

Aplikasi Penghitung Kecepatan Mobil dengan Akurasi Tinggi Menggunakan Yolo untuk Meminimasi Kecelakaan

Application Highly Accurate Car Speed Monitor using Yolo to Minimize Accidents

**Henny¹, Muhammad Azhar Baiquni², Budi Mulyanti², Muhammad Fadli Nasution², Agus
Heri Setya Budi^{2*}**

¹Program Studi Teknik Industri, Fakultas Teknik dan Ilmu Komputer

Universitas Komputer Indonesia Jl. Dipatiukur 112-116 Bandung

²Program Studi Teknik Elektro, Fakultas Pendidikan Teknologi dan Kejuruan

Universitas Pendidikan Indonesia Jl. Dr. Setiabudi No.229

Email*: agusheri@upi.edu

Abstrak - Salah satu penyebab terjadinya kecelakaan adalah kurangnya kewaspadaan pengemudi dan pelanggaran laju kendaraan melampaui batas maksimal. Salah satu cara untuk mengurangi tindak pelanggaran tersebut diperlukan pengawasan lalu lintas pada area jalan terutama di area yang rawan terjadi kecelakaan. Tujuan dari penelitian ini yaitu membangun sistem deteksi laju dan plat nomor kendaraan berbasis video rekaman menggunakan YOLOv5-DeepSORT dan HyperLPR sebagai metode pengawasan lalu lintas di area rawan kecelakaan. Sistem menggunakan YOLOv5 dan DeepSORT untuk mendeteksi dan melacak pergerakan kendaraan sehingga diperoleh perpindahan jarak kendaraan yang digunakan sebagai acuan deteksi laju kendaraan. Adapun HyperLPR digunakan untuk mendeteksi plat nomor dari kendaraan tersebut. Metode yang digunakan dalam penelitian ini yaitu metode eksperimen dengan melakukan perekaman video pada ruas jalan tol Cipali yang digunakan sebagai *masukan* dari program deteksi laju dan plat nomor kendaraan. Hasil pengujian deteksi objek kendaraan menggunakan YOLOv5 diperoleh nilai evaluasi metric *Precision* sebesar 100%. Pengujian deteksi laju kendaraan diperoleh nilai rata-rata persentase *error* sebesar 7,6% terhadap nilai sebenarnya. Adapun dari deteksi plat nomor kendaraan diperoleh hasil akurasi karakter secara keseluruhan sebesar 91,82%. Secara keseluruhan, sistem dapat menjalankan tiga proses yang menjadi fungsi utama yaitu deteksi dan tracking kendaraan, deteksi laju kendaraan dan deteksi plat nomor kendaraan. Pengembangan penelitian kedepan diharapkan mampu mengurangi nilai *error* pada penelitian ini dan dapat dipertimbangkan untuk dipergunakan secara komersial untuk mencegah kecelakaan yang diakibatkan oleh laju kendaraan yang melebihi batas.

Kata kunci: Laju kendaraan, plat nomor kendaraan, YOLOv5, DeepSORT HyperLPR.

Abstract - One of the causes of accidents is the lack of vigilance by drivers and violations of vehicle speed limits. One way to reduce such violations is the implementation of traffic surveillance in accident-prone areas. The aim of this research is to develop a video-based vehicle speed and license plate detection system using YOLOv5-DeepSORT and HyperLPR as methods for traffic monitoring in accident-prone areas. The system utilizes YOLOv5 and DeepSORT to detect and track vehicle movements, obtaining the displacement distance of vehicles as a reference for speed detection. HyperLPR is employed to recognize license plates of the detected vehicles. The research methodology involves experimental techniques, including video recording on the Cipali toll road section, as input for the vehicle speed and license plate detection program. The evaluation of object detection using YOLOv5 yielded a *Precision* metric of 100%. In the testing of vehicle speed detection, an average error percentage of 7.6% was obtained compared to actual values. The license plate detection results showed an overall character accuracy of 91.82%. Overall, the system successfully performs its three main functions: vehicle detection and tracking, vehicle speed detection, and license plate detection. Future research development is expected to further reduce error values and may be considered for commercial use to prevent accidents caused by speeding vehicles.

Keywords: Vehicle speed, vehicle license plate, YOLOv5, DeepSORT, HyperLPR.

I. PENDAHULUAN

Meningkatnya kendaraan dan padatnya lalu lintas pada suatu daerah menyebabkan rawan terjadinya kecelakaan. Salah satu penyebab utama kematian di seluruh dunia yaitu kecelakaan [1]. Berdasarkan data yang diberikan oleh PT Lintas Marga Sedaya, operator jalan tol Cikopo-Palimanan, tercatat ada 1.075 kejadian kecelakaan pada tahun 2019. Ruas jalan tol Cikopo-Palimanan memiliki kondisi geografis yang dominan lurus, sehingga banyak pengguna jalan yang melampaui batas laju maksimal yang ditetapkan untuk jalan tol tersebut [2]. Jalan tol yang digunakan untuk lalu lintas antar kota didesain berdasarkan laju rencana paling rendah 80 km/jam, dan jalan tol di wilayah perkotaan didesain dengan laju rencana paling rendah 60 km/jam [3]. Melalui rekaman video teknologi segmentasi citra dapat digunakan untuk mengawasi kawasan lalu lintas padat disuatu daerah. Segmentasi citra dapat digunakan untuk melakukan proses objek deteksi, diantaranya adalah deteksi plat nomor, jenis kendaraan, dan laju kendaraan [4] [5].

DeepSORT adalah algoritma *tracking* visi komputer untuk melacak objek yang telah ditentukan dan menetapkan ID ke setiap objek. *DeepSORT* dapat didefinisikan sebagai algoritma *tracking* untuk melacak objek tidak hanya berdasarkan laju, gerakan dan tampilan dari objek tersebut [6][7]. *DeepSORT* menggunakan pendekatan *tracking* objek dimana pendekatan dasar seperti filter Kalman dan algoritma Hungaria digunakan untuk melacak objek [8].

License Plate Recognition (LPR) merupakan sistem yang digunakan untuk mengidentifikasi kendaraan berdasarkan pengenalan plat nomor kendaraan [9]. Metode umum untuk pengenalan karakter pelat termasuk *Convolutional Neural Network* (CNN). Algoritma CNN memiliki kemampuan untuk mengekstrak fitur gambar secara otomatis, dan secara umum tampil lebih baik daripada metode lain dalam hal klasifikasi gambar [10]. *HyperLPR* menerapkan beberapa metode CNN yang masing-masing dipisahkan berdasarkan detektor plat nomor, pengumpulan karakter dan pengklasifikasian [11][12].

Berdasarkan penelitian terdahulu tersebut, peneliti berinisiatif untuk menggabungkan deteksi laju kendaraan dan plat nomor kendaraan dalam satu sistem dengan menggunakan algoritma *tracking DeepSORT* untuk melakukan proses *tracking* kendaraan dan library *python HyperLPR* untuk mendeteksi dan mengenali karakter pada plat nomor kendaraan. Pada penelitian ini sistem segmentasi dilakukan menggunakan video

rekaman sebagai *masukan* segmentasi. Pada proses melakukan *tracking* digunakan *DeepSORT* untuk melacak lokasi kendaraan dan menambahkan ID pada setiap kendaraan yang terdeteksi. Selanjutnya akan diterapkan perhitungan perpindahan jarak berdasarkan piksel kendaraan yang terdeteksi untuk proses estimasi laju kendaraan melalui informasi *tracking DeepSORT* berdasarkan lokasi sebelum dan sesudah kendaraan melintas. Selain itu, untuk mendeteksi plat nomor dari setiap kendaraan yang tertangkap kamera menggunakan *HyperLPR*.

Pada penelitian yang dilakukan oleh Zulfikri dkk dilakukan sebuah perancangan sistem penegakan *speed bump* berdasarkan kecepatan kendaraan yang diklasifikasikan *Haarcascade*. Penelitian tersebut mendapatkan akurasi deteksi sebesar 97,92% dan nilai kesalahan rata-rata dalam perbandingan antara hasil deteksi dengan kecepatan sebenarnya sebesar 2,88 [13]. Adapun penelitian yang dilakukan oleh Tsani dkk yaitu perancangan sistem deteksi kecepatan kendaraan menggunakan metode *Frame Difference* dimana dalam hal tersebut sistem akan membandingkan objek per-frame untuk mendeteksi adanya pergerakan pada objek kendaraan. Penelitian tersebut menggunakan perhitungan *Mean Squared Error* (MSE) untuk mendapatkan nilai kesalahan rata-rata yaitu sebesar 12,632 [14]. Sedangkan metode yang digunakan untuk mendeteksi plat nomor kendaraan diantaranya menggunakan *connected components* dan Support Vector Machine (SVM) [7], *Learning Vector Quantization* [8], *Optical Character Recognition* (OCR) [9][15]. Namun dalam penelitian sebelumnya hanya melakukan proses deteksi laju maupun plat nomor kendaraan secara terpisah. Berdasarkan penelitian terdahulu tersebut ini peneliti berinisiatif untuk menggabungkan deteksi laju kendaraan dan plat nomor kendaraan dalam satu sistem dengan menggunakan algoritma *tracking DeepSORT* untuk melakukan proses *tracking* kendaraan dan library *python HyperLPR* untuk mendeteksi dan mengenali karakter pada plat nomor kendaraan. Tujuan Penelitian yang dilakukan adalah untuk mendesain dan melakukan perancangan sistem deteksi laju dan plat nomor kendaraan berbasis video menggunakan *YOLOv5*, *DeepSORT* dan *HyperLPR*.

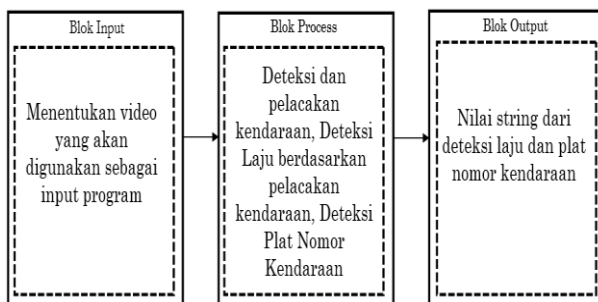
II. METODOLOGI

Dalam penelitian ini digunakan metode eksperimen berbasis pemrosesan video dengan menerapkan serangkaian langkah pemrograman dengan menerapkan *YOLOv5*, *DeepSORT* dan

HyperLPR. Metode ini dipilih karena dapat diterapkan untuk melakukan deteksi pelanggaran batas laju dan plat nomor kendaraan dengan memanfaatkan kamera pengawas area jalan Tol. Proses sistem secara umum ditunjukkan pada **Gambar 1** masukan yang digunakan adalah sebuah rekaman yang diambil pada area ruas jalan tol Cipali. Masukan video dengan melakukan pengambilan video menggunakan kamera yang diposisikan sedemikian rupa sehingga kamera dapat menjangkau setiap kendaraan yang melintas dan dapat menjangkau bagian area plat nomor kendaraan dengan jelas. Bahasa pemrograman yang digunakan dalam penelitian ini adalah bahasa *Python*.

A. Diagram Blok

Pada penelitian ini, telah dirancang sebuah sistem yang menggunakan metode *YOLOv5*, *DeepSORT* dan *HyperLPR* dalam melakukan keseluruhan sistem deteksi. Ilustrasi dalam **Gambar 1** memperlihatkan diagram blok dari sistem pendeteksian plat nomor pada kendaraan. Sistem ini terdiri dari tiga blok utama. Blok pertama berfungsi sebagai blok masukan yang memanfaatkan kamera sebagai sensor untuk menangkap gambar dari setiap kendaraan yang melintas dan dapat menjangkau area plat nomor kendaraan yang akan dianalisis. Blok kedua merupakan blok Proses yang menggunakan *Personal Computer* (PC) sebagai perangkat pengolah data utama. Sebagian besar proses sistem dijalankan pada blok kedua ini. PC bertanggung jawab atas proses inti, yakni pengolahan citra yang diimplementasikan melalui perangkat lunak *Pycharm*. Blok terakhir, yang merupakan blok keluaran, juga menggunakan PC sebagai antarmuka penghubung antara pengguna dengan sistem. Hasil keluaran dari sistem ini berupa rangkaian karakter yang diekstraksi dari gambar yang diambil oleh kamera pada tahap masukan



Gambar 27. Diagram blok sistem

B. Bahan dan peralatan sistem

Dalam penelitian ini terdiri dari perangkat keras dan perangkat lunak. Perangkat keras yang digunakan dalam penelitian yaitu sebuah laptop dengan spesifikasi *processor* Ryzen 4600H, NVIDIA GTX 1650, RAM 8GB, sedangkan perangkat lunak yang digunakan adalah *YOLO*, *DeepSORT*, *HyperLPR*, *Pytorch*, *Pycharm*.

C. Pengumpulan Data

Data yang dimaksud dalam penelitian ini merupakan sebuah video yang dijadikan sebagai *masukan* sistem. Data video terdiri dari rekaman video kondisi jalan pada area ruas jalan tol dengan cakupan jarak area mobil melintas hingga mobil meninggalkan area perekaman video adalah 80 meter. Area lokasi yang digunakan sebagai *masukan* dalam video pemrograman diperlihatkan pada **Gambar 2**.



Gambar 28. Area lokasi tempat pengambilan video

D. Metode Pendeteksian dan Pelacakan Objek

Dalam tahapan ini terdiri dari dua proses yaitu deteksi kendaraan menggunakan *YOLOv5* dan proses pelacakan menggunakan *DeepSORT*. *YOLOv5* melakukan pemrosesan memecah gambar menjadi bagian *grid* kecil, dan setiap *grid* bertanggung jawab untuk mendeteksi objek di dalamnya. *YOLOv5* menggunakan jaringan saraf tiruan untuk mengklasifikasikan dan menghitung kotak pembatas (*bounding box*) untuk objek kendaraan yang terdeteksi. Hasil deteksi ini kemudian melewati proses *Non-Maximum Suppression* (NMS) untuk menghilangkan deteksi yang berlebihan atau tumpang tindih. *YOLOv5* bertanggung jawab atas deteksi awal objek dan menyaring informasi yang relevan. *DeepSORT* memungkinkan pelacakan yang lebih canggih dan akurat terhadap objek yang bergerak di dalam video. *DeepSORT* menggunakan konsep ReID (*Re-identification*) untuk mengidentifikasi dan melacak objek dari satu bingkai ke bingkai berikutnya.

ReID adalah teknik yang membandingkan fitur visual dari objek untuk mengenali objek yang sama di berbagai bingkai. Keluaran dari tahap ini adalah sistem mampu mendeteksi dan menandai setiap kendaraan yang terdeteksi- dengan ID *tracking*. Dalam *DeepSORT* diperlukan pengaturan konfigurasi seperti *max_dist*, *min_confidence*, *max_iou_distance* Indeks IoU *max_age*. Keluaran dari tahap ini adalah sistem mampu mendeteksi dan menandai setiap kendaraan yang terdeteksi- dengan ID *tracking*.

E. Metode Pendeteksian Laju

Laju kendaraan dapat ditentukan berdasarkan deteksi dan pelacakan pada proses sebelumnya. Prinsip utama dalam estimasi laju kendaraan secara real-time adalah dengan merekam perpindahan posisi piksel pada *tracking* ID kendaraan yang dilakukan pada tahap sebelumnya berdasarkan rumus perhitungan jarak pada persamaan (1).

$$D = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \dots (1)$$

Pada persamaan (1) nilai *D* adalah jarak dengan X_1 menunjukkan posisi horizontal kendaraan keadaan awal, X_2 menunjukkan posisi horizontal keadaan akhir. Perekam perpindahan piksel dilakukan pada kondisi sebelum dan sesudahnya dilakukan berdasarkan setiap per 10 *frame*. Konversi perhitungan antara jarak sebenarnya dengan jarak pada piksel diperlihatkan pada persamaan (2).

$$Jarak\ real = \frac{D \times Width \times frame \times 3,6 \times 2}{fps} \dots (2)$$

Pada persamaan konversi antara jarak piksel ke jarak sebenarnya seperti yang ditampilkan pada persamaan (2), dimana *D* merupakan jarak piksel, *Width* merupakan lebar kendaraan, *Frame* merupakan jumlah persetiap *frame* yang digunakan untuk menentukan deteksi laju, konstanta 3,6 merupakan nilai konversi dari km/s menjadi m/s, dan *fps* merupakan *frame per second* yang didapatkan pada video masukan.

F. Metode Deteksi Plat Nomor Kendaraan

Deteksi plat nomor kendaraan pada dasarnya bekerja dengan dua tahapan, yaitu tahap mendeteksi objek plat nomor kendaraan dan tahap mendeteksi karakter dari hasil deteksi plat nomor kendaraan tersebut. Pada *HyperLPR* metode deteksi objek plat nomor kendaraan menggunakan *cascade classifier* sehingga dalam mendeteksi objek relatif cepat namun dalam hal akurasi tidak terlalu akurat. Sedangkan dalam mendeteksi karakter pada plat nomor kendaraan *HyperLPR* melakukan proses *training* model dataset

pengenalan karakter dengan menerapkan CTC *Loss* yang memungkinkan model untuk mempelajari pemetaan sekuensial tanpa pasangan masukan-keluaran yang sekuensial. Ini dilakukan dengan menghitung probabilitas jalur pemetaan menggunakan *forward-backward algorithm* dan menggunakan *backpropagation* untuk mengoreksi model. *HyperLPR* akan mengganti karakter yang ditampilkan pada plat nomor pada ID *tracking* kendaraan akan mengalami pembaruan tergantung pada nilai *confidence*, jika nilai *confidence* plat nomor terdeteksi lebih besar dari *confidence* sebelumnya maka hasil yang ditampilkan akan diganti, hal ini bertujuan untuk mendapatkan hasil deteksi mendekati karakter plat nomor sebenarnya.

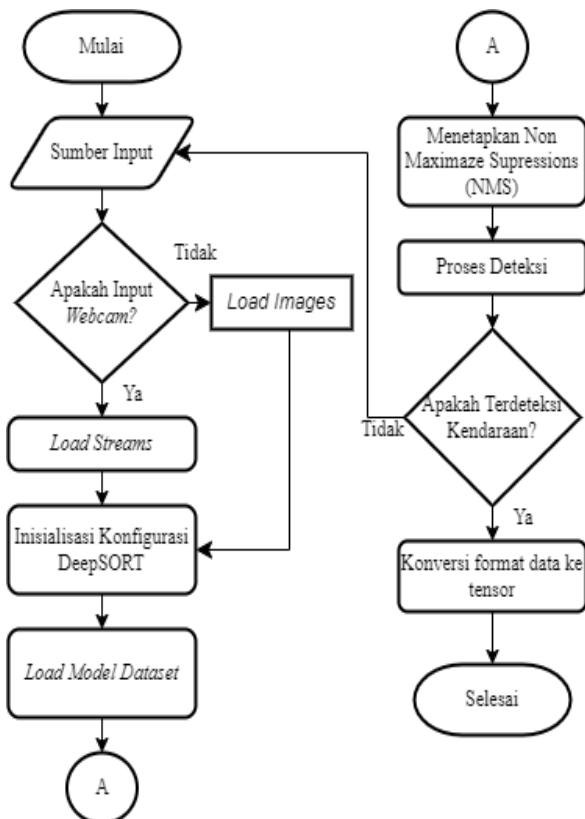
III. HASIL DAN PEMBAHASAN

A. Deteksi Kendaraan

Perancangan deteksi dan *tracking* kendaraan terdapat beberapa proses seperti yang diperlihatkan pada diagram alur pada **Gambar 3** objek deteksi dan *tracking* objek merupakan tahapan awal pada proses deteksi laju kendaraan pada sebuah masukan video. Keluaran dari objek deteksi ini adalah menandai setiap kendaraan yang melintas dan menerapkan *Bounding Box* dan menetapkan ID pada setiap kendaraan yang terlihat pada kamera. Keberhasilan sistem dalam mendeteksi dan melacak objek yang terdeteksi dalam video ditandai dengan *Bounding Box* yang terdapat pada kendaraan secara terus menerus pada setiap perubahan frame dan ditandai dengan tidak berubahnya ID pada masing-masing kendaraan sampai objek kendaraan tersebut menghilang dari frame pada video.

Berdasarkan diagram alur pada **Gambar 3** proses awal yang dilakukan adalah menentukan sumber masukan *load stream* ditentukan jika sumber masukan adalah *streaming* melalui kamera secara langsung sedangkan *load images* digunakan jika sumber masukan adalah sebuah video rekaman. Selanjutnya adalah inisialisasi konfigurasi *DeepSORT* diantaranya adalah *max_dist* yang digunakan untuk menentukan jarak maksimum untuk membedakan objek. *Min_confidence* digunakan untuk menentukan nilai batas minimum kevalidan objek deteksi, *max_iou_distance* digunakan untuk menentukan jarak maksimum IOU untuk menghindari dua prediksi *tracking* yang sama, *max_age* berfungsi untuk menentukan rentang deteksi frame dipertahankan, *n_init* digunakan untuk menentukan batas *frame* yang harus terdeteksi

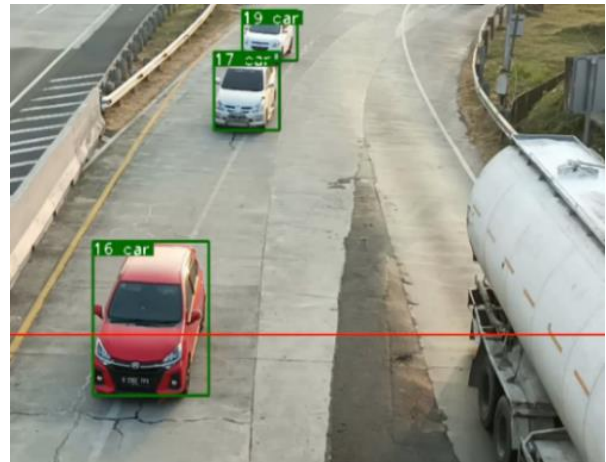
untuk membuat *tracking* baru. Selanjutnya hasil deteksi objek yang telah diolah dan diadaptasi ke format *DeepSORT* akan diserahkan ke dalam algoritma *DeepSORT* untuk dilakukan *tracking* objek. Pada bagian ini *DeepSORT* memberikan informasi berupa lokasi dan ID setiap objek yang sedang dilacak pada *frame* tersebut. Langkah selanjutnya adalah proses deteksi. Proses deteksi dilakukan dengan membuat fungsi *loop* untuk mendeteksi objek dalam sebuah gambar dan memproses hasil deteksi pada setiap *frame*.



Gambar 29. Diagram alur deteksi dan pelacakan kendaraan



Gambar 4 memperlihatkan sampel dari hasil deteksi dan pelacakan kendaraan.



Gambar 30 Contoh hasil deteksi kendaraan

Berdasarkan Gambar 4 terlihat bahwa pada objek kendaraan terdapat sebuah *bounding box* yang mengelilingi kendaraan tersebut serta terdapat *ID* berupa angka pada masing-masing kendaraan yang telah terdeteksi tersebut. Pengujian deteksi kendaraan dan *tracking* kendaraan dilakukan dengan menggunakan masukan video perekaman pada area ruas jalan tol Cipali dengan durasi video selama 2 menit dengan menerapkan fps 30 dan melakukan *export* 1 dari setiap 10 *frame*. Hasil dari pengujian menunjukkan bahwa didapatkan sebanyak 361 *frame* dengan 48 model mobil yang berbeda satu sama lain. Pada deteksi tersebut, terdapat nilai *DR* (*detection rate*) yang mempresentasikan rasio antara kendaraan yang terdeteksi dengan objek sebenarnya. Kendaraan dianggap terdeteksi jika setidaknya lokalisasi 30% dari seluruh waktu dalam video. Dalam menghitung akurasi dalam deteksi objek kendaraan digunakan perhitungan evaluasi metric *Precision*. *Precision* digunakan untuk menentukan nilai sejauh mana model mampu mengidentifikasi mobil dengan benar dari semua deteksi yang dilakukan. Adapun untuk menentukan tingkat akurasi model dalam mengidentifikasi mobil dengan benar dapat dilihat pada persamaan (3).

$$Precision = \frac{TP}{(TP+FP)} \quad \dots(3)$$

Dimana *TP* adalah Jumlah mobil yang terdeteksi, *FP* adalah Jumlah mobil yang salah terdeteksi. Sehingga nilai *Precision* dan *Recall* dapat ditentukan yaitu

$$Precision = \frac{48}{48+0} = 1,0 = 100\%$$

Berdasarkan hasil tersebut sistem dapat mendeteksi objek kendaraan dengan akurat, dengan memiliki nilai *Precision* sebesar 100%. Hasil tersebut didapatkan karena masukan yang digunakan pada pengujian berupa video yang memiliki kualitas yang baik serta fps yang memadai sehingga seluruh

objek yang dideteksi oleh algoritma dapat terbaca dengan jelas. Selain itu, hasil dari precision yang didapatkan menyerupai dengan beberapa penelitian yang menggunakan DeepSORT untuk menghitung kendaraan menggunakan YOLOv3 dengan *Precision* diatas 90% dan menggunakan YOLOv4 dengan akurasi lebih rendah yaitu [16] [7].

B. Hasil Pengujian Deteksi Laju Kendaraan

Pengujian dilakukan dengan melakukan pengambilan video di area ruas jalan tol Cipali pada mobil yang lajunya telah ditentukan dan selama melakukan perekaman dilakukan perekaman pada *speedometer* dari dalam mobil untuk dapat membandingkan antara hasil deteksi laju kendaraan dengan laju kendaraan sebenarnya. Hasil dari sistem estimasi laju kemudian akan dibandingkan dengan laju sebenarnya. Sistem deteksi laju kendaraan dilakukan berdasarkan banyaknya *frame* dalam video. laju dilakukan dengan menghitung berdsarkan perpindahan posisi piksel dari setiap kendaraan yang telah ditetapkan. laju kendaraan dengan laju kendaraan sebenarnya. Hasil dari sistem estimasi laju kemudian akan dibandingkan dengan laju sebenarnya. Sistem deteksi laju kendaraan dilakukan berdasarkan banyaknya *frame* dalam video. laju dilakukan dengan menghitung berdsarkan perpindahan posisi

piksel dari setiap kendaraan yang telah ditetapkan ID yang telah dilakukan pada proses *tracking* sebelumnya Perhitungan perpindahan posisi piksel ID dilakukan setiap per 10 *frame* sehingga hasil deteksi pada setiap kendaraan tertentu akan terus diperbaharui. hingga kendaraan tersebut meninggalkan area deteksi atau ketika durasi video selesai. Percobaan penelitian dilakukan sebanyak 5 kali dengan menggunakan video yang berbeda. Dalam masing-masing video terdapat perubahan kecepatan sebanyak 4 kali. Sampel perbandingan antara hasil deteksi laju kendaraan dengan laju sebenarnya dapat dilihat pada **Tabel I**.

Dalam mengukur keakuratan deteksi laju kendaraan dilakukan dengan menggunakan *Mean Absolute Percentage Error* (MAPE). Perhitungan dengan menggunakan MAPE ditunjukkan pada persamaan (4)

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - F_i}{A_i} \right| \times 100 \% \dots (4)$$

Dimana dengan menggunakan persamaan tersebut A_i merupakan nilai laju kendaraan sebenarnya dan F_i merupakan nilai hasil deteksi laju kendaraan. Berdasarkan pengujian sistem deteksi laju kendaraan dengan menggunakan 5 masukan video rekaman yang berbeda didapatkan 20 laju yang terdeteksi dari keseluruhan masukan tersebut. Hasil dari masing-masing deteksi laju tersebut ditunjukkan pada **Tabel II** sampai dengan **Tabel VI**.

Tabel I. Perbandingan deteksi laju dengan laju sebenarnya

No	Deteksi Laju	Laju sebenarnya	No	Deteksi Laju	Laju Sebenarnya
1			3		
2			4		

Tabel I. Selisih kesalahan deteksi laju pada video 1

Video	Hasil Deteksi (km/jam)	Laju Sebenarnya (km/jam)	Selisih kesalahan (%)
<u>1</u>	28,8	30	4,17
1	31,7	32	0,95
1	37,1	37	0,27
1	42,8	41	4,20

Tabel II. Selisih kesalahan deteksi laju pada video 2

Video	Hasil Deteksi (km/jam)	Laju Sebenarnya (km/jam)	Selisih kesalahan (%)
<u>1</u>	28,8	30	4,17
1	31,7	32	0,95
1	37,1	37	0,27
1	42,8	41	4,20

Tabel III. Selisih kesalahan deteksi laju pada video 3

Video	Hasil Deteksi (km/jam)	Laju Sebenarnya (km/jam)	Selisih kesalahan (%)
<u>1</u>	28,8	30	4,17
1	31,7	32	0,95
1	37,1	37	0,27
1	42,8	41	4,20

Tabel V. Selisih kesalahan deteksi laju pada video 4

Video	Hasil Deteksi (km/jam)	Laju Sebenarnya (km/jam)	Selisih kesalahan (%)
<u>1</u>	28,8	30	4,17
1	31,7	32	0,95
1	37,1	37	0,27
1	42,8	41	4,20

Tabel VI. Selisih kesalahan deteksi laju pada video 5

Video	Hasil Deteksi (km/jam)	Laju Sebenarnya (km/jam)	Selisih kesalahan (%)
<u>1</u>	28,8	30	4,17
1	31,7	32	0,95
1	37,1	37	0,27
1	42,8	41	4,20

Berdasarkan data pada tabel, nilai MAPE dapat ditentukan dengan membagi jumlah selisih kesalahan antara hasil deteksi dan laju sebenarnya dengan total kecepatan yang muncul dalam keseluruhan video.

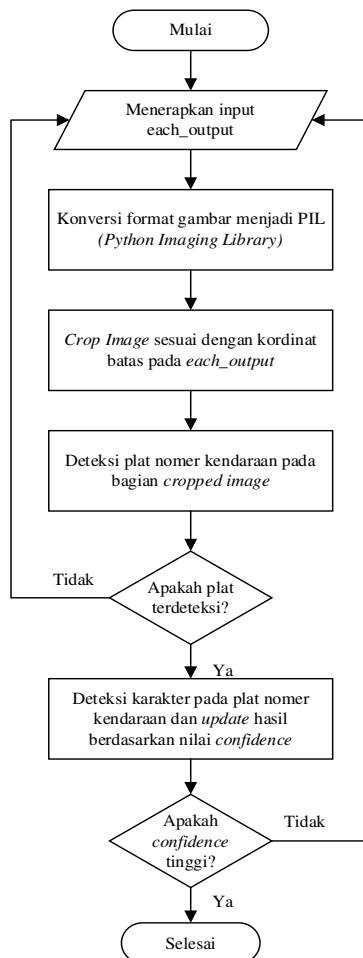
$$\text{MAPE} = \frac{145,7784}{20} \times 100 \% = 7,2 \%$$

Nilai MAPE pada sistem didapatkan sebesar 7,2%. Semakin kecil persentase MAPE, maka performa dari sistem tersebut akan semakin baik. Hasil *error* tersebut juga menunjukkan bahwa sistem yang dibangun menyerupai dengan metode lainnya seperti menggunakan BLOB (*Binary Large Object*) dengan rata-rata *error* 6,1% yang dilakukan oleh Gembong dkk [17]. Perbedaan hasil akurasi ini disebabkan oleh beberapa faktor diantaranya yaitu perbedaan jenis algoritma yang digunakan dan kualitas masukan video. Pada penelitian Gembong dkk menggunakan algoritma BLOB dan data masukan video yang diambil berada di ketinggian 5 meter dari jalan sedangkan pada penelitian ini algoritma yang digunakan yaitu DeepSORT dengan pengambilan data video berada di ketinggian 80 meter.

C. Hasil Pengujian Deteksi Plat Nomor Kendaraan.

Terdapat beberapa proses dalam mendeteksi plat nomor kendaraan. Diagram alur perancangan deteksi plat nomor kendaraan ditunjukkan pada Gambar 5. Langkah awal dalam deteksi plat nomor kendaraan adalah mengkonversi gambar dari format OpenCV menjadi citra PIL (*Python Imaging Library*). Hal ini bertujuan untuk dapat memanipulasi gambar dengan menggunakan fitur pada *library* PIL. Bagian selanjutnya yaitu pemotongan gambar yang sesuai dengan *B* yang ditemukan pada *each_keluaran*. Hal ini bertujuan untuk memfokuskan pada area plat nomor kendaraan yang terdeteksi.

Selanjutnya adalah deteksi plat nomor dan karakter pada plat nomor kendaraan pada bagian gambar yang telah dipotong. Deteksi tersebut dilakukan dengan menggunakan *library HyperLPR*. Hasil deteksi karakter pada plat nomor kendaraan tersebut akan diperbaharui berdasarkan nilai *confidence* dari hasil tersebut.



Gambar 31 Diagram alur deteksi plat nomor kendaraan

Dalam pengujian deteksi plat nomor kendaraan pengujian dilakukan dengan menjalankan program dengan masukan beberapa video area ruas jalan tol Cipali dan video yang diambil dari yang diambil dari Youtube. Pengujian dilakukan pada 5 mobil yang terdeteksi secara acak berdasarkan masukan video yang diambil dari rekaman ruas jalan tol Cipali dan 5 mobil yang terdeteksi berdasarkan masukan video yang diambil dari Youtube. Sampel hasil deteksi plat nomor kendaraan dapat dilihat pada **Gambar 6.** dan **Gambar 7.**

Berdasarkan pengujian sistem deteksi plat nomor kendaraan pada 10 plat nomor kendaraan yang terdeteksi dari beberapa video masukan yang berbeda dari total 10 mobil pengujian pada Tabel 8 didapatkan hasil dengan total 71 karakter plat nomor, 6 mobil mendeteksi karakter dengan benar secara keseluruhan, 1 mobil mendeteksi plat nomor secara keseluruhan namun terdapat tambahan 1 deteksi karakter. 2 mobil mendeteksi dengan perbedaan 1 karakter, dan 1 mobil mendeteksi dengan perbedaan 2 karakter. Perbedaan karakter pada sistem disebabkan oleh beberapa faktor seperti karakter huruf dan angka yang menyerupai. Evaluasi keakuratan deteksi plat nomor kendaraan

dengan menggunakan metrik akurasi karakter (Ak). Perhitungan akurasi karakter dilakukan dengan membagi jumlah total karakter yang terdeteksi dengan jumlah total karakter sebenarnya, kemudian hasilnya dikalikan dengan 100%.



Gambar 32 Hasil deteksi plat nomor kendaraan



Gambar 33 Hasil deteksi plat nomor kendaraan

Dari hasil pengujian perbandingan antara hasil deteksi plat nomor kendaraan dan plat nomor sebenarnya yang telah dilakukan, didapatkan hasil perbandingan dan nilai akurasi yang didapatkan ditunjukkan pada **Tabel VII.**

Tabel III. Perbandingan dan akurasi deteksi plat nomor dengan plat nomor sebenarnya

No	Plat sebenarnya	Hasil Deteksi	Akurasi (%)
1.	E 1518 LT	E1518LT	100
2.	E1689DE	E1G89DE	85,71
3.	B 1912 UYT	B1912UMT	87,5
4.	E 8632 BE	E18632BE	100
5.	AD 1631 OC	D1G31OC	75
6.	NAI3 NRU	NAI3 NRU	100
7.	APO5 JEO	APO5 JEO	100
8.	AK64 DNV	AK64 DNV	100
9.	NA54 KGJ	NA54 KGJ	100
10.	RO2 FKD	RO2 FKD	100

Dalam **Tabel VIII**, jumlah karakter dan posisi yang sesuai dihitung sebagai jumlah karakter pada plat nomor yang terdeteksi dengan benar pada posisi yang benar dan total karakter plat nomor sebenarnya dihitung sebagai jumlah karakter sebenarnya pada 10 mobil, sehingga nilai akurasi karakter pada plat nomor di masing-masing mobil dapat ditentukan. Nilai akurasi karakter secara keseluruhan dapat dihitung dengan menjumlahkan hasil akurasi karakter pada masing-masing mobil lalu dibagi dengan jumlah mobil yang diuji lalu dikali 100%. Berdasarkan **Tabel VIII** nilai akurasi karakter secara keseluruhan dapat dihitung sebagai berikut:

$$Ak = \frac{\sum_{i=1}^{10} Akurasi}{10} \times 100\%$$

$$Ak = 94,82\%$$

Dengan demikian dalam pengujian deteksi plat nomor kendaraan pada 10 mobil, hasilnya menunjukkan akurasi karakter sebesar 94.82%. Berdasarkan hasil pendeteksian karakter menggunakan algoritma HyperLPR dihasilkan nilai yang lebih baik dari CNN, *Optical Character Recognition* (OCR) dan OCR dengan modifikasi *Template Matching* yang mendapatkan nilai akurasi sebesar 88% dan 90% [18] [19].

IV. KESIMPULAN

Secara keseluruhan sistem sistem deteksi laju dan plat nomor kendaraan berdasarkan video rekaman menggunakan *YOLOv5-DeepSORT* dan *HyperLPR* terdiri dari tiga proses deteksi yaitu deteksi dan *tracking* kendaraan, deteksi laju kendaraan dan deteksi plat nomor kendaraan. Dalam proses deteksi kendaraan dilakukan dengan menggunakan *YOLOv5*, sedangkan dalam melakukan *tracking* dilakukan dengan menggunakan *DeepSORT*. Pada proses deteksi laju kendaraan dilakukan dengan perhitungan perpindahan jarak pada piksel objek kendaraan yang terdeteksi dalam setiap 10 frame. Selanjutnya dalam proses deteksi plat nomor kendaraan dilakukan dengan menggunakan *HyperLPR*.

Sistem deteksi laju kendaraan diuji dengan masukan berupa video rekaman yang diambil di area ruas jalan tol Cipali Kabupaten Cirebon. Ruas jarak untuk cakupan deteksi laju kendaraan yang digunakan sepanjang 80 meter. Pengujian dilakukan dengan 5 kali percobaan dengan masukan video rekaman yang berbeda dengan rata-rata durasi pada masing-masing video selama 10 detik. Sistem mendapatkan nilai MAPE sebesar 7,2 %, sedangkan pengujian deteksi plat nomor kendaraan dilakukan dengan menguji 10 kendaraan

yang terdeteksi pada beberapa masukan video area ruas jalan tol Cipali dan masukan video yang diambil dari youtube dan membandingkan hasil deteksi dengan hasil sebenarnya. Hasil pengujian tersebut didapatkan hasil akurasi karakter secara keseluruhan sebesar 94.82%. Berdasarkan hasil pendeteksian karakter menggunakan algoritma HyperLPR dihasilkan nilai yang lebih baik dari CNN, *Optical Character Recognition* (OCR) dan OCR dengan modifikasi *Template Matching* yang mendapatkan nilai akurasi sebesar 88% dan 90% [18] [19].

DAFTAR PUSTAKA

- [1] S. L. Gomes, E. S. Reboucas, E. C. Neto, J. P. Papa, V. H. C. Albuquerque, P. R. Filho, J. M. R. S. Tavares, "Embedded real-time speed limit sign recognition using image processing and machine learning techniques," *Neural Comput. Appl.*, vol. 28, hlm. 573–584, 2017, doi: 10.1007/s00521-016-2388-3.
- [2] I. Maulana, E. Purwanto, dan A. Anggriat, "Analisis komparasi kecelakaan lalu lintas di jalan tol cikopo – palimanan sebelum dan pada saat pandemi covid-19," *J. Penelit. Transp. Darat*, vol. 23, no. 2, hlm. 184–192, 2021, doi: 10.25104/jptd.v23i2.1920.
- [3] B. Setyabudi, "Kajian peran tempat istirahat (rest area) kendaraan guna menurunkan tingkat kecelakaan dan kelelahan pengemudi pada jalan tol ruas jakarta-cikampek," *War. Penelit. Perhub.*, vol. 23, no. 4, p. 371, 2019, doi: 10.25104/warlit.v23i4.1092.
- [4] S. G. Hao Yu, Xingqi Wang, Yangli Shao, Feiwei Qin, Bin Chen, "Research on license plate location and recognition in complex environment," *J. Real-Time Image Process.*, vol. 19, hlm. 823–837, 2022, doi: https://doi.org/10.1007/s11554-022-01225-z.
- [5] L. Yang, J. Luo, X. Song, M. Li, P. Wen, dan Z. Xiong, "Robust vehicle speed measurement based on feature information fusion for vehicle multi-characteristic detection," *Entropy*, vol. 23, no. 7, hlm. 1–23, 2021, doi: 10.3390/e23070910.
- [6] R. A. Dinata, I. Candradewi, dan B. N. Prastowo, "Sistem pengawasan physical distancing di tempat umum menggunakan kamera berbasis deep learning," vol. 12, no. 1, hlm. 103–114, 2022, doi: 10.22146/ijeis.70886.
- [7] A. M. Santos, C. J. A. Bastos-filho, A. M. A. Maciel, dan E. Lima, "Counting vehicle with high-precision in brazilian roads using YOLOv3 and Deep SORT," hlm. 69–76, 2020, doi: 10.1109/SIBGRAP151738.2020.00018.
- [8] N. S. Punni, S. K. Sonbhadra, S. Agarwal, dan G. Rai, "Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques," hlm. 1–10, 1927.
- [9] E. Sugiarto dan F. Budiman, "Optimasi metode support vector machine dengan discrete wavelet transform untuk pengenalan karakter plat nomor kendaraan," vol. 18, no. 2, hlm. 133–142, 2021.
- [10] C. Gou, K. Wang, Y. Yao, and Z. Li, "Vehicle license plate recognition based on extremal regions and restricted boltzmann machines," vol. 17, no. 4, hlm. 1096–1107, 2016.
- [11] J. I. Zong Chen, "Automatic vehicle license plate detection using K-Means Clustering algorithm and CNN," *J. Electr. Eng. Autom.*, vol. 3, no. 1, hlm. 15–23, 2021, doi: 10.36548/jeea.2021.1.002.
- [12] Y. Qian, MA Dan-feng, W. Bin, P. Jun, C. Jian-Hai, Z. Wu-Jie, L. Jing-Sheng, "Spot evasion attacks: Adversarial examples for license plate recognition systems with convolutional neural networks," *Comput. Secur.*, vol. 95, p. 101826, 2020, doi: 10.1016/j.cose.2020.101826.
- [13] M. Zulfikri, E. Yudaningsy, dan R. Rahmadwati, "Sistem penegakan speed bump berdasarkan kecepatan kendaraan yang diklasifikasikan Haar Cascade Classifier," *J. Teknol. dan Sist.*

- Komput., vol. 7, no. 1, hlm. 12–18, 2019, doi: 10.14710/jtsiskom.7.1.2019.12-18.
- [14] N. Hilman Tsani, I. M. Burhanuddin Dirgantoro T, dan A. S. Luhur Prasasti T, “Impelementasi deteksi kecepatan kendaraan menggunakan kamera webcam dengan metode frame difference the implementation of vehicle speed detection using webcam with Frame Difference Method,” e-Proceeding Eng., vol. 4, no. 2, hlm. 2373–2381, 2017.
- [15] N. D. W. I. Cahyo, “Pengenalan nomor plat kendaraan dengan metode optical character recognition,” vol. 2, hlm. 75–84, 2019.
- [16] A. P. Kusumah, D. Djayusman, G. R. Setiadi, A. C. Nugraha, and P. Hidayatullah, “Counting Various Vehicles using YOLOv4 and DeepSORT,” J. Integr. Adv. Eng., vol. 3, no. 1, hlm. 1–6, 2023, doi: 10.51662/jiae.v3i1.68.
- [17] G. E. Setyawan, B. Adiwijaya, dan H. Fitriyah, “Sistem Deteksi Jumlah, Jenis dan Kecepatan Kendaraan Menggunakan Analisa Blob Berbasis Raspberry Pi,” J. Teknol. Inf. dan Ilmu Komput., vol. 6, no. 2, pp. 211–218, 2019, doi: 10.25126/jtiik.2019621405.
- [18] K. Ibnutama, Z. Panjaitan, dan E. F. Ginting, “Modifikasi Metode Template Matching pada OCR Untuk Meningkatkan Akurasi Deteksi Plat Nomor Kendaraan,” J. Teknol. Sist. Inf. dan Sist. Komput. TGD, vol. 2, no. 2, pp. 21–29, 2019.
- [19] Y. Galahartlambang, T. Khotiah, Zahrudin Fanani, dan Afifatul Aprilia Yani Solekhah, “Deteksi Plat Nomor Kendaraan Otomatis Dengan Convolutional Neural Network Dan Ocr Pada Tempat Parkir ITB Ahmad Dahlan Lamongan,” J. Manaj. Inform. dan Sist. Inf., vol. 6, no. 2, pp. 114–122, 2023, doi: 10.36595/misi.v6i2.754.