

Sistem Perhitungan Kendaraan Menggunakan Algoritma YOLOv5 dan DeepSORT

<http://dx.doi.org/10.28932/jutisi.v10i1.7519>

Riwayat Artikel

Received: 27 September 2023 | Final Revision: 29 April 2024 | Accepted: 30 April 2024

Creative Commons License 4.0 (CC BY – NC)



Putri Anggia Cahyani^{#1}, Mardiana^{✉#2}, Puput Budi Wintoro^{#3}, Meizano Ardhi Muhammad^{#4}

[#] Program studi Teknik Informatika, Universitas Lampung

Jl. Prof. Dr. Ir. Sumantri Brojonegoro, Kec. Rajabasa, Kota Bandar Lampung, Lampung, 35144, Indonesia

¹putri.anggia104219@students.unila.ac.id

²mardiana@eng.unila.ac.id

³budi.wintoro@eng.unila.ac.id

⁴meizano@eng.unila.ac.id

✉Corresponding author: mardiana@eng.unila.ac.id

Abstrak — Pencemaran udara menjadi isu serius di kota-kota besar, seperti di kota Bandar Lampung. Hal ini diakibatkan oleh tingginya aktivitas transportasi menggunakan kendaraan bermotor. Data tahun 2021 menunjukkan peningkatan 4,30% jumlah kendaraan bermotor di Indonesia, berdampak pada emisi karbon. Sebagai respons terhadap permasalahan tersebut, Greenmetric Universitas Lampung memiliki program kerja transportasi hijau yang berfokus dalam mengurangi emisi kendaraan bermotor. Untuk mendukung tujuan tersebut dilakukan pemantauan lalu lintas secara otomatis dengan menerapkan bidang *Computer Vision*, yaitu *object tracking*. Pembuatan sistem *object tracking* dalam melakukan pemantauan lalu lintas menggunakan kombinasi algoritma YOLOv5 dan DeepSORT. Metode yang digunakan dalam penelitian ini adalah metode Scrum yang dilakukan sebanyak tiga kali *sprint*, dan dibagi menjadi tiga tahap yaitu *pre-game*, *game*, dan *post-game*. Hasil dari penelitian ini adalah sebuah sistem *object tracking* yang berhasil membedakan dan menghitung tiga jenis kendaraan (motor, mobil, dan bis) secara otomatis, dan telah dilakukan pengujian secara *realtime* dengan nilai rata-rata *precision* sebesar 99%, *recall* sebesar 97%, *F1 score* sebesar 97.2%, *accuracy* sebesar 96.8%, dan rata-rata ketepatan perhitungan sistem sebesar 97.65%.

Kata kunci— *Computer Vision*; DeepSORT; Emisi Karbon; *Object Tracking*; YOLOv5.

Vehicle Counting System Using YOLOv5 and DeepSORT Algorithms

Abstract — Air pollution is a serious issue in big cities, such as Bandar Lampung. This is caused by high transportation activities using motorized vehicles. Data from 2021 shows a 4.30% increase in the number of motorized vehicles in Indonesia, impacting carbon emissions. In response to this problem, Greenmetric Lampung University has a green transportation work program that focuses on reducing motor vehicle emissions. To support this goal, automatic traffic monitoring is carried out by applying the field of *Computer Vision*, namely *object tracking*. The making of an *object tracking* system in monitoring traffic uses two combinations of the YOLOv5 and DeepSORT algorithms. The method used in this research is the Scrum method which is carried out in three sprints and divided into three stages, namely *pre-game*, *game*, and *post-game*. The result of this research is an *object tracking* system that successfully distinguishes and counts three types of vehicles (motorcycle, car, and bus) automatically, and has been tested in real-time with an average precision value of 99%, recall of 97%, F1 score of 97.2%, accuracy of 96.8%, and average accuracy of system calculation of 97.65%.

Keywords— *Carbon Emissions*; *Computer Vision*; DeepSORT; *Object Tracking*; YOLOv5.

I. PENDAHULUAN

Pencemaran udara menjadi permasalahan yang seringkali dijumpai di kota-kota besar. Hal ini disebabkan oleh tingginya intensitas aktivitas transportasi yang melibatkan penggunaan kendaraan individu maupun sarana transportasi publik. Berdasarkan data yang bersumber dari situs *web* dataindonesia.id pada tahun 2021, jumlah kendaraan bermotor di Indonesia meningkat sebesar 4,30% dari tahun sebelumnya. Motivasi utama penelitian ini adalah merespon dampak negatif dari peningkatan jumlah kendaraan bermotor terhadap lingkungan. Meningkatnya jumlah kendaraan bermotor dapat berdampak langsung pada pembuangan emisi karbon ke lingkungan. Kota Bandar Lampung sendiri tergolong sebagai salah satu kota besar yang mengalami peningkatan pencemaran udara yang signifikan. Meskipun saat ini terdapat kendaraan listrik sebagai solusi potensial dalam mengurangi emisi karbon, akan tetapi rasio populasi pengguna kendaraan listrik dengan kendaraan diesel di Indonesia masih tergolong rendah, sebesar 1:2304. Rasio ini didapat dari perbandingan jumlah kendaraan listrik di Indonesia sebesar 68.207 *unit* dan kendaraan diesel sebesar 157.484.407 *unit* berdasarkan data pada *web* inews.id [1] pada tahun 2023. Dengan melihat perbandingan yang signifikan ini, didapatkan persentase kendaraan diesel sebesar 99,96%, dan dapat dikatakan bahwa pengaruh kendaraan listrik masih sangat minim. Sebagai respons terhadap permasalahan ini, maka dibentuk *greenmetric* oleh Universitas Indonesia yaitu sebuah indeks yang digunakan untuk menilai kinerja berkelanjutan dalam mengurangi dampak negatif terhadap lingkungan, termasuk mengurangi emisi karbon dan pencemaran udara [2].

Greenmetric Universitas Lampung merupakan salah satu tim di Universitas Lampung yang mendorong peran serta seluruh *civitas* untuk melakukan kolaborasi dan kontribusi dalam pembangunan komunitas kampus yang berkelanjutan sehingga terbentuk kampus hijau yang berfokus pada kelestarian lingkungan hidup. Salah satu program kerja *greenmetric* adalah transportasi hijau, yaitu program kerja yang berfokus pada pengurangan emisi kendaraan bermotor. Untuk menunjang hal tersebut, perlu dilakukan pemantauan kepadatan lalu lintas di jalan kampus Universitas Lampung [3]. Pemantauan kepadatan lalu lintas dapat dilakukan secara otomatis dengan memanfaatkan pengaplikasian bidang studi *Computer Vision*.

Computer Vision atau Visi komputer adalah bidang studi yang berfokus pada masalah dalam membantu komputer untuk mereproduksi kemampuan penglihatan seperti layaknya manusia. Secara luas, *computer vision* merupakan sub bidang dari *Artificial Intelligence (AI)* dan *Machine Learning (ML)*, yang bertujuan untuk memahami konten gambar digital [4]. Salah satu elemen penting dalam *computer vision* adalah citra. Citra adalah gambaran visual yang merepresentasikan pantulan cahaya dalam suatu objek. Citra yang diciptakan oleh sebuah mesin digital disebut sebagai citra digital [5], yaitu *object tracking*. *Object tracking* atau pelacakan objek adalah salah satu pengaplikasian bidang studi visi komputer mendasar yang mengacu pada serangkaian metode untuk melacak lintasan gerak suatu objek dalam video secara tepat [6]. *Object tracking* dapat diaplikasikan untuk pengawasan lalu lintas, pengenalan tingkah laku dan sebagainya. *Object tracking* melakukan pelacakan lintasan gerak suatu objek dalam video secara tepat [7], [8]. *Multiple Object Tracking (MOT)* merupakan sub bagian dari metode *object tracking*, yang memiliki tujuan untuk melakukan pelacakan banyak objek dalam satu video dan merepresentasikannya menjadi sekumpulan lintasan berakurasi tinggi [6]. Permasalahan dari MOT adalah ketika objek yang sama tidak diberikan ID yang sama dalam seluruh *frame*. Permasalahan tersebut disebabkan karena pertukaran ID dan *occlusion* [7]. Kombinasi algoritma DeepSORT dan YOLOv5 menjadi salah satu kombinasi algoritma yang dapat digunakan dalam pembuatan sistem *object tracking*. Pemilihan kedua algoritma ini berdasarkan studi literatur yang telah dilakukan. DeepSORT merupakan algoritma lanjutan dari SORT yang dibuat untuk menutupi kekurangan pada SORT dengan mengurangi permasalahan tertukarnya ID sebesar 45% [9] sedangkan algoritma *object detection* yang terkenal adalah YOLO. YOLO memiliki banyak versi, dengan versi terbaru YOLO saat ini adalah YOLOv8 dengan tingkat akurasi yang menyaingi YOLOv5. Meskipun begitu, performa YOLOv5 dalam pendeteksian sistem *real time* sangat stabil dan lebih cepat dibandingkan versi YOLO lainnya [10].

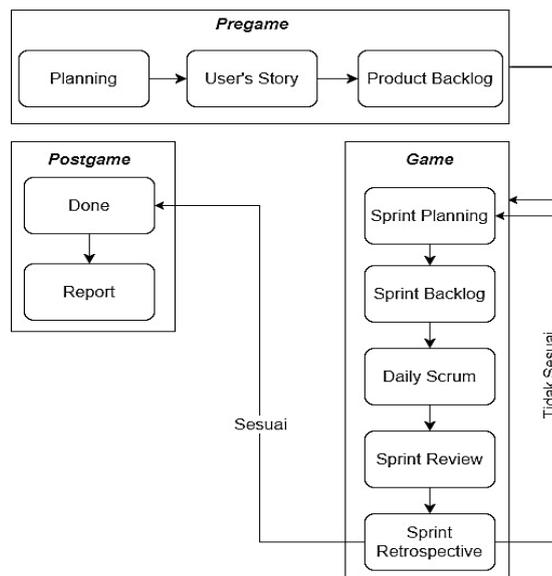
Dengan memanfaatkan kedua algoritma tersebut, dibangun sebuah sistem *object tracking* yang dapat membantu dalam pemantauan awal kepadatan jalan kampus Universitas Lampung terkait permasalahan yang dihadapi. Tujuan utama sistem ini adalah untuk secara efisien mengidentifikasi dan menghitung tiga jenis kendaraan penyumbang emisi karbon (mobil, motor, dan bus) yang paling sering melintasi jalan kampus. Cara kerja sistem melibatkan pengumpulan data perhitungan jumlah kendaraan yang terdeteksi, dan hasilnya disimpan dalam sebuah *database* terpusat. Informasi ini kemudian dapat diakses dan dievaluasi melalui halaman data kendaraan yang disajikan dalam bentuk tabel. Data perhitungan ini nantinya akan menjadi sumber informasi penting bagi tim *Greenmetric* Universitas Lampung, membantu mereka dalam pemantauan jumlah kendaraan dan dampaknya terhadap emisi karbon di sekitar kampus. Selain itu, tujuan lain pembuatan sistem ini adalah sebagai langkah awal dalam pengembangan sistem yang memiliki fungsi untuk menghitung pembuangan emisi karbon secara otomatis.

II. METODE PENELITIAN

Metode penelitian yang digunakan dalam pembuatan sistem deteksi dan perhitungan kendaraan di Kawasan jalan kampus Universitas Lampung adalah metode Scrum. Metode Scrum adalah kerangka proses untuk mengelola pengembangan produk kompleks. Siklus Scrum dimulai dari *Product Backlog*, yaitu tahapan yang menjabarkan keseluruhan pekerjaan yang perlu

dilakukan, kemudian terdapat tahapan *Sprint* berkisar 4 minggu. Dari *Sprint* dihasilkan *Sprint Backlog* yang di tahap ini juga DoD atau *Definition of Done* ditentukan [11].

Pada *Sprint Backlog* ini, dibagi menjadi beberapa bagian status seperti *In-Progress* dan *done*. Selanjutnya, terdapat *Daily Scrum* yang bertujuan untuk memantau progress pekerjaan yang telah dilakukan dan menentukan pekerjaan yang akan dilakukan nantinya. Proses selanjutnya adalah *Increment* yang dapat ditambahkan sewaktu-waktu jika memang terdapat pekerjaan yang perlu ditambahkan. Kemudian, *Sprint Review* yang dilakukan di akhir *Sprint*, yaitu pemeriksaan hasil *Sprint* yang telah dilakukan. Terakhir, *Sprint Retrospective*, yaitu mengevaluasi bagaimana hasil pekerjaan yang telah dilakukan dan mencari cara untuk dapat menjadi lebih baik lagi. Siklus tersebut kemudian kembali lagi pada *Sprint Planning*, dan seterusnya hingga DoD telah tercapai seluruhnya. Pada penelitian, tahapan proses Scrum dibagi menjadi 3 fase yaitu, *Pregame*, *Game*, dan *Postgame* seperti pada gambar 1.



Gambar 1. Diagram alur penelitian menggunakan metode Scrum

A. *Pregame*

Fase *pregame* berfokus pada aktivitas dalam mengidentifikasi permasalahan yang dihadapi dan melakukan perencanaan dalam penentuan solusi dari masalah yang dihadapi. Pada Fase *pregame* terdapat 3 tahapan yaitu *planning*, *user's story*, dan *product backlog*. Keluaran yang didapatkan dari fase ini berupa *product backlog* yang terdapat pada gambar 2.



Gambar 2. *Product backlog*

B. Game

Tahapan *game* merupakan tahapan pengembangan kebutuhan sistem yang dilakukan berdasarkan *product backlog* yang telah dirancang. Pengembangan sistem pada penelitian ini menggunakan metode Scrum yang digunakan untuk mengelola pengembangan produk kompleks dengan 5 tahapan dalam Scrum.

- 1) *Sprint Planning*. Pada tahap ini dilakukan penentuan item yang dikerjakan pada suatu sprint dengan melihat level prioritas *item*. Selanjutnya, dilakukan analisis pada item *product backlog* untuk menentukan langkah bagaimana item tersebut akan diselesaikan.
- 2) *Sprint Backlog*. Tahapan dalam menentukan daftar tugas yang perlu dilakukan untuk mencapai suatu *item product backlog* yang telah ditentukan selama 1 *sprint*.
- 3) *Daily Scrum*. Selama *sprint* berlangsung, diadakan *daily scrum* yang dilakukan dengan durasi 15 menit setiap harinya. Pada penelitian ini, *daily scrum* dilakukan selama hari kerja (Senin hingga Jumat) setiap minggunya.
- 4) *Sprint Review*. Peninjauan *progress* yang dibantu menggunakan *tools* Trello. Selain peninjauan, pada tahap ini dilakukan *testing* sistem secara *realtime*. Pengambilan *sample testing* menggunakan laptop, *tripod* dan *webcam* selama 1 jam setiap harinya dalam 5 hari. Hasil pengujian diukur menggunakan *confusion matrix* dan *black-box testing*.
- 5) *Sprint Retrospective*. Peninjauan dan refleksi Sumber Daya Manusia (SDM) yang dilakukan untuk meningkatkan kualitas dan efektivitas pengerjaan untuk *sprint* selanjutnya. Hasil dari tahapan ini berupa keputusan mengenai hal yang dilakukan pada *sprint* berikutnya.

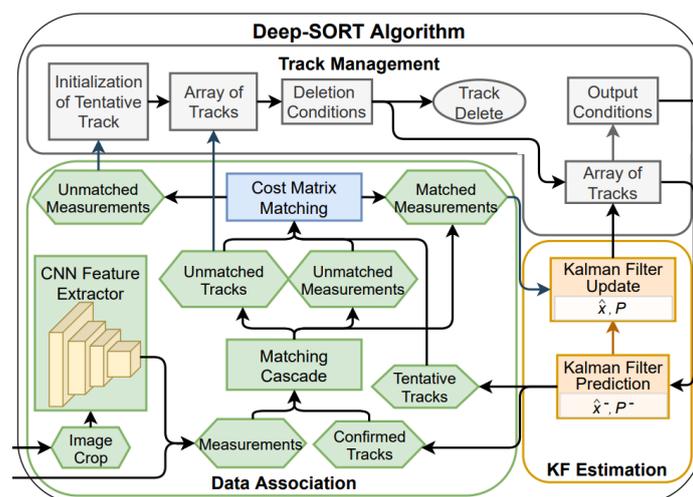
C. Post-game

Fase terakhir dari penelitian ini adalah fase *postgame*. Pada fase *postgame*, sistem yang dibangun telah memenuhi kebutuhan pengguna berdasarkan permasalahan yang dihadapi. Pada tahapan ini dilakukan pembuatan laporan berdasarkan aktivitas penelitian yang telah dilakukan. Hasil laporan yang telah dibuat akan dianalisa untuk pengambilan kesimpulan dan saran terhadap pengembangan sistem yang telah dibangun.

D. YOLOv5

YOLOv5 merupakan algoritma deteksi banyak objek (*multiple object detection*) yang dikembangkan oleh Ultralytics. Secara keseluruhan, arsitektur YOLOv5 terdiri dari 3 bagian utama yaitu *backbone*, *neck*, dan *head*. Pada YOLOv5 dilakukan berbagai teknik augmentasi data untuk meningkatkan kemampuan model dalam melakukan generalisasi dan mengurangi *overfitting*. Teknik-teknik augmentasi tersebut terdiri dari *mosaic augmentation*, *copy-paste augmentation*, *random affine transformations*, *mix-up augmentation*, *albumentations*, *HVS augmentation*, dan *random horizontal flip*. Selain itu, diterapkan juga beberapa strategi *training* yang dapat meningkatkan performa model dalam melakukan deteksi seperti *multiscale training*, *autoanchor*, *warmup and cosine LR scheduler*, *exponential moving average*, *mixed precision training*, dan *hyperparameter evolution* [12]. Pada penelitian ini, model YOLOv5 yang digunakan saat *training* adalah YOLOv5s.

E. DeepSORT



Gambar 3. Algoritma DeepSORT [13]

Deep SORT merupakan ekstensi dari algoritma SORT. Berdasarkan gambar 3, tahapan Deep SORT dibagi menjadi 3 komponen, yaitu *data association*, *KF estimation*, dan *track management*. Pada *data association*, dilakukan ekstraksi fitur menggunakan CNN. DeepSORT menggunakan algoritma *Hungarian* dalam mencocokkan tampilan dan informasi pergerakan objek. Selanjutnya, *KF estimation* merupakan komponen yang digunakan DeepSORT untuk melakukan estimasi *state* objek yang telah dilacak sebelumnya. Terakhir, *track management* melakukan pengaturan pelacakan dengan memperbarui *state* objek setiap saat. Hal yang membedakan antara DeepSORT dan SORT adalah pada DeepSORT ditambahkan model CNN untuk ekstraksi fitur pada citra yang dibatasi oleh detektor. Untuk menutupi kekurangan algoritma SORT, algoritma DeepSORT menambahkan informasi tampilan fitur objek dan melakukan peminjaman model ReID untuk mengekstrak fitur tampilan, serta mengurangi permasalahan tertukarnya ID sebesar 45%.[6].

F. Pengujian Sistem

Pada penelitian, dilakukan pengujian untuk mengukur kinerja sistem yang dibuat. Pengujian dilakukan menggunakan metode pengujian *Confusion Matrix* dan *Black Box Testing*.

1) Confusion Matrix

Confusion Matrix adalah representasi dari pengklasifikasian kinerja sebuah model dengan format biner yang mewakili nilai performa model yang dibuat.[14] *Confusion Matrix* merupakan *array* persegi yang berdimensi $n \times n$, dengan n yang menyatakan jumlah atau banyak kelas. Evaluasi tingkat efektivitas suatu model klasifikasi dapat dihitung dengan menggunakan metrik *precision*, *recall*, *F1 score*, dan *accuracy*, berdasarkan pada *Confusion Matrix* dengan persamaan 1,2,3, dan 4 sebagai berikut:

$$Precision = \frac{True\ Positive}{Actual\ Results} = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{True\ Positive}{Predicted\ Results} = \frac{TP}{TP+FN} \quad (2)$$

$$F1\ Score = 2 \times \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (4)$$

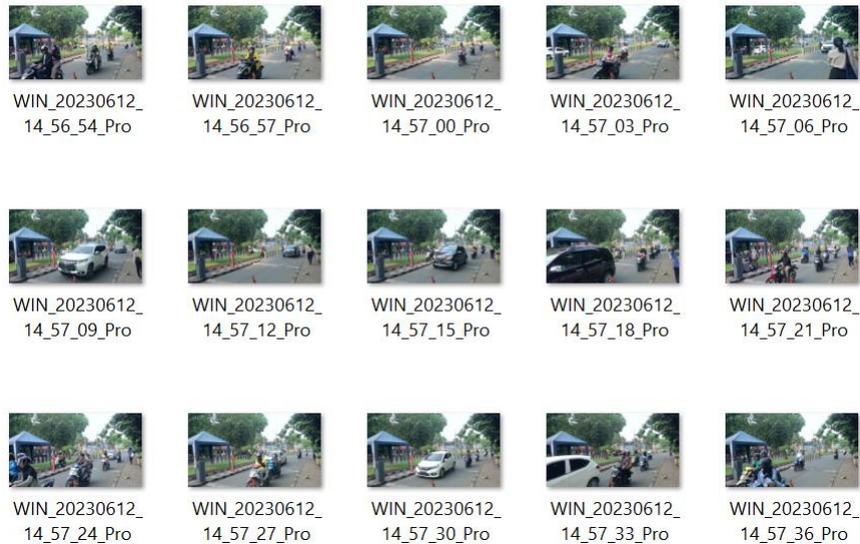
2) Black Box Testing

Black-box testing adalah pengujian sistem yang bertujuan untuk memeriksa apakah sebuah sistem atau program berjalan sesuai dengan kebutuhan yang ditentukan. Pengujian ini tidak membutuhkan pengetahuan mengenai cara kerja kode program (implementasi program). [15]

III. HASIL DAN PEMBAHASAN

A. Pengumpulan Dataset

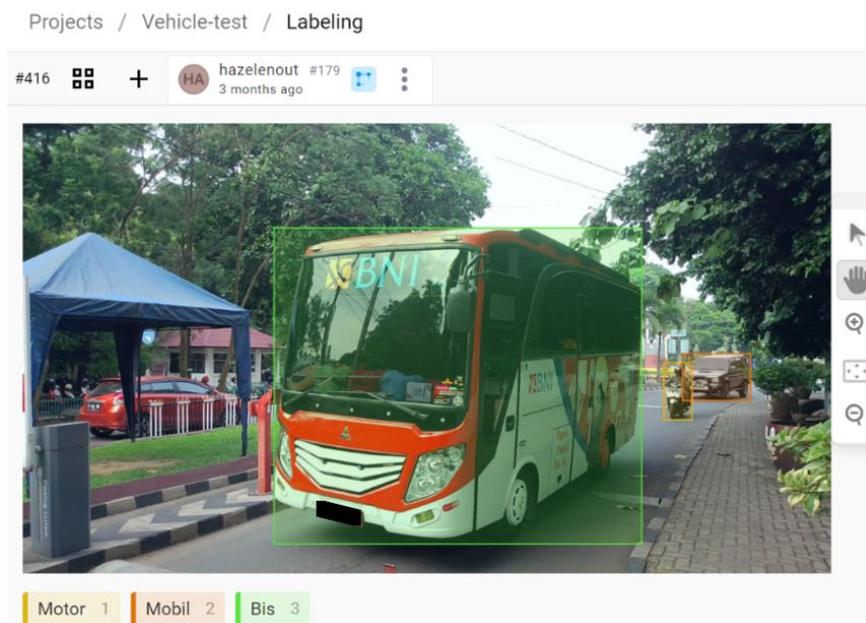
Dataset yang digunakan dalam penelitian ini merupakan kumpulan gambar sebanyak 199 gambar kendaraan dengan lokasi pengambilan gambar di gerbang masuk depan Universitas Lampung seperti pada gambar 4. Pengambilan gambar dilakukan menggunakan 3 alat yaitu *tripod* (2 meter), *webcam* dan *laptop*.



Gambar 4. Pengumpulan dataset kendaraan

B. Pelabelan Dataset

Labeling dalam penelitian ini dilakukan menggunakan Label Studio seperti pada gambar 5. Terdapat 3 pelabelan yang digunakan dalam proses, yaitu Motor (label kuning), Mobil (label orange), dan Bis (label hijau). Dari tahapan ini, didapatkan hasil pelabelan kelas motor sebanyak 213, kelas mobil sebanyak 84, dan bis sebanyak 33. Selama *labeling*, terdapat beberapa hal yang perlu diperhatikan saat melakukan anotasi untuk mendapatkan kualitas keakuratan yang baik dalam anotasi gambar. Hal-hal yang dilakukan untuk meningkatkan kualitas anotasi dalam proses ini adalah memastikan kotak pembatas (*bouding box*) sesuai dengan tepi objek, dan menghindari atau mengurangi kotak pembatas yang tumpang tindih satu sama lain (*overlapping bounding boxes*).

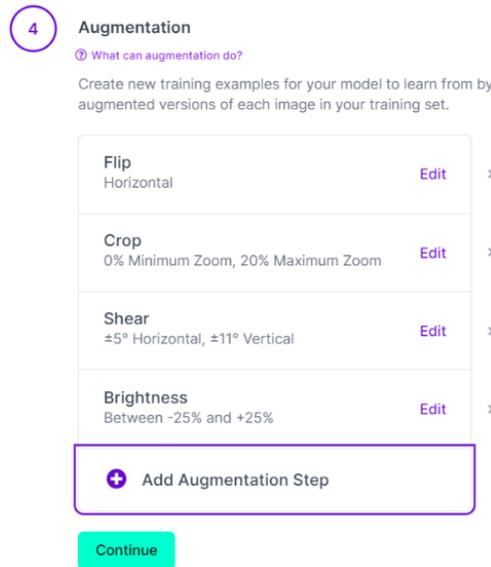


Gambar 5. Labeling menggunakan Label Studio

C. Augmentation Dataset

Pada penelitian ini, digunakan 4 *augmentation option* seperti pada gambar 6. Melalui tahapan yang telah dilakukan, didapatkan hasil *generate* gambar sebanyak 3 kali *multiple data training*, sehingga jumlah kelas pada data *training* bertambah. Jumlah kelas motor menjadi 414, mobil menjadi 171, dan bis menjadi 75. Terdapat total 473 gambar dengan rincian 411

gambar untuk *training*, 41 gambar untuk *validation*, dan 21 gambar untuk *testing*. Sebanyak 411 *dataset* dilakukan *training* menggunakan algoritma YOLOv5. Model YOLOv5 yang digunakan untuk melatih *dataset* adalah YOLOv5s.



Gambar 6. Augmentation pada Roboflow

D. Training Dataset

Training dataset dilakukan pada Google Colab dengan membagi *dataset* menjadi 16 *batch*. Iterasi *training* dilakukan sebanyak 1000 *epochs* menggunakan konfigurasi *yolov5s.yaml*, yang merupakan file penting yang mengatur berbagai aspek model YOLOv5s dengan model *backbone* yang digunakan adalah 's' atau *small*. Gambar 7 merupakan proses *training* yang telah dilakukan sebanyak 1000 *epochs*. Dari *training*, didapatkan *weight* yang diberi nama 'best.pt' yang digunakan sebagai bobot pada sistem perhitungan.

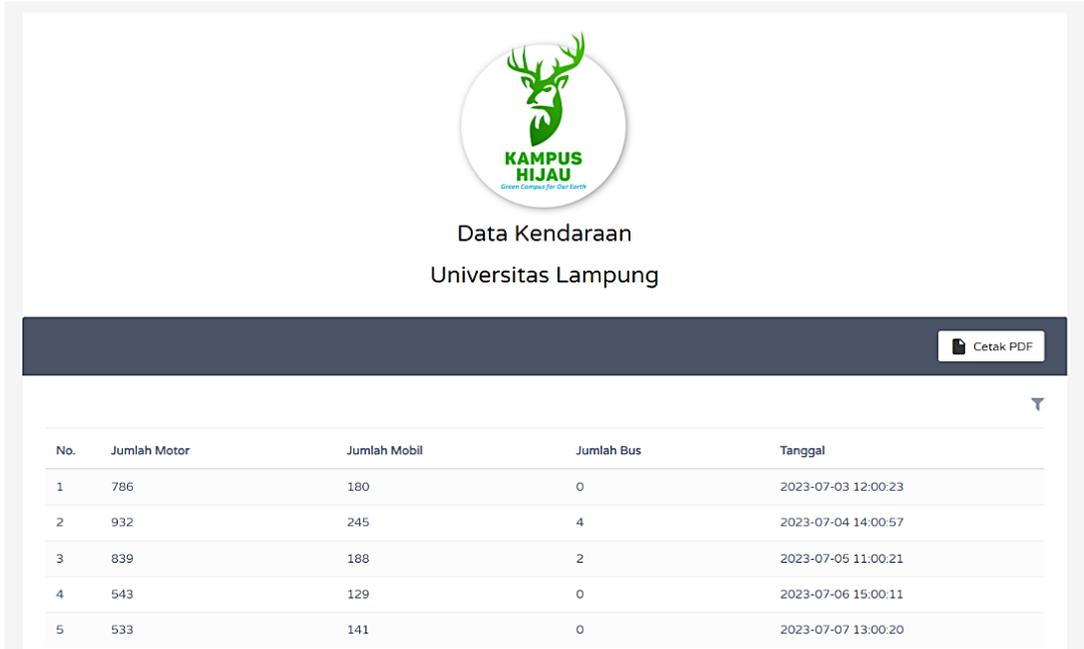
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
995/999	1.85G	0.01093	0.005325	0.0001504	28	416: 100% 26/26 [00:05<00:00, 5.13it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 2/2 [00:00<00:00, 5.26it/s]
	all	41	70	0.946	0.945	0.966 0.811
996/999	1.85G	0.01089	0.005501	0.0002162	26	416: 100% 26/26 [00:03<00:00, 6.57it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 2/2 [00:00<00:00, 5.21it/s]
	all	41	70	0.946	0.946	0.966 0.811
997/999	1.85G	0.01077	0.00539	0.0001908	18	416: 100% 26/26 [00:04<00:00, 5.90it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 2/2 [00:00<00:00, 3.72it/s]
	all	41	70	0.946	0.946	0.966 0.811
998/999	1.85G	0.01048	0.005647	0.00024	34	416: 100% 26/26 [00:04<00:00, 5.57it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 2/2 [00:00<00:00, 5.71it/s]
	all	41	70	0.948	0.945	0.966 0.811
999/999	1.85G	0.01032	0.005131	0.0002309	30	416: 100% 26/26 [00:04<00:00, 6.49it/s]
	Class	Images	Instances	P	R	mAP50 mAP50-95: 100% 2/2 [00:00<00:00, 6.18it/s]
	all	41	70	0.939	0.947	0.966 0.813

Gambar 7. Proses *training* sebanyak 1000 *epochs*

E. Pengujian Sistem

Setelah *training dataset* selesai dilakukan, tahapan selanjutnya menambahkan algoritma DeepSORT dan perhitungan pada sistem. Selain itu, dibuat halaman *report* berbasis *web* menggunakan *framework* Django. Data hasil perhitungan yang

dihasilkan oleh sistem akan ditampilkan melalui halaman ini dalam bentuk penyajian tabel. Adapun tampilan halaman *report* sebagaimana terdapat pada gambar 8.



No.	Jumlah Motor	Jumlah Mobil	Jumlah Bus	Tanggal
1	786	180	0	2023-07-03 12:00:23
2	932	245	4	2023-07-04 14:00:57
3	839	188	2	2023-07-05 11:00:21
4	543	129	0	2023-07-06 15:00:11
5	533	141	0	2023-07-07 13:00:20

Gambar 8. Tampilan halaman *report*

Halaman *report* menampilkan informasi yang disajikan dalam bentuk tabel untuk memudahkan pengguna dalam membaca data hasil perhitungan. Tabel terdiri atas kolom 'No' yang menunjukkan nomor. Kemudian, terdapat kolom 'Jumlah Motor', 'Jumlah Mobil', 'Jumlah Bus', yang menunjukkan banyak motor, mobil dan bus yang terdeteksi dan dihitung oleh sistem. Terakhir, kolom 'Tanggal', yang menunjukkan tanggal dan waktu pendeteksian. Terdapat tombol 'Cetak PDF' yang apabila diklik akan mencetak halaman *report* dalam bentuk PDF.

F. Pengujian Sistem Secara Realtime Menggunakan Confusion Matrix

Pada tahap ini, pembuatan sistem telah selesai. Sistem sudah dapat melakukan perhitungan kendaraan secara *realtime* seperti pada gambar 9, dan sudah dapat menampilkan hasil perhitungan pada *web* halaman *report*.



Gambar 9. Sistem perhitungan yang telah selesai dibuat

Pada penelitian ini, pengujian sistem dilakukan secara *realtime* selama 5 hari kerja (Senin, Selasa, Rabu, Kamis, dan Jumat) seperti pada gambar 10. Waktu pengambilan *sample* memiliki durasi 1 jam setiap harinya dan dilakukan secara acak tiap harinya dalam rentang waktu jam 10.00 WIB hingga 15.00 WIB. Selama pengujian, terdapat dua hasil perhitungan yaitu perhitungan yang dilakukan oleh sistem dan perhitungan yang dilakukan manual.



Gambar 10. *Sample* hasil pengujian *realtime*

1) Hasil pengujian *realtime* pada hari Senin

Pengujian dilakukan pada hari Senin tanggal 3 Juli 2023, pukul 11.00 WIB hingga 12.00 WIB. Data hasil pengujian pada hari senin terdapat dalam tabel 1 dan tabel 2. Berdasarkan hasil pengujian pada tabel 1, sistem mendeteksi dan menghitung 786 motor sedangkan saat perhitungan manual didapatkan sebanyak 792 motor. Kemudian pada mobil, didapatkan selisih 1 mobil dengan sistem sebanyak 180 dan manual sebanyak 181. Tabel 2 menunjukkan *confusion matrix* hasil pengujian, dengan baris *predict* menunjukkan hasil prediksi dari target objek, dan kolom *actual* yang menunjukkan hasil aktual dari target objek. Terdapat 0 bis yang terprediksi dan pada kenyataannya 0 bis. Lalu terprediksi 180 mobil yang kenyataannya terdapat 181 mobil, akan tetapi sistem tidak mengenali 1 objek tersebut (*Background FN*). Terakhir, terprediksi 783 motor dan pada kenyataannya terdapat 792 motor, akan tetapi sistem tidak dapat mengenali 9 objek motor lainnya.

TABEL 1
 DATA PERBANDINGAN HASIL PERHITUNGAN PENGUJIAN PADA HARI SENIN

Hasil Perhitungan	Motor	Mobil	Bis
Sistem	786	180	0
Manual	792	181	0

TABEL 2
 CONFUSION MATRIX PENGUJIAN PADA HARI SENIN

		Actual			
		Bis	Mobil	Motor	Background FP
Predict	Bis	0			
	Mobil		180		
	Motor			783	3
	Background FN		1	9	

Berdasarkan data pada tabel 2, dilakukan perhitungan nilai *precision*, *recall*, *F1 score*, dan *accuracy* menggunakan persamaan 1, 2, 3, dan 4 dengan hasil seperti pada tabel 3.

TABEL 3
PRECISION, RECALL, F1 SCORE, DAN ACCURACY PADA HARI SENIN

	Precision	Recall	F1 Score	Accuracy
Nilai	99%	98%	98%	98%

2) Hasil pengujian *realtime* pada hari Selasa

Pada hari Selasa, 4 Juli 2023 dilakukan pengujian mulai pukul 13.00 WIB hingga 14.00 WIB. Data hasil pengujian pada hari selasa terdapat dalam tabel 4 dan tabel 5. Tabel 4 menunjukkan perbedaan hasil perhitungan yang dilakukan oleh sistem dengan hasil perhitungan manual. Dari perhitungan, sistem berhasil menghitung motor, mobil dan bis sebanyak 932 motor, 245 mobil, dan 2 bis. Sedangkan untuk perhitungan yang dilakukan manual (aktual) didapatkan 957 motor, 248 mobil dan 4 bis. Tabel 5 menyajikan *confusion matrix*, terprediksi 2 bis yang pada kenyataannya terdapat 4 bis, akan tetapi sistem tidak mengenali 2 objek lainnya sebagai salah satu dari label target. Lalu, terprediksi 245 mobil yang pada kenyataannya terdapat 248 mobil. Dan terakhir, terprediksi 932 motor dengan 1 objeknya dianggap motor yang sebenarnya bukan motor (*Background FP*), dengan hasil sebenarnya adalah sebanyak 957 motor dan sistem tidak dapat mengenali 26 objek motor.

TABEL 4
DATA PERBANDINGAN HASIL PERHITUNGAN PENGUJIAN PADA HARI SELASA

Hasil Perhitungan	Motor	Mobil	Bis
Sistem	932	245	2
Manual	957	248	4

TABEL 5
CONFUSION MATRIX PENGUJIAN PADA HARI SELASA

		Actual			
		Bis	Mobil	Motor	Background FP
Predict	Bis	2			
	Mobil		245		
	Motor			931	1
	Background FN	2	3	26	

Menurut data pada tabel 5, dilakukan perhitungan nilai *precision*, *recall*, *F1 score*, dan *accuracy* dengan hasil perhitungan seperti pada tabel 6.

TABEL 6
PRECISION, RECALL, F1 SCORE, DAN ACCURACY PADA HARI SELASA

	Precision	Recall	F1 Score	Accuracy
Nilai	99%	97%	97%	97%

3) Hasil pengujian *realtime* pada hari Rabu

Pengujian dilakukan pada hari Rabu tanggal 5 Juli 2023, pukul 10.00 WIB. Data hasil pengujian pada hari Rabu terdapat dalam tabel 7 dan tabel 8. Tabel 7 menyajikan data hasil perhitungan yang dilakukan oleh sistem dengan motor sebanyak 839, mobil sebanyak 188 dan bis sebanyak 2. Untuk hasil perhitungan manual didapatkan motor sejumlah 861, mobil sejumlah 189 dan bis sejumlah 0. Tabel 8 menampilkan data *confusion matrix*, dengan prediksi bis sebanyak 2, dengan aktualnya sebanyak 0 dikarenakan sistem menganggap objek lain sebagai bis. Lalu, terprediksi 188 mobil, dengan aktualnya sebanyak 189. Sistem salah mengenali 1 objek lain sebagai mobil dan tidak dapat mengenali 2 objek mobil sebagai mobil. Untuk motor, terprediksi sebanyak 839 motor yang pada nyatanya terdapat 861 motor, akan tetapi sistem salah mengenali 1 objek lain sebagai motor, dan tidak dapat mengenali 23 objek motor.

TABEL 7
DATA PERBANDINGAN HASIL PERHITUNGAN PENGUJIAN PADA HARI RABU

Hasil Perhitungan	Motor	Mobil	Bis
Sistem	839	188	2
Manual	861	189	0

TABEL 8
CONFUSION MATRIX PENGUJIAN PADA HARI RABU

		<i>Actual</i>			
		Bis	Mobil	Motor	Background FP
<i>Predict</i>	Bis	0			2
	Mobil		187		1
	Motor			838	1
	Background FN		2	23	

Menggunakan data pada tabel 8, dilakukan perhitungan nilai *precision*, *recall*, *F1 score*, dan *accuracy* dengan hasil perhitungan yang terdapat pada tabel 9.

TABEL 9
PRECISION, RECALL, F1 SCORE, DAN ACCURACY PADA HARI RABU

	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>	<i>Accuracy</i>
Nilai	99%	97%	97%	97%

4) Hasil pengujian *realtime* pada hari Kamis

Pada hari Kamis, 6 Juli 2023 dilakukan pengujian yang berlangsung dari pukul 14.00 WIB hingga 15.00 WIB. Data hasil pengujian pada hari Kamis terdapat dalam tabel 10 dan tabel 11. Berdasarkan tabel 10, didapatkan hasil perhitungan sistem sebanyak 543 untuk motor, 129 untuk mobil dan 1 untuk bis. Sedangkan perhitungan manual didapatkan 563 motor, 133 mobil dan 0 bis. Tabel 11 merupakan data confusion matrix yang menunjukkan bahwa bis terprediksi 0, dengan nilai aktualnya adalah 0. Selanjutnya, terdapat mobil terprediksi sebanyak 129. Dengan nilai aktualnya sebesar 133, dikarenakan sistem tidak dapat mengenali 4 objek mobil. Dan, terprediksi motor sebanyak 543 dengan nilai aktualnya sebanyak 563. *Background FN* pada baris *predict* menunjukkan bahwa sistem tidak dapat mengenali 20 objek motor sebagai motor.

TABEL 10
DATA PERBANDINGAN HASIL PERHITUNGAN PENGUJIAN PADA HARI KAMIS

Hasil Perhitungan	Motor	Mobil	Bis
Sistem	543	129	1
Manual	563	133	0

TABEL 11
CONFUSION MATRIX PENGUJIAN PADA HARI KAMIS

		<i>Actual</i>			
		Bis	Mobil	Motor	Background FP
<i>Predict</i>	Bis	0			1
	Mobil		129		
	Motor			543	
	Background FN		4	20	

Mengacu data pada tabel 11, dilakukan perhitungan nilai *precision*, *recall*, *F1 score*, dan *accuracy* dengan hasil perhitungan yang terdapat pada tabel 12.

TABEL 12
PRECISION, RECALL, F1 SCORE, DAN ACCURACY PADA HARI KAMIS

	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>	<i>Accuracy</i>
Nilai	99%	96%	96%	96%

5) Hasil pengujian *realtime* pada hari Jumat

Pengujian dilaksanakan pada hari Jumat tanggal 7 Juli 2023, pukul 12.00 WIB hingga 13.00 WIB. Data hasil pengujian pada hari Kamis terdapat dalam tabel 13 dan tabel 14. Melalui tabel 13, dapat dilihat bahwa hasil perhitungan yang didapatkan oleh sistem untuk motor sebanyak 533, mobil sebanyak 141, dan bis sebanyak 0. Sedangkan pada perhitungan manual, didapatkan motor sebanyak 548, mobil sebanyak 145, dan bis sebanyak 0. Lalu, pada tabel 14 disajikan data confusion matrix yang menjelaskan lebih rinci tabel 13. Berdasarkan tabel 14, terprediksi bis sebanyak 1 dengan nilai aktualnya adalah 0. Pada bis, sistem mengenali 1 objek lain sebagai bis. Selanjutnya, terprediksi 141 mobil dengan nilai aktualnya sebanyak 145 mobil.

Terdapat 4 objek mobil yang tidak dapat dikenali oleh sistem. Terakhir, motor diprediksi sebanyak 533 dengan nilai aktualnya sebanyak 548. Terdapat 15 objek motor yang tidak dapat dikenali oleh sistem, dan 1 objek asing yang diprediksi sebagai motor.

TABEL 13
DATA PERBANDINGAN HASIL PERHITUNGAN PENGUJIAN PADA HARI JUMAT

Hasil Perhitungan	Motor	Mobil	Bis
Sistem	533	141	1
Manual	548	145	0

TABEL 14
CONFUSION MATRIX PENGUJIAN PADA HARI JUMAT

		Actual			
		Bis	Mobil	Motor	Background FP
Predict	Bis	0			1
	Mobil		141		
	Motor			532	1
	Background FN		4	15	

Mengacu data pada tabel 14, dilakukan perhitungan nilai *precision*, *recall*, *F1 score*, dan *accuracy* dengan hasil perhitungan yang terdapat pada tabel 15.

TABEL 15
PRECISION, RECALL, F1 SCORE, DAN ACCURACY PADA HARI JUMAT

	Precision	Recall	F1 Score	Accuracy
Nilai	99%	97%	97%	96%

Berdasarkan pengujian secara *realtime* selama 5 hari, didapatkan nilai rata-rata *precision*, *recall*, *F1 score*, dan *accuracy* sistem yang telah dihitung menggunakan *confusion matrix* pada tabel 16.

TABEL 16
DATA HASIL PERHITUNGAN RATA-RATA CONFUSION MATRIX PENGUJIAN SECARA REALTIME

	Precision	Recall	F1 Score	Accuracy
Nilai	99%	97%	97,2%	96,8%

Setelah mendapatkan data hasil perhitungan rata-rata *confusion matrix* pengujian secara *realtime*, selanjutnya adalah menghitung rata-rata selisih perhitungan antara sistem dengan manual yang didapatkan saat pengujian *realtime* (rata-rata persentase kesalahan), sehingga nilai rata-rata persentase kesalahan sistem dalam melakukan perhitungan terdapat pada tabel 17.

TABEL 17
DATA PERSENTASE KESALAHAN PERHITUNGAN SISTEM PADA SELURUH KELAS

Seluruh kelas	
Rata-rata persentase kesalahan	2.35%

Tabel 17 menunjukkan rata-rata persentase kesalahan pada perhitungan yang dilakukan dari sistem sebesar 2.35%, sehingga ketepatan sistem dalam melakukan perhitungan secara *realtime* sebesar 97.65%. Hasil yang didapatkan sudah sangat baik apabila merujuk pada penelitian terdahulu dengan judul “*Vehicle counting based on vehicle detection and tracking from aerial videos*” yang dilakukan pada tahun 2018 [16]. Dalam penelitian tersebut didapatkan ketepatan perhitungan diatas 90% pada pengujian video stasioner.

G. Pengujian Sistem Secara Realtime Menggunakan Black Box Testing

Sistem perhitungan yang dibuat telah diselesaikan. Setelah dilakukan *testing* menggunakan *confusion matrix* yang difokuskan untuk mengukur performa sistem dalam melakukan pendeteksian dan perhitungan kendaraan, maka pada tahapan ini dilakukan *black box testing* untuk mengukur kinerja keseluruhan sistem secara fungsional. *Black box testing* dilakukan menggunakan skenario uji yang telah ditentukan dengan hasil yang terdapat pada tabel 18.

TABEL 18
SKENARIO UJI *BLACK BOX TESTING*

No	Skenario Uji	Hasil yang Diharapkan	Hasil Pengamatan	Kesimpulan
1	Sistem memiliki nilai akurasi >70%	Nilai akurasi sistem 70%	Sistem memiliki akurasi sebesar 96,8%	Sesuai
2	Sistem dapat mendeteksi dan membedakan 3 jenis kendaraan (motor, mobil, bis) secara tepat.	Sistem dapat mendeteksi dan membedakan 3 jenis kendaraan (motor, mobil, bis) secara tepat	Sistem dapat mengenali 3 jenis kendaraan secara tepat.	Sesuai
3	Sistem dapat menghitung tiap jenis kendaraan yang terdeteksi.	Sistem dapat menghitung tiap jenis kendaraan yang terdeteksi.	Sistem dapat menghitung masing-masing kendaraan yang terdeteksi secara tepat.	Sesuai
4	Sistem dapat menyimpan hasil perhitungan tiap jenis kendaraan yang terdeteksi.	Sistem dapat menyimpan hasil perhitungan tiap jenis kendaraan yang terdeteksi.	Sistem menyimpan hasil perhitungan pada <i>database</i> .	Sesuai
5	Sistem dapat menampilkan data hasil perhitungan tiap jenis kendaraan yang terdeteksi pada halaman <i>report</i> .	Sistem dapat menampilkan data hasil perhitungan tiap jenis kendaraan yang terdeteksi pada halaman <i>report</i> .	Sistem dapat menampilkan data melalui halaman <i>report</i> secara tepat.	Sesuai

IV. SIMPULAN

Sistem perhitungan kendaraan menggunakan *object tracking* berhasil melacak dan menghitung tiga jenis kendaraan yaitu motor, mobil, dan bis dengan algoritma YOLOv5 dan DeepSORT. Hasil perhitungan yang didapat berhasil disimpan ke dalam *database* dan berhasil ditampilkan melalui *web* halaman *report*. Berdasarkan hasil pengujian yang dilakukan secara *realtime*, didapatkan nilai rata-rata *precision* sebesar 99%, *recall* sebesar 97%, *F1 Score* sebesar 97.2%, *accuracy* sebesar 96.8%, dan persentase ketepatan sistem dalam melakukan perhitungan sebesar 97.65%. Sehingga sistem memiliki performa yang cukup baik dalam mendeteksi kendaraan dan menghitung kendaraan. Pada penelitian ini, terdapat *dataset* antar kelas masih belum seimbang, dan hal ini mempengaruhi performa model dengan mengacu pada penelitian "*On Learning from Natural Language to Develop Intelligent System to Identify Potential Medical Condition*" [17]. *Dataset* kelas motor jauh lebih banyak dibandingkan *dataset* mobil dan bis. Oleh karena itu, pada penelitian selanjutnya perlu dilakukan penambahan *dataset* mobil dan bis untuk mendapatkan performa sistem yang lebih baik. Pada penelitian ini belum sampai pada tahap membedakan antara jenis kendaraan listrik dan kendaraan bensin/diesel. Sehingga, perlu dilakukan penelitian khusus yang berfokus dalam membedakan antara kendaraan listrik dan kendaraan bensin/diesel, seperti memperhatikan hal pembeda kedua jenis kendaraan dari sisi bentuk maupun desain.

UCAPAN TERIMA KASIH

Ucapan terima kasih kepada para dosen Universitas Lampung yang telah memberikan arahan dan saran selama proses penelitian ini. Dalam penulisan jurnal ilmiah ini, terdapat banyak pihak yang telah memberikan dorongan dalam bentuk moral maupun material sehingga penulisan jurnal ini dapat berjalan dengan baik.

DAFTAR PUSTAKA

- [1] M. Fadli Ramadan, "Infografis Jumlah Kendaraan Listrik di Indonesia." Sep. 2023. Accessed: Feb. 28, 2024. [Online]. Available: <https://www.inews.id/multimedia/infografis/infografis-jumlah-kendaraan-listrik-di-indonesia-baru-68000-ac>
- [2] Universitas Indonesia, "UI GreenMetric," pp. 1–13, 2019, [Online]. Available: <https://greenmetric.ui.ac.id/>
- [3] M. Utomo, "Kampus Hijau Universitas Lampung* *," in *Lokakarya Kampus Hijau dalam rangka Dies Natalis Unila ke-42*, 2023. [Online]. Available: <https://greenmetric.unila.ac.id/>
- [4] J. Brownlee, "Deep Learning for Computer Vision Image Classification , Object Detection , and Face Recognition in Python," *Deep Learn. Comput. Vis.*, p. 532, 2019.
- [5] K. C. Kirana, "Pengolahan Citra Digital: Teori dan Penerapan Pengolahan Citra Digital pada Deteksi Wajah." Ahlimedia Book, Malang, p. 68,

- 2021.
- [6] P. Spagnolo, P. Luigi Mazzeo, and S. Ramakrishnan, *Visual Object Tracking with Deep Neural Networks*. London: IntechOpen, 2019.
 - [7] Y. Park, L. M. Dang, S. Lee, D. Han, and H. Moon, "Multiple object tracking in deep learning approaches: A survey," *Electronics (Switzerland)*, vol. 10, no. 19. MDPI, Basel, 2021.
 - [8] Mardiana, M. A. Muhammad, and Y. Mulyani, "Library Attendance System using YOLOv5 Faces Recognition," in *Proceedings - ICCTEIE 2021: 2021 International Conference on Converging Technology in Electrical and Information Engineering: Converging Technology for Sustainable Society*, 2021, pp. 68–72.
 - [9] F. Yang, X. Zhang, and B. Liu, "Video object tracking based on YOLOv7 and DeepSORT," *J. Latex Cl. Files*, vol. 14, no. 8, p. 2, 2022.
 - [10] H. Chen *et al.*, "MC-YOLOv5: A Multi-Class Small Object Detection Algorithm," *Biomimetics*, vol. 8, no. 4, 2023.
 - [11] G. Verheyen, *Scrum - a Pocket Guide - 3rd Edition*, Third. Hertogenbosch: Van Haren Publishing, 2021.
 - [12] G. Jocher, "YOLO: A Brief History." 2023. [Online]. Available: <https://docs.ultralytics.com/#where-to-start>
 - [13] R. Pereira, G. Carvalho, L. Garrote, and U. J. Nunes, "Sort and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation with New Data Association Metrics," *Appl. Sci.*, vol. 12, no. 3, 2022.
 - [14] L. Sharma and M. Carpenter, *Computer Vision and Internet of Things*, First. Boca Raton: Chapman and Hall/CRC, 2022. [Online]. Available: <https://www.taylorfrancis.com/books/9781003244165>
 - [15] R. Bierig, S. Brown, E. Galván, and J. Timoney, *Essentials of Software Testing*. Cambridge: Cambridge University Press, 2021. [Online]. Available: <https://www.cambridge.org/highereducation/product/9781108974073/book>
 - [16] X. Xiang, M. Zhai, N. Lv, and A. El Saddik, "Vehicle counting based on vehicle detection and tracking from aerial videos," *Sensors (Switzerland)*, vol. 18, no. 8, 2018.
 - [17] N. Bhoj and A. Tripathi, "On Learning from Natural Language to Develop Intelligent System to Identify Potential Medical Condition." 2021. [Online]. Available: <https://www.researchsquare.com/article/rs-953323/v1>