

Deteksi Helm Pengendara dan Plat Nomor Kendaraan pada CCTV Lampu Lalu Lintas Menggunakan Algoritma YOLO

Muhammad Priyo Anugrah¹, Bagus Fatkhurrozi², Hery Teguh Setiawan³

^{1,2,3} Universitas Tidar Magelang Jawa Tengah, Indonesia

*Corresponding author e-mail : priyo.anugrah1122@gmail.com

ABSTRAK

Perlengkapan berlalu lintas merupakan hal yang wajib digunakan pada saat ini. Keselamatan menjadi alasan utama diharuskannya pengendara mengenakan perlengkapan berkendara khususnya helm. Namun masih banyak pengendara sepeda motor yang melanggar peraturan berlalulintas dengan tidak menggunakan helm saat berkendara. Perlu adanya sistem yang dapat mengawasi pengendara yang tidak mengenakan helm saat mengendarai sepeda motor. Hal ini bisa diterapkan menggunakan salah satu algoritma deteksi objek yaitu YOLO yang menggunakan Bahasa pemrograman Python. YOLO digunakan sebagai alat yang akan mendeteksi dan mengenali objek seperti pengendara sepeda motor yang tidak menggunakan helm saat berkendara dari CCTV lampu lalulintas. Penelitian ini dilakukan dengan harapan peneliti mengetahui apakah algoritma deteksi objek seperti yolov5 bisa dimanfaatkan sebagai alat untuk mendeteksi pelanggaran berkendara seperti helm dan menangkap gambar pelanggar apabila kamera diambil dari CCTV lampu lalulintas. Penelitian ini akan memiliki 2 tahap penelitian yaitu, tahap pelatihan dataset dan tahap pengujian hasil training. Pada tahap pelatihan berfokus dalam melakukan training custom data untuk algoritma yolo dan tahap pengujian dilakukan untuk menguji apakah algoritma yolo yang telah dilakukan pelatihan sesuai dengan yang di harapkan. Hasil pengujian didapatkan nilai akurasi rata-rata dari semua class objek yang dipakai sebesar 94.52% dan error yang dihasilkan oleh aplikasi yang dibuat adalah sebesar 4,66%.

Kata kunci : CCTV, Deteksi objek, helm, Python, YOLOv5.

ABSTRACT

The use of traffic safety equipment is now mandatory. Safety is the main reason why riders wear safety equipment, especially helmets. However, there are still many motorcyclists who violate traffic regulations by not wearing helmets while riding. Therefore, a system is needed that can monitor riders who do not wear helmets while riding motorcycles. This system can be implemented using one of the object detection algorithms, YOLO, which uses the Python programming language. YOLO can be used as a tool to detect and recognize objects, such as motorcyclists who do not use helmets while driving, through CCTV traffic lights. This research was conducted with the aim to find out whether object detection algorithms, such as YOLOv5, can be utilized as a tool to detect driving violations, such as not wearing a helmet, and capture images of violators through traffic light CCTV cameras. This research consists of 2 stages, namely the dataset training stage and the training result testing stage. The training stage focuses on training data specific to the YOLO algorithm, while the testing stage aims to test whether the YOLO algorithm that has been trained is as expected. The results of testing the application show that the average accuracy value of all object classes used is 94.52%, and the error generated by the application is 4.66%.

Keywords: CCTV, Object detection, helmet, YOLOv5, Python.

I. PENDAHULUAN

Di Indonesia pada saat ini terdapat banyak sekali jenis transportasi darat, mulai dari kendaraan bermotor, mobil, dan angkutan umum. Pengguna kendaraan tersebut juga terdapat dari beberapa

kalangan mulai dari remaja sampai orang tua. Peningkatan kapasitas kendaraan bermotor setiap tahun yang terus mengalami peningkatan. Suasana ini membuat keadaan di jalan raya menjadi padat dan tidak bisa dikontrol.[1] Peningkatan kapasitas

penggunaan alat transportasi berupa motor selalu bisa memberi dampak pada meningkatnya pengendara bermotor yang melanggar dan terjadinya hal yang tidak diinginkan seperti kecelakaan saat berlalu lintas. Meningkatnya jumlah pengendara yang melanggar peraturan berlalulintas adalah salah satu alasan mengapa kecelakaan lalu lintas bisa terjadi. Saat meningkatnya kapasitas alat transportasi meningkat pula kemungkinan pengendara melanggar peraturan lalu lintas.

Menurut Pasal 106 ayat (8) UU No. 22/2009 semua pengguna kendaraan sepeda motor pengemudi ataupun penumpang diwajibkan menggunakan helm. Namun pengendara motor biasanya hanya menggunakan kelengkapan berkendara apabila terdapat aparat kepolisian yang sedang berjaga. Perlu adanya sebuah sistem yang mengawasi kegiatan berlalulintas secara *real time* dan terus menerus secara 24 jam. Sistem yang dibutuhkan harus memiliki akurasi dalam mengenali objek seperti perlengkapan berkendara dan dibutuhkan juga sebuah sistem yang murah guna mengurangi anggaran serta diperlukan juga sebuah sistem yang dapat dengan otomatis merekam nomor plat kendaraan bermotor. Sistem yang bisa dipakai adalah image detection yang bisa di terapkan pada CCTV lampu lalu lintas. *Image detection* bisa menggunakan algoritma *You Only Look Once (YOLO)*. [1]

You Only Look Once (YOLO) adalah sebuah metode buatan dari Joseph Redmon pada tahun 2016 yang digunakan sebagai alat mengolah deteksi objek. YOLO bisa digunakan sebagai alat memproses deteksi objek secara *real time*. apabila algoritma ini dibedakan dengan sistem deteksi objek *real time* yang lain, YOLO mempunyai mAP dan FPS yang lebih besar dibandingkan yang lain. Dengan pengertian model YOLO ini, *Transfer Learning* bisa diproses dengan model dan model bisa mendeteksi dan mengklasifikasikan objek yang diinginkan. [2]

OpenCV adalah *library Open Computer Vision*, yaitu *library* pemrosesan gambar gratis yang dikelola perusahaan Intel Corporation. Fungsi dari *library open CV* ialah menambahkan fungsi pada computer sehingga memiliki alat yang serupa dengan pemrosesan penglihatan manusia. Modul *library OpenCV* ini dibangun dengan sangat efisien dan fleksibel untuk menyelesaikan sebagian besar masalah vision komputer yang solusinya sudah tersedia, seperti pemangkasan gambar, peningkatan kualitas gambar dengan mengubah kecerahan, ketajaman, kontras, deteksi bentuk, segmentasi gambar, objek bergerak, mengenali objek dan lain-lain. [6]

Python adalah suatu *program language* pada software yang interaktif dan berorientasi objek dan

biasanya digunakan pada semua sistem operasi seperti Linux, Windows, Mac dan sistem operasi lainnya. Python mudah dipelajari karena sintaksnya yang mudah dipahami dan digabungkan menggunakan modul struktur data yang canggih, kuat, dan mudah digunakan. Kode sumber aplikasi bahasa pemrograman Python biasanya dikompilasi menjadi bentuk perantara yang disebut *bytecode*, yang kemudian dieksekusi. [5]

Menurut penelitian yang dilakukan oleh Tirtana dkk(2021) yang membahas mengenai algoritma YOLO sebagai *object detection* dan Tesseract-OCR yang digunakan untuk mendata nomor kendaraan bermotor di tempat umum yang menggunakan hardware Raspberry Pi. Pada saat Model YOLO telah ditraining akan mempunyai angka *precicion* yang tinggi dalam deteksi objek yaitu plat nomor, yang memiliki angka mencapai 100% apabila nilai *threshold* diatur 50%; 94, 5% jika nilai *threshold* diset 60%; dan 80, 31% jika batasan nilai *threshold* diset 70%. Untuk memproses program deteksi objek memerlukan waktu dalam melakukan YOLO, waktu yang dalam memproses yaitu dibutuhkan rata-rata 7, 11 detik untuk memprosesnya. [10]

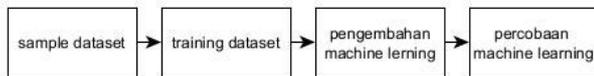
Menurut penelitian yang dilakukan oleh Cao dkk (2019) yang meneliti tentang algoritma yolo dimana pada algoritma yolo dilakukan monitoring dan perbandingan dengan algoritman image deteksi yang lain berdasarkan dari traffic flow. Saat memakai model YOLO- UA yang dipakai untuk lintas statistik, posisi kamera harus disesuaikan sehingga kamera tidak sangat besar serta bidang pandang tidak sangat besar. Bila letaknya sangat besar, mereka bisa menimbulkan deteksi yang terlampaui. Deteksi yang salah terjalin bila bidang pandang sangat besar. Bila permasalahan posisi ini dicatat serta disesuaikan, maksimal dampak statistik bisa dicapai dengan menerapkan nilai ambang batasan.

Berdasarkan hasil penelitian tersebut maka penelitian ini merancang system untuk melakukan object detection pada objek berupa helm dan kepala. Serta penelitian ini bertujuan untuk mengetahui tentang penerapan algoritma yolo apabila dipasangkan pada CCTV lampu lalulintas. Dimana penelitian dilakukan dengan dua tahap yaitu tahap pelatihan dan pengembangan.

II. METODE

Penelitian ini dibagi menjadi empat tahap yaitu tahap pengambilan data seperti data pada setiap class yang akan dipakai sebagai sample training seperti pengendara, helm, plat nomor, dan kepala. Selanjutnya adalah tahap training data yang

menggunakan machine learning dari yolov5 dan google colab. Selanjutnya tahap pengembangan machine learning yang menggunakan Bahasa pemrograman python dan tahap pengujian aplikasi dengan mengambil 4 sampel gambar dari video yang telah dilakukan pengujian. Gambar 1 merupakan alur tahap penelitian.



Gambar 1. metode penelitian

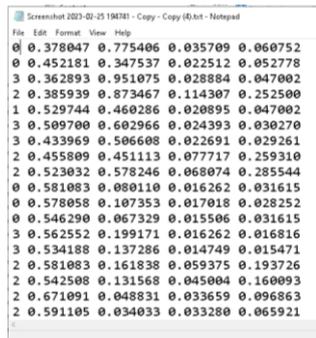
A. Pengambilan Sampel Dataset

Sampel dataset diambil dari Dinas Perhubungan Kota Magelang berupa video CCTV yang akan dipakai sebagai sample training data. Video tersebut kemudian dikonversi menjadi gambar sehingga bisa dilakukan penglabelan pada setiap class objek yang akan dideteksi. Total dari objek yang akan digunakan untuk penglabelan sejumlah 4000 sample dengan dibagi tiap masing-masing class sejumlah 500 sample setiap classnya. Gambar 2 merupakan contoh gambar sample yang akan dilakukan penglabelan.



Gambar 2. Dataset Dari CCTV

Gambar yang sudah dikonversi kemudian dilakukan penglabelan sehingga mempermudah dalam melakukan training data. Penglabelan dataset disesuaikan dengan class yang akan dideteksi dalam hal ini terdapat 4 class. Pada label pertama menggunakan class helm, label kedua menggunakan class head atau no helm, label ketiga menggunakan class pengendara dan label keempat menggunakan class number plate. Gambar 3 merupakan hasil dari penglabelan.



Gambar 3. Label Pada Sample Gambar

Pada gambar 3 dapat dilihat isi dari file label yang telah didownload. Dapat dilihat terdapat angka – angka sesuai dengan penglabelan yang telah dilakukan. Angka pada baris pertama menjelaskan nama class pada saat penglabelan yaitu dimulai dari 0 yang merupakan class dari helm, 1 merupakan class dari head, 2 merupakan class dari pengendara, dan 3 merupakan class dari plat nomor atau number plate. Sedangkan angka selanjutnya merupakan titik kordinat bounding box pada gambar. Pada yolov5 titik kordinat ditandai dengan symbol x1, y1, x2, dan y2. Label – label yang telah didapatkan kemudian disatukan dengan sampel gambar dan disusun pada sebuah folder dengan nama folder image untuk menandakan folder gambar dan label yang menandakan folder label. Didalam setiap folder dibagi lagi menjadi dua folder yaitu train dan folder val yang merupakan folder training dan validasi.

B. Training Dataset

Pada penelitian ini, training dataset menggunakan machine learning milik yolov5 dan google colab. Training dilakukan dengan menggunakan 60 iterasi dan menghasilkan weight file dengan format pytorch. Gambar 4. Merupakan hasil dari training dataset.

```

Completed in 1.073 hours.
tripped from runs/train/exp3/weights/last.pt, 14.4MB
tripped from runs/train/exp3/weights/best.pt, 14.4MB

runs/train/exp3/weights/best.pt...
PS...
ry: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
Class Images Instances P R mAP50 mAP50-95: 14
all 684 5066 0.912 0.882 0.934 0.727
helm 684 1337 0.918 0.845 0.92 0.702
no_helm 684 651 0.87 0.865 0.9 0.614
pengendara 684 1670 0.873 0.974 0.986 0.954
number_plate 684 1408 0.987 0.846 0.93 0.636
ed to runs/train/exp3
  
```

Gambar 4. Hasil Dari Proses Training Yolov5

Hasil training yolov5 yang terdapat pada gambar 4 menjelaskan beberapa variable yang dihasilkan selama training custom data. Untuk recall dan precision yang dihasilkan dari keempat model class rata-rata berjumlah 0.882 dan 0.912. Untuk menentukan bagus atau tidaknya suatu model diperlukan sebuah percobaan pada model yang telah di training.

C. Pengujian Machine Learning

Untuk melakukan pengujian aplikasi diperlukan suatu variable penelitian sebagai tolak ukur dalam menentukan kebagusan dari sebuah aplikasi. Penelitian ini memiliki beberapa variable yaitu:

a. Akurasi

Akurasi ialah beberapa tolak ukur uji yang membuktikan kelayakan pada suatu fungsi dalam melakukan klasifikasi objek deteksi [3]. Parameter akurasi berguna untuk menetapkan kebagusan dari suatu sistem dalam melakukan deteksi objek terhadap

sampel dataset. Akurasi dapat ditentukan menggunakan persamaan [4]

$$A = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (1)$$

b. *Presisi*

Presisi ialah perbandingan dari total objek yang terdeteksi dengan benar atau *True Positive* dibandingkan pada semua data yang terprediksi positif [4]. Berdasarkan persamaan (2) bisa disimpulkan bahwa semakin besar nilai *False Positive*, maka presisi semakin rendah dan begitu pun sebaliknya.

$$A = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

c. *Recall*

Recall adalah perbandingan dari objek yang dideteksi dengan benar atau *True Positive* berbanding dengan semua data yang positif. Nilai recall yang tinggi menandakan bahwa suatu sistem bisa melakukan klasifikasi kelas objek dengan benar. Recall dapat dikalkulasikan dengan persamaan [4]

$$A = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

d. **F1Score**

F1 Score adalah rata-rata perbandingan dari presisi dengan *recall*. *F1 Score* mempunyai nilai tertinggi sebesar 1 dan terendah sebesar 0. Jika nilai *F1 Score* semakin mendekati 1, menandakan kinerja sistem telah baik. Secara matematis, nilai *F1 Score* dapat dilihat pada persamaan berikut: [5]

$$F1\ Score = 2 \times \frac{Recall \times Presisi}{Recall + Presisi} \quad (4)$$

Setelah menentukan Pengujian aplikasi dilakukan menggunakan video yang diambil dari CCTV lampu lalu lintas dari dinas perhubungan kota Magelang. Video dipotong yang semula berdurasi 4 jam menjadi dua video yang masing-masing berdurasi 8 menit dan diambil bagian yang terdapat pengendara yang tidak menggunakan helm. Setelah dilakukan pengujian maka didapatkan data berupa nilai. Untuk lebih jelasnya dapat dilihat pada Tabel 1.

Tabel 1. Data Hasil Pengujian Terdeteksi

No	Frame	helm	No helm	pengendara	Number plate
1	14	4	1	6	5
2	74	6	1	7	4
3	430	4	2	5	5
4	535	7	1	7	6
5	585	2	1	2	2

6	600	3	1	3	3
7	670	13	2	12	9
8	865	6	0	8	5
9	1121	8	0	6	5
10	1145	6	2	8	4
Total	59	11	57	48	

Hasil pengujian dikelompokkan berdasarkan dari jenis class objek yang terdeteksi berdasarkan dari *frame video*. *Frame video* diambil dari potongan video yang dikonversikan menjadi gambar yang berupa *frame*. Pada *frame* tersebut diambil 10 sampel yang akan digunakan untuk menghitung nilai *confuse matrix*. Berdasarkan Tabel 1. dapat dilihat bahwa jumlah nilai prediksi yang didapatkan setiap class objek berbeda seperti objek helm yang terdeteksi sejumlah 59 objek, no helm 11 objek, pengendara 57 objek, dan *number plate* 48 objek. Nilai tersebut bisa digunakan untuk menentukan nilai *True Positive*(TP). Tabel 2. Merupakan tabel yang digunakan untuk menentukan nilai *False Negative*(FN).

Tabel 2. Data Hasil Pengujian Tidak Terdeteksi

No	Frame	helm	No helm	pengendara	Number plate
1	14	0	0	0	0
2	74	2	0	1	3
3	430	1	0	1	1
4	535	1	0	2	1
5	585	0	0	0	0
6	600	4	0	3	1
7	670	2	0	0	2
8	865	0	0	0	3
9	1121	1	0	0	0
10	1145	0	0	1	4
Total	11	0	8	15	

Selanjutnya terdapat juga objek yang tidak terdeteksi oleh program seperti objek helm yang tidak terdeteksi sejumlah 11 objek, pengendara 8 objek, number plate 15 objek. Nilai tersebut dapat digunakan untuk menentukan nilai *False negative*(FN). Untuk menentukan nilai *False positive*(FP) dapat dilihat berdasarkan Tabel 3.

Tabel 3. Data Hasil Pengujian Prediksi Salah

No	Frame	helm	No helm	pengendara	Number plate
1	14	0	0	0	0
2	74	0	0	0	0
3	430	0	0	0	0
4	535	0	0	0	0
5	585	0	0	0	0

6	600	1	1	0	0
7	670	0	0	0	0
8	865	0	1	0	0
9	1121	1	0	0	0
10	1145	1	1	0	0
Total	3	3	0	0	0

Tabel 4. dapat digunakan untuk menentukan nilai False Positive(FN). FN merupakan prediksi yang salah dari class yang dideteksi oleh program. Selanjutnya untuk nilai True Negative(TN), didapatkan dengan menjumlahkan nilai TP dari objek yang tidak termasuk dalam class yang terdeteksi

III. HASIL DAN PEMBAHASAN

Spesifikasi peralatan yang dipakai untuk melakukan pengujian sistem ini, yaitu bahasa pemrograman Python 3.8, dan hardware, dengan spesifikasi: processor AMD Radeon A5, memori internal SSD 128 GB dan HDD 500 GB, RAM 12 GB dan memiliki VGA bawaan dari motherboard yaitu AMD Radeon Graphics.

D. Hasil Perhitungan

Setelah melakukan pengujian maka didapatkan hasil berupa confuse matrix yang berisi True Positive(TP), True Negative(TN), False Positive(FP), dan False Negative(FN). Tabel 4. Merupakan nilai confuse matrix yang didapat.

Tabel 4. Nilai Confuse Matrix

no	objeck	TP	FP	FN	TN
1	pengendara	57	0	8	118
2	number_plate	48	0	15	127
3	helm	59	3	11	116
4	no_helm	11	3	1	164

Setelah didapatkan nilai TP, FP, FN, dan TN, maka selanjutnya dapat menentukan nilai akurasi, *precision*, *recall*, *f1score*, dan *error* dari model objek deteksi yang telah *training*. Untuk hasil perhitungan yang didapatkan dapat dilihat pada Tabel 5.

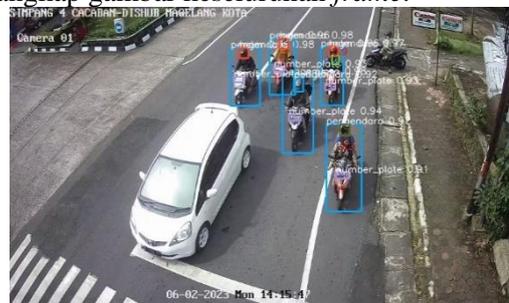
Tabel 5. Hasil Perhitungan

objek	akurasi	recall	precision	f1score	error
pengendara	0,956	0,876	1	0,934	0,043
Number plate	0,921	0,761	1	0,864	0,078
helm	0,925	0,842	0,951	0,893	0,058
no_helm	0,977	0,916	0,785	0,846	0,005
nilai rata – rata	0,945	0,849	0,934	0,884	0,046

Pada Tabel 5. didapatkan nilai akurasi rata-rata sebesar 0.9452 atau 94.52%. Untuk nilai *recall* didapatkan rata-rata sebesar 0,8495 atau 84,95%. Untuk nilai *precision* didapatkan rata-rata sebesar 0,9343 atau 93,43%. Selanjutnya nilai *f1score* yang terhitung memiliki nilai rata-rata sebesar 0,8848 atau 88,248%. Hal ini menunjukkan bahwa model dataset yang telah *training* sudah sesuai dengan yang diharapkan. Untuk nilai *error* pada model *training* yang didapatkan memiliki nilai rata-rata sebesar 0,0466 atau 4,66% dimana batas *error rate* yang diharapkan sebesar 7.32%.

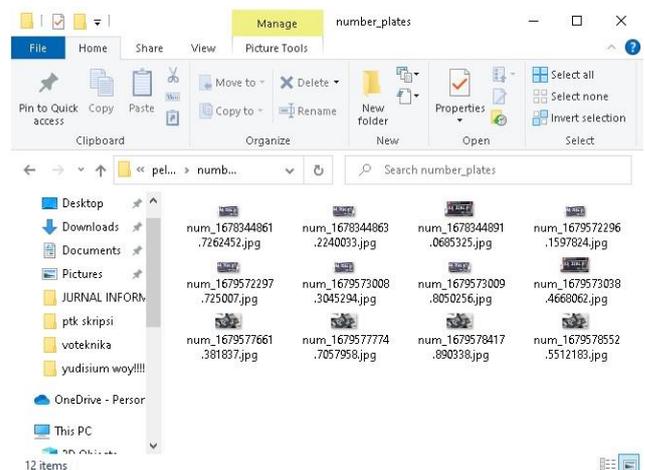
E. Hasil Pengujian Aplikasi

Saat aplikasi dijalankan, mesin *learning* akan melakukan tangkap layar yang dilakukan dengan tiga kondisi yaitu: menangkap gambar pengendara, menangkap gambar plat nomor dan menangkap gambar keseluruhan *frame*.



Gambar 5. Gambar Keseluruhan Frame

Kondisi yang pertama dalam menangkap gambar pelanggar adalah dengan cara menangkap gambar pengendara seluruh *frame*. Kondisi ini akan terjadi apabila terdapat pengendara yang tidak menggunakan helm dan terdeteksi dan di beri *bounding box* oleh sistem. Dengan program yang telah dibuat apabila *bounding box* pelanggar atau *class head* terdeteksi berada didalam *bounding box* dari *class* pengendara maka sistem akan otomatis melakukan *capture* layar seluruh *frame*. Untuk hasil *capture* layar dapat dilihat pada gambar 5.



Gambar 6. Gambar Capture Plat Nomor

Kondisi yang kedua adalah menangkap plat nomor pengendara yang tidak menggunakan helm. Kondisi ini bisa terjadi apabila terdapat *bounding box* pada *class head* yang terdeteksi dan berada didalam *bounding box class* pengendara. Bila hal ini telah terjadi maka aplikasi akan melakukan *capture* layar pada bagian *class number plate* yang berada didalam *bounding box* dari *class* pengendara yang terdeteksi memiliki *bounding box* pelanggar atau *class head* didalamnya. Untuk contoh *capture* plat nomor dapat dilihat pada gambar 6.



Gambar 7. Gambar Capture Pengendara

Kondisi yang ketiga adalah melakukan *cropping* pada bagian pengendara yang tidak menggunakan helm. Hal ini terjadi apabila terdapat *bounding box* pelanggar atau *class head* yang terdeteksi berada didalam *bounding box class* pengendara. Apa bila kondisi ini terjadi maka aplikasi akan melakukan *cropping* gambar pada bagian pengendara yang didalamnya terdapat *bounding box* pelanggar atau *class head*. Untuk contoh *cropping* pada pengendara dapat dilihat pada gambar 8.

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dapat ditarik kesimpulan yaitu:

- Yolov5 dapat digunakan untuk mendeteksi objek dari pengendara motor dari CCTV lampu lalu lintas dan menangkap objek tersebut. Hasil dari objek yang ditangkap memiliki resolusi yang rendah sehingga tidak bisa digunakan untuk melakukan pembacaan plat nomor otomatis. Hal ini disebabkan oleh jarak dari kamera CCTV yang terlalu jauh dan gambar yang diambil menggunakan metode tangkap layar yang membuat kualitas gambar menjadi buram ketika dizoom.
- Nilai akurasi yang didapatkan berdasarkan dari nilai *confusion* matrix yang telah dihitung

memiliki nilai rata-rata sebesar 94,52 %. Dengan nilai akurasi rata-rata setiap class yang dideteksi sebesar 95,62% untuk pengendara, 92,10% untuk *number plate*, 92,59% untuk helm, dan 97,76% untuk no helm.

V. SARAN

Berdasarkan hasil penelitian yang telah dilakukan dengan judul “Deteksi Helm Pengendara Dan Plat Nomor Kendaraan Pada CCTV Lampu Lalu lintas Menggunakan Algoritma YOLO” memiliki saran sebagai berikut:

- Perlu ditambah lagi sampel data gambar yang digunakan untuk *training data*. tambahan sampel bisa berupa jumlah sampel dan variasi bentuk dari sampel yang akan digunakan.
- Untuk pengujian bisa menggunakan kamera pribadi agar pada saat pengujian aplikasi objek deteksi gambar pelanggar yang tidak menggunakan helm dapat ditangkap dengan lebih jelas dan dapat dibaca plat nomornya.

DAFTAR PUSTAKA

- [1] S. Aulia, P. Maria, and R. Ramianti, “Aplikasi Pendeteksi Plat Nomor Kendaraan Berbasis Raspberry Pi Menggunakan Website Untuk Pelanggaran Lalu Lintas,” *Elektron J. Ilm.*, vol. 11, no. 2, pp. 84–89, 2019.
- [2] S. Jupiyandi, F. R. Saniputra, Y. Pratama, M. R. Dharmawan, and I. Cholissodin, “Pengembangan Deteksi Citra Mobil untuk Mengetahui Jumlah Tempat Parkir Menggunakan CUDA dan Modified YOLO,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 4, p. 413, 2019.
- [3] F. Rachmawati and D. Widhyaestoeti, “Deteksi Jumlah Kendaraan di Jalur SSA Kota Bogor Menggunakan Algoritma Deep Learning YOLO,” *Pros. LPPM UIKA Bogor*, pp. 360–370, 2020.
- [4] Y. Zhao and S. Geng, “Face occlusion detection algorithm based on yolov5,” *J. Phys. Conf. Ser.*, vol. 2031, no. 1, 2021.
- [5] T. A. A. H. Kusuma, K. Usman, and S. Saidah, “People Counting for Public Transportations Using You Only Look Once Method,” *J. Tek. Inform.*, vol. 2, no. 1, pp. 57–66, 2021.
- [6] A. S., J. Yankey, and E. O., “An Automatic Number Plate Recognition System using OpenCV and Tesseract OCR Engine,” *Int. J. Comput. Appl.*, vol. 180, no. 43, pp. 1–5, 2018.
- [7] D. Aryanto and V. Augusman, “Penerapan Machine Learning Untuk Mengategorikan Sampah Plastik Rumah Tangga,” *J. TIMES*, vol. 10, no. 1, pp. 39–43, 2021.

- [8] N. D. Miranda, L. Novamizanti, and S. Rizal, "Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50," *J. Tek. Inform.*, vol. 1, no. 2, pp. 61–68, 2020.
- [9] S. Kuan-Hung, C. Ching-Te, and P. Yen-Yu, "REAL-TIME OBJECT DETECTION VIA PRUNING AND A CONCATENATED MULTI-FEATURE ASSISTED REGION PROPOSAL NETWORK Kuan-Hung Shih , Ching-Te Chiu , Yen-Yu Pu Department of Computer Science , National Tsing Hua University , Hsinchu , Taiwan," pp. 1398–1402, 2019.
- [10] Tirtana, E., Gunadi, K., & Sugiarto, I. (2021). Penerapan Metode YOLO dan Tesseract-OCR untuk Pendataan Plat Nomor Kendaraan Bermotor Umum di Indonesia Menggunakan Raspberry Pi. *Jurnal Infra*, 9(2), 241-247.