari objek yang terdeteksi, berdasarkan parameter-parameter visual seperti bentuk, warna, atau pola, yang mengacu pada dataset.

e) **Tracking** Tahap pelacakan objek secara berkelanjutan sesuai pergerakan piksel dalam video, untuk memastikan sistem dapat mengikuti objek yang bergerak.

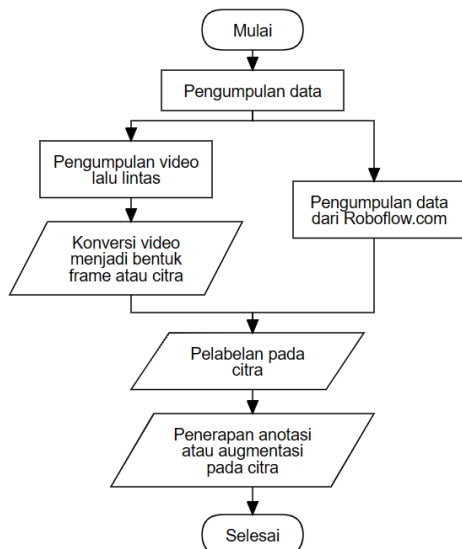
f) **Counting System** Proses perhitungan jumlah objek yang melintasi area atau garis tertentu. Jika objek melewati garis markah yang telah ditentukan, sistem mencatatnya sebagai pelanggaran.

g) **Results** Hasil akhir dari proses pendeteksian dan perhitungan ditampilkan melalui antarmuka sistem, baik secara langsung melalui monitor atau disimpan dalam folder output yang telah ditentukan.

F. Pengumpulan Data

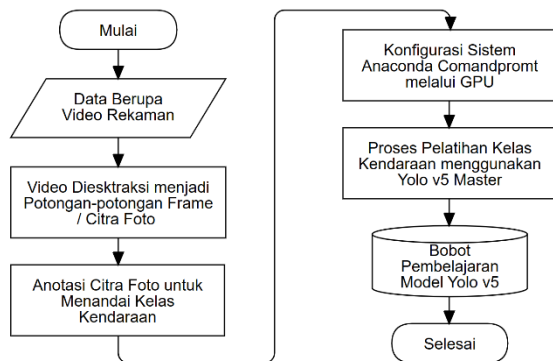
Proses pengumpulan data meliputi pengambilan sample gambar sebagai bahan pembuatan dataset, melalui roboflow, dilanjutkan dengan pelabelan atau anotasi citra menggunakan platform online roboflow.com. Berlanjut pada proses augmentasi, dimana proses ini bertujuan memperbanyak jumlah

referensi oleh sistem untuk pendeteksian objek yang diekpor dalam format yaml.



Gambar 3. Diagram alir pengumpulan data

G. Pelatihan Dataset



Gambar 4. Flowchart pelatihan dataset.

Setelah didapati sebuah data yaml, pelatihan dapat dilakukan secara mandiri dengan memberikan perintah pada terminal python sebagai berikut.

```
python train.py --img 320 --data data.yaml --epochs 100
--weights " --cfg yolov5s.yaml --batch 16 --device 0 --workers 2
```

Gambar 5. Perintah pelatihan data.

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
3/99	1.09G	0.08816	0.06804	0.03305	142	320: 100% [██████████] 99/99 [02:09<00:00, 1.0s/it]
Class						mAP50 mAP50-95: 100% [██████████] 6/6 [00:11<00:00]
all		175	2609	0.411	0.0455	0.06697 0.06198
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
4/99	1.09G	0.08806	0.06557	0.02832	160	320: 100% [██████████] 99/99 [02:02<00:00, 1.0s/it]
Class						mAP50 mAP50-95: 0% [██████████] 0/6 [00:00<00:00]
all		175	2609	0.411	0.0455	0.06697 0.06198
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
5/99	1.09G	0.07410	0.06271	0.02345	98	320: 100% [██████████] 99/99 [02:06<00:00, 1.0s/it]
Class						mAP50 mAP50-95: 100% [██████████] 6/6 [00:00<00:00]
all		175	2609	0.429	0.237	0.235 0.0802
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
6/99	1.09G	0.06934	0.06166	0.02005	378	320: 84% [██████████] 83/99 [01:40<00:21, 1.1s/it]
Class						mAP50 mAP50-95: 100% [██████████] 6/6 [00:00<00:00]
all		175	2609	0.276	0.156	0.661 0.0192

Gambar 6. Proses pelatihan data.

Perintah ini digunakan untuk melatih model YOLOv5 dari awal tanpa memanfaatkan bobot pralatih (pretrained weights). Pelatihan dilakukan dengan ukuran input gambar yang disesuaikan menjadi 320×320 piksel, menggunakan file konfigurasi data.yaml sebagai referensi dataset hasil augmentasi (seperti ditunjukkan pada Gambar 3.9). Proses pelatihan dilakukan selama 100 epoch dengan arsitektur model yolov5s.yaml, serta menggunakan batch size sebesar 16, yang berarti model memproses 16 gambar dalam setiap iterasi pelatihan. Sehingga dihasilkan sebuah bobot dalam model deteksi objek

best.pt	28/10/2023 12:05	PT File	13.967 KB
last.pt	28/10/2023 12:05	PT File	13.967 KB

Gambar 7. Hasil pelatihan data

H. Implementasi

Untuk mengurangi resiko kesalahan perhitungan oleh sistem, peneliti menambahkan sebuah lohika SORT (Simple Online and Realtime tracking) bertujuan untuk meningkatkan performa dari sistem dalam menghitung pelanggaran melawan arus dengan memberikan id unik pada objek, agar tidak terhitung lebih dari satu [7]saat melintas pada lalu lintas yang ada. Pengimplementasian logika sort dapat dilihat pada gambar 3.8.

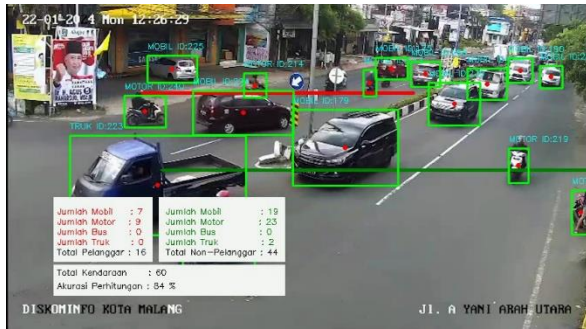
```

for obj in tracked_objects:
    xmin, ymin, xmax, ymax, obj_id = obj
    xmin, ymin, xmax, ymax, obj_id = map(int, (xmin,
    ymin, xmax, ymax, obj_id))
    x_mid = xmin + (xmax - xmin)//2
    y_mid = ymin + (ymax - ymin)//2
    direction = None
    if obj_id in previous_centroids:
        prev_x, prev_y = previous_centroids[obj_id]
        dy = y_mid - prev_y
        if dy > 3:
            direction = "down" # Bergerak ke bawah
        elif dy < -3:
            direction = "up" # Bergerak ke atas
        previous_centroids[obj_id] = (x_mid, y_mid)
        label = None
    for _, det in outputs.iterrows():
        if (det['xmin'] < x_mid < det['xmax']) and
        (det['ymin'] < y_mid < det['ymax']):
            class_id = int(det['class'])
            if class_id < len(classes):
                label = classes[class_id]
            else:
                label = str(class_id)
            break
    if label is None:
        label = 'unknown'
  
```

Gambar 8. Pengimplementasian sort pada algoritma yolov5.

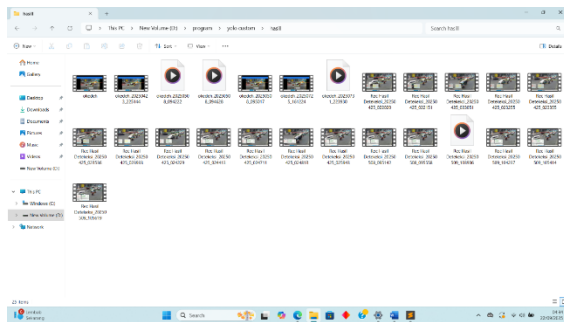
III. HASIL DAN PEMBAHASAN

A. Pengujian



Gambar 9. Citra hasil deteksi.

Citra berhasil di deteksi, klasifikasi dan terhitung. Untuk penerapan sort pada algoritma tersebut juga juga berhasil ditambahkan. Pada gambar 9 terlihat pada setiap objek yang dideteksi memiliki boundingbox, centroid, label dan id unik yang di berikan oleh logika sort. Sistem juga berhasil menyimpan video citra hasil tersebut pada sebuah folder yang ditentukan dalm skrip program keseluruhan seperti pada gambar 10.



Gambar 10. Citra berhasil di simpan dalam folder.

B. Pengujian Performa

Pengujian dilakukan penulis dengan menggunakan video arus lalu lintas dua arah yang di skenariokan bahwa yang salah satu arahnya adalah pelanggar. Hal ini disebabkan keterbatasan peneliti mendapatkan video pengawasan lalu lintas yang terdapat kasus nyata pelanggaran.

Pengujian dilakukan dengan 3 sampel video berbeda dan diuji antar frame. Setiap video diambil sampel acak 5 variabel pada frame yang akan di bandingkan dengan nilai groundtruth yang ada pada video dan bukan menurut sistem. Sehingga dengan begitu, didapati hasil pengujian yang diinginkan.

Tabel 3. Rasio total pada Sample 1.

Mobil : 4	Sepeda Motor : 2	Bus : 0	Truk : 1	
Pelanggar				
Mobil : 1	Sepeda Motor : 3	Bus : 0	Truk : 0	
Total				
Mobil : 27	Sepeda Motor : 41	Bus : 0	Truk : 3	

Tabel 4. Rasio total pada Sample 2.

Frame ke-1361				
Non-Pelanggar				
Mobil : 5	Sepeda Motor : 2	Bus : 0	Truk : 0	
Pelanggar				
Mobil : 1	Sepeda Motor : 2	Bus : 0	Truk : 0	
Total				
Mobil : 18	Sepeda Motor : 28	Bus : 0	Truk : 1	

Tabel 5. Rasio total pada Sample 3.

Frame ke-1505				
Non-Pelanggar				
Mobil : 4	Sepeda Motor : 6	Bus : 0	Truk : 0	
Pelanggar				
Mobil : 3	Sepeda Motor : 3	Bus : 0	Truk : 1 (1m)	
Total				
Mobil : 20	Sepeda Motor : 16	Bus : 0	Truk : 0	

Tabel 3 sampai 5 merupakan hasil akhir dari total pengujian pada 5 frmae berbeda. Berdasarkan paparan pengujian yang dilakukan pada tabel 3. sampai 5. Berdasarkan hasil prediksi jaringan YOLOv5 pada dataset, disusun matriks kebingungan yang menunjukkan performa deteksi untuk empat kelas kendaraan (mobil, motor, bus, truk), sebagaimana tercantum di Tabel 6.

Tabel 6. Confusion matrix hasil deteksi.

Confusion Matrix		Aktual			
		Mobil	Motor	Bus	Truk
Prediksi	Mobil	73	0	0	0
	Sepeda Motor	0	71	0	0
	Bus	0	0	0	0
	Truk	0	0	0	6
	Tidak Terdeteksi	-8	14	0	-2

Merujuk pada tabel 6 barulah dapat ditemukan bobot jumlah kebenaran positif, kesalahan positif, kesalahan negatif, kepresisian, bobot balik pemanggilan, dan skor keakurasian F1 hasil pengolahan jaringan dalam satu kesatuan skenario. Peemaparan bentuk jumlah keseluruhan poin untuk mencari nilai-nilai tersebut.

$$\text{True Positive} = 73 + 71 + 6 = 150$$

$$\text{False Positive} = 1 + 1 + 1 + 1 + 2 + 1 + 2 + 1 = 10$$

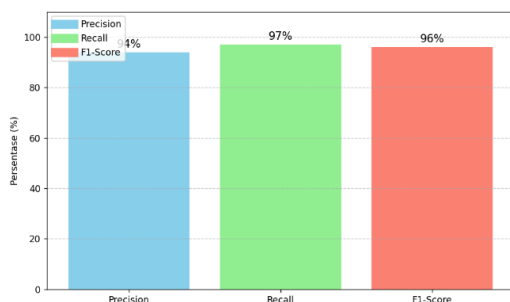
$$\text{False Negative} = (-8) + 14 + (-2) = 4$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{150}{150+9} = 0,94$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{150}{150+4} = 0,97$$

$$\text{F1-Score} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) = 2 \times \left(\frac{0,94 \times 0,97}{0,94 + 0,97} \right) = 0,96$$

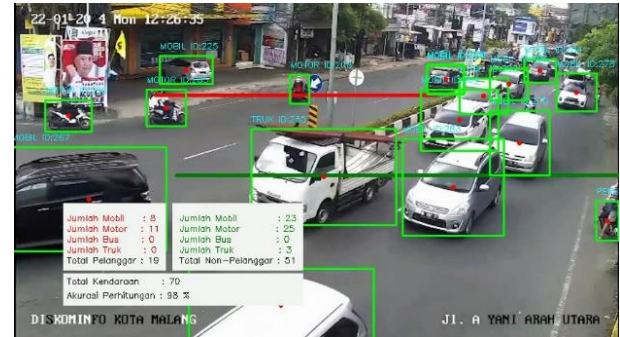
Setelah didapati persamaan tersebut, diketahui terdapat 150 jumlah objek dari kelas yang dikenali oleh sistem jaringan sebagai kebenaran positif, sejumlah 9 objek otomotif yang nilai sebenarnya tidak sesuai dengan pelafalannya sebagai kesalahan positif, dan 4 total pada kesalahan negative, Dimana 4 lolos dari pengenalan sistem. Kepresisian, keakurasian dari skor F1, dan bobot balik pemanggilan masing masing berjumlah berbeda dimulai 0,94 kepresisian, 0,97 bobot balik pemanggilan, sampai 0,97 keakurasian skor F1. Sesuai dengan tujuan dan capaian yang diinginkan paparan tersebut dirasa dapat memenuhi dan dirasa cukup karena bisa dikatakan berbobot nilai tinggi.



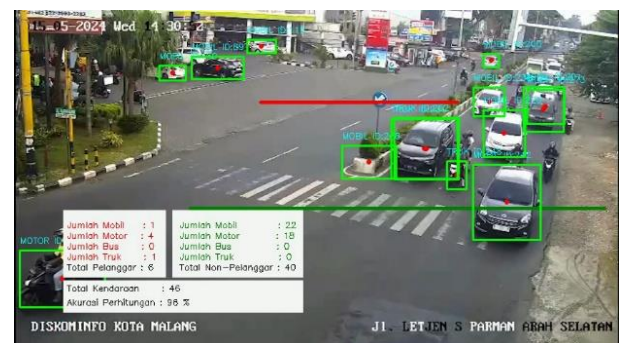
Gambar 11. Statistik hasil pengujian performa.

C. Pengujian Perhitungan Kendaraan

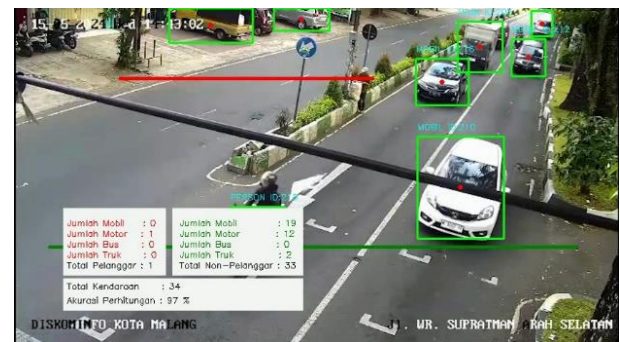
Hasil dari pengujian memaparkan data jumlah kendaraan yang berlalu lalang pada jalan raya terakumulasi menggunakan fitur counter didalam jaringan program sistem berdasar model YOLOv5.



Gambar 12. Hasil perhitungan pada Jl. A Yani.



Gambar 13. Hasil hitung pada Jl. Letjen S Parman.



Gambar 14. Hasil hitung pada Jl. WR Supratman.

Pengujian terhadap program YOLO menunjukkan bahwa sistem mampu melakukan perhitungan jumlah kendaraan secara otomatis. Hal ini ditandai dengan pemanfaatan dua garis bantu berwarna hijau dan merah. Garis hijau digunakan untuk mengidentifikasi kendaraan yang melintas secara normal (non-pelanggar), sedangkan garis merah mendeteksi kendaraan yang melawan arah (pelanggar). Hasil deteksi divisualisasikan pada tampilan antarmuka di pojok kiri bawah, dengan penandaan warna berbeda untuk membedakan pelanggar dan non-pelanggar, sehingga informasi yang ditampilkan mudah dibaca dan dipahami.

Pengambilan data dilangsungkan setelah terlaksanainya rentetan program dengan cara menghitung jumlah kendaraan yang melintas setiap interval 10 detik hingga 1 menit. Hasil dari sistem kemudian dibandingkan dengan hasil perhitungan manual untuk mengukur tingkat akurasi sistem. Uji coba dilakukan menggunakan 3 sampel video rekaman dari CCTV Dinas Perhubungan pada server opensource. Bentuk analisa yang disusun penulis dapat diperhatikan sebagai berikut ;

Tabel 7. Hasil pengujian perhitungan kendaraan.

No.	Lokasi Implementasi	Jumlah Kendaraan		
		Aktual	Terdeteksi	Deteksi Error
1	Jl. A Yani	71	70	4
2	Jl. Letjen S Parman	47	46	5
3	Jl. WR Supratman	36	34	1
Total		154	150	10

Prosesi pengambilan dan penarikan simpulan hasil dari aplikasi program dan pengakumulasian manual guna melihat kesesuaian beratannya setelah menerapkan pengujian masih saja mendapati sejumlah kesalahan yang dianggap menjadi pengurang nilai positif dari sistem. Terlebih lanjut untuk memperjelas perhitungan secara manual mengetahui keakuratan keseluruhan poin yang ada seperti perhitungan berikut.

$$\begin{aligned}
 \text{Keakuratan data} &= 100\% - \left(\frac{\text{Deteksi Error}}{\text{Aktual}} \times 100\% \right) \\
 &= 100\% - \left(\frac{10}{154} \times 100\% \right) \\
 &= 100\% - 6,46\% \\
 &= 93\%
 \end{aligned}$$

IV. KESIMPULAN

Akhir dari proses penelitian menghasilkan penambahan fitur sebelumnya menjadi pembaharuan dalam fiturisasi kompleks terhadap kendaraan secara otomatis menggunakan algoritma YOLOv5 berbasis bahasa pemrograman Python. Sistem dirancang melalui tahapan analisis kebutuhan, pemilihan algoritma, hingga pengujian performa, dengan fokus pada empat kategori objek: sepeda motor, mobil, bus, dan truk. Proses klasifikasi dilakukan berdasarkan pelabelan objek dan mendukung variasi sudut pandang citra.

Simulasi pelanggaran lalu lintas dilakukan dengan mengasumsikan pergerakan kendaraan pada jalur tertentu sebagai pelanggaran arus, memungkinkan sistem diuji meskipun tanpa data nyata pelanggaran. Hasil pengujian menunjukkan bahwa sistem memiliki performa deteksi yang tinggi dengan precision sebesar 94%, recall 97%, dan F1-score 96%. Akurasi sistem dalam menghitung kendaraan mencapai 93%, jauh lebih baik dibandingkan pendekatan manual dari studi sebelumnya yang hanya mencapai akurasi di bawah 50%. Dengan demikian, sistem menunjukkan potensi

yang kuat untuk mendukung pemantauan lalu lintas secara real-time dan efisien

DAFTAR PUSTAKA

- [1] D. Iskandar Mulyana and M. A. Rofik, "Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5," 2022.
- [2] Harahap M et al., "Sistem Cerdas Pemantauan Arus Lalu Lintas dengan YOLO (You Only Look Once v3)". 2020.
- [3] A. R. K. Amiril Wiuldan S, "Deteksi Pelanggaran Lalu Lintas Tidak menggunakan Helm dengan YOLOv4 pada Sistem ETLE" Jurnal Teknik Kompt. AMIKBSI, Vol.8, 2022.
- [4] F. Rofii, Mashudi A dan Mukhsim M, "Sistem Kamera Cerdas untuk Deteksi Pelanggaran Marka Jalan" Science Elektro. Vol.01., 2020.
- [5] Hutaeruk J. S. W., et al "Deteksi Kendaraan Secara Realtime Menggunakan Metode YOLO Berbasis Android." Vol.09., 2020.
- [6] M. R. Ardiansyah, Y. Supit, M. S. Said, S. Komputer, S. Tinggi, and M. Informatika, "Menggunakan Algoritma Yolo," vol. 7, no. 1, 2022.
- [7] A. Salim, "ESTIMASI KECEPATAN KENDARAAN MELALUI VIDEO PENGAWAS LALU LINTAS MENGGUNAKAN PARALLEL LINE MODEL" vol.-, no.-, 2020.
- [8] Alfiyandi Abdurrafi D, "Deteksi Klasifikasi dan Menghitung Kendaraan Berbasis Algoritma You only Look Once Menggunakan Kamera Cctv" Prodi Tek. Elektro, Fak. Teknik, Unisma. 2023.
- [9] Martinus Ade M dan Wiyli Y, "Implementasi Metode You Only Look Once (YOLOv5) dalam Deteksi Pelanggaran Helm", Pro. Studi Sistem Informasi, Fak. Teknik, Unesa, 2024.
- [10] Reza Ifmanur I, "Aplikasi Penghitung Kendaraan yang Melintas di Jalan Raya Berdasarkan Metode Yolo Object Detection" Prog. Studi Tek. Komputer, Fak. Teknik dan Informatika, 2020.