



Sistem Deteksi Rambu Lalu Lintas Berbasis *You Only Look Once* (Yolov8) Menggunakan Raspberry Pi

Traffic Signs Detection System Based on You Only Look Once (Yolov8) using Raspberry Pi

Mawaring Wening¹⁾, Jans Hendry¹⁾, Ardhi Wicaksono Santoso²⁾

¹ Teknologi Rekayasa Instrumentasi dan Kontrol, Sekolah Vokasi, Universitas Gadjah Mada

² Teknologi Rekayasa Internet, Sekolah Vokasi, Universitas Gadjah Mada

¹ Bulaksumur, Daerah Istimewa Yogyakarta, Indonesia

² Bulaksumur, Daerah Istimewa Yogyakarta, Indonesia

jans.hendry@ugm.ac.id

Diterima: 31 Desember 2024 || Direvisi: 1 Februari 2025 || Disetujui: 15 Maret 2025

Abstrak – Kendaraan *autonomous vehicle* (AV) direncanakan mewarnai alat transportasi darat di Ibu Kota Nusantara (IKN). Hal ini membuka kemungkinan penelitian–penelitian terkait sebelum implementasi benar–benar dilakukan. Salah satu kemampuan yang dimiliki oleh AV adalah mengenali jenis dari rambu lalu lintas. Untuk itu, penelitian ini bertujuan untuk merancang sistem deteksi rambu lalu lintas sebagai wawasan dan dukungan penerapan *autonomous vehicle* di IKN Nusantara. Untuk itu, sebanyak 11.157 citra yang berisi 30 jenis rambu lalu lintas sebagai dataset primer telah diambil di sepanjang jalan Daerah Istimewa Yogyakarta. Ditambahkan juga variasi dataset berupa penambahan *noise*, *blur*, dan *dark*. Pada *training* model dilakukan konfigurasi *hyper-parameter* berupa *learning rate*, *epoch*, dan ukuran citra. Penelitian ini menggunakan algoritma *You Only Look Once* v8. Pengujian menggunakan data siang menghasilkan *precision* 96%, *recall* 83%, dan *accuracy* 80%. Sedangkan, pengujian menggunakan data malam menghasilkan *precision* 93%, *recall* 70%, dan *accuracy* 67%. Pengujian ini efektif untuk kendaraan bergerak dengan kecepatan di bawah 40 km/jam karena keterbatasan kecepatan komputasi perangkat keras Raspberry Pi.

Kata Kunci: rambu lalu lintas, Yolov8, IKN, *hyper-parameter*, Raspberry Pi

Abstract – *Autonomous vehicle (AV) is projected to become a part of land transportation in the New Capital City of Nusantara (IKN). . This opens up opportunities for related research to be conducted prior to actual implementation.. One of the capabilities of AV is recognizing different types of traffic signs. Therefore, this study aims to design a traffic sign detection system as an insight and support for the implementation of autonomous vehicles in IKN Nusantara. To achieve this, a total of 11,157 images containing 30 types of traffic signs were collected as the primary dataset along the roads of the Special Region of Yogyakarta. Variations of the dataset were also added in the form of noise, blur, and dark. During the model training, hyper-parameter configurations such as learning rate, epoch, and image size were performed. In this study, the You Only Look Once v8 method is used. The results of testing with daytime data showed an accuracy of 80%, recall of 83%, and precision of 96%. In contrast, tests with night data showed 93% precision, 70% recall, and 67% accuracy. This test works well for cars moving at speeds below than 40 km/h because of the Raspberry Pi hardware's computing speed constraints.*

Keywords: *traffic signs, Yolov8, IKN, hyper-parameter, Raspberry Pi*

PENDAHULUAN

Saat ini, *Artificial Intelligent* (AI) berkembang pesat di berbagai bidang, salah satunya bidang otomotif yaitu *Autonomous Vehicle* (AV). Indonesia merupakan salah satu negara yang sedang mengembangkan teknologi AV dari banyak negara lainnya. Di Indonesia, teknologi ini diperkirakan menjadi tulang punggung

transportasi di Ibu Kota Negara (IKN) Nusantara sebagaimana yang disampaikan oleh Kepala Otorita IKN Bambang Susantono bahwa teknologi AV menjadi salah satu elemen utama pendukung sistem transportasi maju sebagai bagian dari teknologi *smart city* di IKN (Saputra, 2022). Terdapat beberapa tujuan dari AV yaitu menjadikan mobilitas perkotaan yang

lebih aman dan efisien, menghilangkan kemacetan dan emisi Gas Rumah Kaca (GRK), dan menyediakan fasilitas untuk penyandang disabilitas dan berkebutuhan khusus (Saputra, 2022). Sementara itu, Kepala Badan Riset dan Inovasi Nasional (BRIN) menyampaikan bahwa pengembangan AV pada tahun 2020–2024 difokuskan pada kendaraan listrik (*Electric Vehicle*). Dalam kurun waktu tersebut, dicanangkan sebagai masa penguasaan teknologi yang meliputi sistem deteksi objek/sensor, *computer vision*, sistem telekomunikasi, dan interaksi manusia dan kendaraan (*human to vehicle interaction*) (BRIN, 2022). Melihat hal tersebut, peneliti terdorong untuk melakukan penelitian yang didasarkan pada sistem deteksi dan *computer vision*, karena kedua teknologi tersebut merupakan bagian penting dari sistem AV khususnya untuk navigasi dalam pergerakan mobil.

Penting untuk diperhatikan bahwa keamanan dan keselamatan berkendara merupakan hal yang harus diutamakan. Sistem pada *Autonomous Vehicle* harus memiliki kemampuan dalam mengenali berbagai jenis rambu lalu lintas untuk mengurangi kecelakaan lalu lintas yang disebabkan oleh kesalahan manusia atau pengemudi yang tidak mematuhi rambu lalu lintas. Angka kecelakaan yang terjadi di Indonesia cukup tinggi sebagaimana data yang disampaikan oleh Korps Lalu Lintas (Korlantas) Polri bahwa pada Januari sampai 13 September 2022 terjadi 94.617 kasus kecelakaan lalu lintas. Jumlah tersebut meningkat 34,6 persen dibandingkan tahun 2021 dengan periode yang sama yaitu 70 ribu kasus kecelakaan (Saptohutomo, 2022). Salah satu penyebab kecelakaan lalu lintas adalah karena perilaku melanggar lalu lintas. Pelanggaran lalu lintas terjadi karena kurangnya kesadaran akan kepatuhan dan pemahaman terhadap aturan lalu lintas. Menurut data dari Pusat Informasi Kriminal Nasional (Pusiknas) Polri, pada periode I-2022 telah terjadi (Rafly Adli, 2023) kasus pelanggaran lalu lintas. Faktor penyebab terjadinya kasus pelanggaran lalu lintas sangat beragam, salah satunya melanggar rambu lalu lintas. Oleh sebab itu, penting pada sistem *Autonomous Vehicle* memiliki kemampuan dalam mengenali berbagai jenis rambu lalu lintas. Melihat fenomena tersebut, peneliti tertarik untuk mengembangkan sistem untuk mengidentifikasi objek berupa rambu lalu lintas.

(Lavanya & Pande, 2023) Terdapat dua macam algoritma yang bisa digunakan untuk mendeteksi objek pada pengelolaan citra yaitu berdasarkan *region proposal* dan *Single Shot Detector* (SSD). R-CNN, Fast

Tabel 1 Kelas rambu lalu lintas

Rambu Lalu Lintas	Keterangan	Rambu Lalu Lintas	Keterangan
	Larangan Kecepatan >40 Km/Jam		Dilarang Berjalan Terus
	Larangan Kecepatan >30 Km/Jam		Dilarang Berhenti
	Dilarang Putar Balik		Dilarang Mendahului
	Dilarang Parkir		Dilarang Belok Kanan
	Dilarang Masuk		Larangan Kendaraan MST > 10 Ton Putar Balik
	Perintah Pilih Satu Jalur		Pemberhentian Bus
	Perintah Ikuti Bundaran		Tempat Parkir
	Jalur Penyebrangan		SPBU
	Perintah Lajur Kiri		Masjid
	Perintah Jalur Sepeda		Balai Pertolongan Pertama
	Gereja		Rumah Sakit
	Rumah Sakit		Banyak Anak-Anak
	Persimpangan 3 sisi kiri prioritas		Persimpangan 3 sisi kanan prioritas
	Lampu Lalu Lintas		Persimpangan 3 Prioritas
	Persimpangan Empat		Hati-Hati

R-CNN dan Faster R-CNN merupakan jenis algoritma *region proposal*. Sedangkan, jenis algoritma *Single Shot Detector* adalah *You Only Look Once* (YOLO). Algoritma *Single Shot Detector* komputasinya lebih cepat dibandingkan *region proposal* (Huang et al.,

2017). Algoritma YOLO merupakan salah satu model *deep learning* yang digunakan untuk mendeteksi objek secara *real time* dengan komputasi deteksi yang cepat sehingga cocok untuk AV. Selain itu, algoritma YOLO memiliki kemampuan mendeteksi dan mengklasifikasikan objek dalam satu tahap atau *One Stage Detection* sehingga lebih efisien dan cepat. Terdapat berbagai versi YOLO mulai dari YOLOv2, YOLOv3 (Auliya et al., 2022), YOLOv4 (Wang et al., 2022), YOLOv5 (Nacir et al., 2022), YOLOv6, YOLOv7 (Yung et al., 2022), dan YOLOv8. YOLOv8 merupakan versi terbaru dari algoritma YOLO. Karena masih terbilang baru jadi belum banyak jurnal penelitian yang membahas YOLOv8 bahkan menurut (Lavanya & Pande, 2023) hanya terdapat 8 penelitian bereputasi yang telah menggunakannya hingga tahun 2021. Meskipun hampir 3 tahun berselang, namun *trend* hasil *review* penelitian tersebut mengindikasikan kecenderungan penambahan yang tidak terlalu signifikan, khususnya untuk deteksi rambu lalu lintas.

Oleh karena itu, pada penelitian ini akan dibuat sistem untuk mengidentifikasi rambu lalu lintas di Indonesia dengan metode YOLO menggunakan Rasperry pi 4 Model B. Pada penelitian ini menggunakan dataset dan *hyper-parameter* yang bervariasi untuk mendapatkan performa sistem yang lebih baik. Variasi pada dataset berupa penambahan derau (*noise*), *blur*, dan kecerahan cahaya dikurangi (*dark*). Sedangkan variasi pada *hyper-parameter* berupa variasi nilai *learning rate*, *epoch*, dan ukuran citra.

METODOLOGI PENELITIAN

Penelitian ini terdiri dari beberapa tahapan yaitu pengumpulan, pengolahan, dan *training* dataset untuk menghasilkan model yang akan digunakan sebagai *detector*. Sistem yang dibangun akan memiliki 3 fitur yaitu kemampuan mendeteksi rambu lalu lintas pada citra (*still image*), video, dan *real time* menggunakan webcam. Selama proses deteksi, hasilnya dapat diamati melalui LCD. Untuk pengukuran akurasi, model yang telah didapat diuji menggunakan *confusion matrix*.

Rambu Lalu Lintas

Pada penelitian ini bahan utama yang digunakan adalah tiga puluh jenis rambu lalu lintas berbeda yang terbagi dalam jenis rambu larangan, perintah, dan peringatan. Beberapa contoh dari rambu – rambu lalu lintas di Indonesia diilustrasikan pada Tabel 1. Simbol – simbol tersebut tidak digunakan dalam *training* atau

testing melainkan data primer yang diambil langsung di jalan raya.

Pengumpulan Data Primer

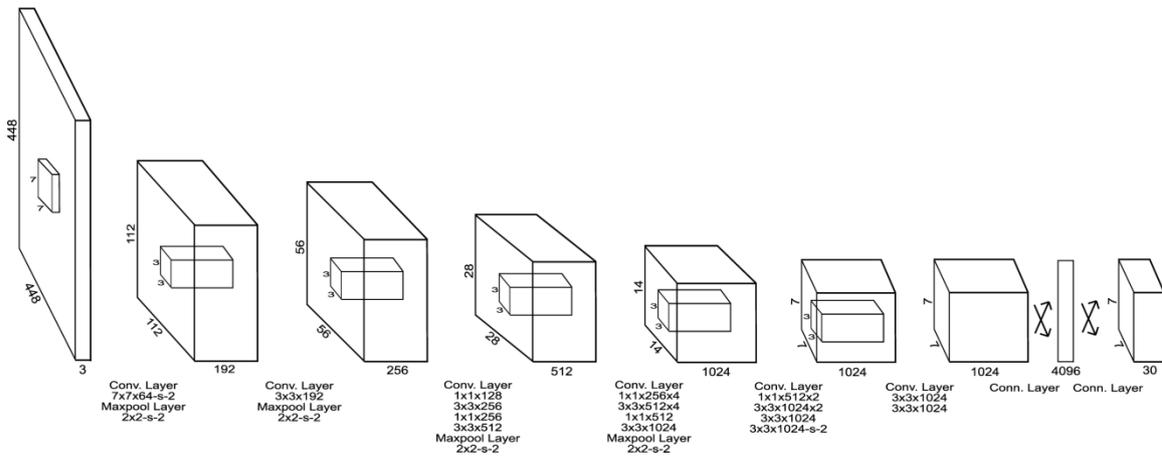
Dataset untuk *training* dan *testing* yang digunakan merupakan data primer berupa video - video yang direkam sepanjang jalan di Daerah Istimewa Yogyakarta. Video-video tersebut dipecah menjadi *frame* demi *frame* menjadi citra tunggal. Video dipecah menjadi per *frame* menggunakan *software* Free Video to JPG Converter (DVDVideoSoft, 2023). Sebuah video dapat hanya mengandung satu jenis rambu, ada juga yang mengandung beberapa rambu berbeda. Namun, video dengan satu jenis rambu lebih banyak dibandingkan video dengan beraneka jenis rambu. Beberapa sampel citra yang diekstraksi dari video tersebut ditunjukkan oleh Gambar 1. Berdasarkan citra rambu lalu lintas tersebut tampak kualitas citra yang berbeda karena pengaruh sumber cahaya karena waktu dan area yang berbeda.



Gambar 1 Beberapa Citra Rambu Lalu Lintas yang Diekstraksi dari Rekaman Vvideo

Algoritma Yolov8

Algoritma Yolov8 merupakan jenis *unified detection* yang masuk dalam algoritma *single shot detector*. Arsitektur dari algoritma Yolo secara umum ditunjukkan oleh Gambar 2. Algoritma tersebut menggunakan 24 *convolutional layers* dalam bentuk umum dan 9 *convolutional layers* dalam bentuk *fast Yolo* (Redmon et al., 2015). Ketika dikembangkan, algoritma ini dilatih pada dataset ImageNet dengan 1000 kelas (Russakovsky et al., 2015) pada *framework* Darknet (Redmon, 2024). Hasil validasi menunjukkan akurasi hingga 80% pada dataset ImageNet 2012 yang kompetitif terhadap hasil dari model GoogLeNet (Mishkin, 2012).



Gambar 2. Arsitektur Yolo Secara Umum (Redmon et al., 2015)

Pengembangan awal algoritma ini masih memiliki keterbatasan terutama batasan jumlah objek berbeda yang berdekatan dan ketidakmampuan mengenali objek berukuran kecil yang berkerumun atau berkelompok (Redmon et al., 2015). Namun berdasarkan penelitian (G. Yang et al., 2023) tentang deteksi kematangan buah tomat menggunakan Yolov8, akurasi algoritma ini lebih baik dibandingkan pendahulunya. Bahkan ukuran model hasil pelatihan berkurang cukup signifikan dengan kecepatan proses deteksi sekitar 138.8 FPS sehingga tepat untuk proses deteksi secara *real-time*. Selain penelitian tersebut, juga dijelaskan oleh (L. Yang & Shami, 2020), yang menerapkan algoritma ini secara *real-time*, bahwa Yolov8 mewarisi keunggulan pendahulunya dengan fitur tambahan yaitu perbaikan pada *training* yang adaptif, arsitektur yang dapat diubah, dan *data augmentation* yang lebih baik.

Hyper-parameter Pada *Machine Learning*

Algoritma *machine learning* setidaknya memiliki dua kategori parameter yaitu parameter model dan *hyper-parameter*. Parameter model merupakan parameter yang diinisialisasi dan diperbarui ketika *training* seperti bobot–bobot *neuron* dalam *neural network*. Sementara *hyper-parameter* merupakan parameter yang berfungsi untuk mengkonfigurasi model sebelum *training* atau menentukan algoritma yang akan dipakai untuk meminimalkan *lost function*, contohnya nilai *penalty parameter* dan *learning rate* (L. Yang & Shami, 2020).

Jenis *hyper-parameter* yang mengalami penyetelan (*tuning*) adalah *learning rate*, α . Namun, ukuran citra dan jumlah *epoch* ternyata memberikan perbedaan akurasi dalam *mean Average Precision* (mAP) sehingga kedua parameter tersebut juga mengalami *tuning*.

Indikator Kinerja Model

Kinerja dari model hasil *training* diukur menggunakan variabel *accuracy* (1), *precision* (2), dan *recall* (3). Ketiga variabel tersebut dihitung berdasarkan akumulasi *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN) yang didapatkan dari *confusion matrix*.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$precision = \frac{TP}{TP+FP} \quad (2)$$

$$recall = \frac{TP}{TP+FN} \quad (3)$$

Ketika hasil deteksi sama dengan sebenarnya dan bernilai *TRUE* maka disebut dengan TP, sedangkan hasil deteksi sama dengan sebenarnya dan bernilai *FALSE* maka disebut dengan TN. Jika hasil deteksi bernilai *TRUE* padahal seharusnya *FALSE* maka disebut dengan FP, sebaliknya disebut dengan FN (Nugroho et al., 2022) dan (Sumanto et al., 2022).

Persiapan Dataset

Persiapan dataset menggunakan aplikasi berbasis *browser* pada Roboflow (Roboflow, 2024). Dataset berupa citra yang diekstrak dari rekaman video yang diambil di sepanjang jalan di Daerah Istimewa Yogyakarta. Lalu, dataset citra diunggah di laman Roboflow dimana proses ini memerlukan waktu tergantung jumlah citra yang diunggah. Setelah itu, dilakukan proses *labelling* dengan memberikan keterangan berupa kelas dan *bounding box*. Selanjutnya, proses data *splitting* yaitu kegiatan membagi dataset ke dalam data *training*, *validation*, dan *testing*. Proses selanjutnya adalah *preprocessing* berupa proses normalisasi ukuran citra menjadi 640x640 px. Untuk menghasilkan dataset yang

bervariasi maka ditambahkan proses *augmentation* yaitu proses modifikasi citra, dalam hal ini berupa pemberian *noise* sebesar 2%, *blur* 1 px, dan kecerahan dikurangi sebesar 40% (*dark*). Seluruh citra yang telah melewati seluruh proses ini digunakan dalam tahap *training* di Google Colab (Google, 2024).

Implementasi Sistem

Sistem pendeteksi rambu lalu lintas ini diimplementasikan pada Raspberrypi 4 model B menggunakan Thonny (Annamaa, 2015). Untuk kemudahan mengoperasikan alat maka sistem dilengkapi dengan *user interface* yang direalisasikan menggunakan *QT designer* (QT Designer, 2024) yang ditampilkan pada layar LCD.

HASIL DAN PEMBAHASAN

Dataset yang digunakan sebagai data primer berupa citra tiga puluh jenis rambu lalu lintas yang diambil di jalanan Daerah Istimewa Yogyakarta sebanyak 11.157 buah yang telah dinormalisasi menjadi 640x640 px. Dari dataset tersebut dibagi ke dalam tiga kategori yaitu data *training* sebanyak 9.762 (87,5%), data *validation* 929 (8,3%), dan data *testing* sebesar 466 (4,2%). Citranya berupa citra RGB (*Red, Green, Blue*) dengan variasi *blur, noise*, dan kecerahan yang diturunkan.

Dalam pengujian secara *real-time*, model hasil *training* ditanamkan dalam realisasi alat yang ditunjukkan oleh Gambar 3. Di dalam alat tersebut terdapat kamera *webcam* dan *single board computer* Raspberrypi 4 model B yang memiliki RAM 8 Giga Byte. Alat ini dilengkapi dengan layar LCD untuk kemudahan navigasi karena terdapat *user interface* untuk pengoperasiannya.



Gambar 3 Realisasi Alat Pendeteksi Rambu Lalu Lintas

Sistem diuji kembali menggunakan data citra rambu lalu lintas berbeda yang direkam dalam format video yang dipecah-pecah menjadi citra tunggal. Data citra diambil di jalan-jalan sekitar provinsi Daerah Istimewa Yogyakarta pada siang dan malam hari. Jumlah citra yang siang hari sebanyak 1.954 citra dan malam hari

sebanyak 2.297 citra. Kedua kelompok citra tersebut sekaligus digunakan untuk mengukur kinerja sistem.

Pemilihan Nilai *Learning Rate*

Hyper-parameter pertama yang diuji adalah *learning rate, α*. Dalam percobaan terdapat tiga nilai *α* yang diuji yaitu *α* = {0,05 0,10 0,15}. Berdasarkan hasil *training* dengan variasi *learning rate* pada Tabel 2, nilai *learning rate* terbaik yaitu 0,05. *Learning rate* berpengaruh terhadap hasil akurasi model. Semakin kecil nilai *learning rate* maka semakin besar akurasi model karena tingkat ketelitiannya besar. Sebaliknya, semakin besar nilai *learning rate* maka semakin kecil nilai akurasinya karena ketelitiannya kecil. Di dalam pengujian, ukuran citra ditetapkan 320x320 px dengan harapan akan mampu menyeleksi nilai *α* terbaik dengan kualitas citra yang rendah.

Tabel 2 Hasil *Training* Model dengan Variasi *Learning Rate*

<i>Learning Rate</i>	<i>Epoch</i>	Ukuran Citra	mAP@0,5 <i>best.pt</i>
0,05	100	320 x 320	0,990
0,10	100	320 x 320	0,982
0,15	100	320 x 320	0,946

Penentuan Nilai *Epoch*

Dengan menggunakan nilai *α* terbaik pada langkah sebelumnya, pengujian nilai *epoch* dilakukan untuk menentukan *epoch* yang terbaik berdasarkan mAP dan waktu *training*nya. Variasi nilai *epoch* yang diuji adalah 50, 78, 100, dan 300. Berdasarkan *training* dengan variasi *epoch* pada Tabel 3 diketahui bahwa nilai *epoch* terbaik yaitu 78 dan 100. Pada penelitian ini akan menggunakan *epoch* 78 karena waktu pelatihan model lebih cepat dibandingkan *epoch* 100 dengan selisih waktu pelatihan model sebesar 1 jam 49 menit dan mAP sebesar 0,990.

Tabel 3 Hasil *Training* Model dengan Variasi *Epoch*

<i>Learning Rate</i>	<i>Epoch</i>	Ukuran Citra	mAP@0,5 <i>best.pt</i>
0,05	50	320 x 320	0,989
0,05	78	320 x 320	0,990
0,05	100	320 x 320	0,990
0,05	300	320 x 320	0,988

Penentuan Ukuran Citra

Nilai *α* dan *epoch* terpilih digunakan pada uji ukuran citra untuk mendapatkan nilai mAP tertinggi. Tiga variasi ukuran digunakan yaitu 320x320 px, 512x512 px, dan 640x640 px. Hasil *training* ditampilkan pada Tabel 4 yang menunjukkan bahwa

nilai mAP yang sama untuk 320 px dan 512 px, hanya terpaut 0,001 dengan citra berukuran 640 px. Dengan demikian, kami memutuskan untuk menggunakan citra berukuran 320 px sebagai input untuk *training*, *validation*, dan *testing* meskipun sebelumnya telah menggunakan ukuran citra 640 px. Untuk itu model hasil *training* yang disebut dengan *best.pt* digunakan sebagai model final untuk sistem ini.

Tabel 4 Hasil *Training* Model dengan Variasi Ukuran Citra

<i>Learning Rate</i>	<i>Epoch</i>	Ukuran Citra	mAP@0,5 <i>best.pt</i>
0,05	78	320 x 320	0,990
0,05	78	512 x 512	0,990
0,05	78	640 x 640	0,991

Hasil Pengujian Citra dengan *Augmentation*

Dengan menggunakan ketiga *hyper-parameter* yang terbaik berdasarkan hierarki pemilihan yang telah dijelaskan, maka model diuji dengan citra yang mengandung variasi *augmentation* yaitu *blur*, *dark*, dan *noise*. Jenis *noise* yang diberikan adalah *salt & pepper* karena *noise* yang umum muncul pada optik dan jenis *blur* yang digunakan adalah *Gaussian*. Hanya satu jenis rambu yang digunakan dalam pengujian yaitu rambu “hati-hati”.

Pada Tabel 5 merupakan variasi nilai *blur* pada citra. Dengan nilai *blur* sebesar 40%, 70%, dan 100% sistem masih dapat mengidentifikasi rambu lalu lintas dengan tepat. Namun, nilai *confidence level* turun seiring meningkatnya nilai persentase *blur*.

Tabel 5 Hasil Pengujian dengan Variasi *Blur*

<i>Blur (%)</i>	Hasil dalam <i>Confidence level</i>
40	Terdeteksi 92%
70	Terdeteksi 89%
100	Terdeteksi 87%

Pada Tabel 6 merupakan variasi nilai kecerahan (*dark*) pada citra. Kecerahan pada citra diturunkan sebesar 75%, 85%, dan 95%. Pengujian menunjukkan bahwa sistem dapat mengidentifikasi rambu lalu lintas dengan tepat. Nilai maksimum kecerahan pada citra yang masih dapat diidentifikasi adalah - 95%.

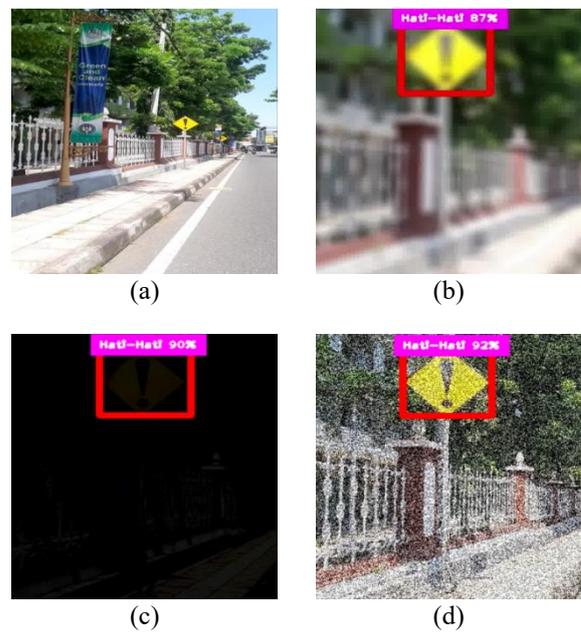
Tabel 6 Variasi Kecerahan (*dark*) pada Citra

Kecerahan (%)	Hasil dalam <i>Confidence level</i>
-75	Terdeteksi 92%
-85	Terdeteksi 92%
-95	Terdeteksi 90%

Sementara itu, Tabel 7 merupakan variasi nilai *noise* pada citra. Dengan nilai *noise* sebesar 40%, 70%, dan 100% sistem masih dapat mengidentifikasi rambu lalu lintas dengan tepat. Ilustrasi hasil pengujian untuk ketiga *augmentation* ditunjukkan oleh Gambar 4.

Tabel 7 Variasi *Noise* pada Citra

<i>Noise (%)</i>	Hasil dalam <i>Confidence level</i>
40	Terdeteksi 92%
70	Terdeteksi 92%
100	Terdeteksi 92%



Gambar 4 Hasil Pengujian dengan Variasi *Augmentation* pada Citra a) citra asli, b) *blur*, c) *dark*, d) *noise*

Hasil Pengujian Citra Data *Testing*

Pengujian selanjutnya menggunakan citra yang terdapat pada dataset *testing*. Total jumlah data *testing* pada penelitian ini adalah 466 citra. Berdasarkan pengujian yang telah dilakukan, didapatkan data hasil pengujian yang ditunjukkan pada Tabel 8. Nilai batas minimal *confidence level* pada saat pengujian diatur sebesar 70.

Berdasarkan *confusion matrix* pada Tabel 8 diperoleh nilai TP sebesar 491 yang artinya terdapat 491 rambu lalu lintas yang berhasil diidentifikasi dengan tepat sesuai kelasnya, TN sebesar 0 yang artinya terdapat 0 rambu lalu lintas yang teridentifikasi bernilai salah oleh sistem karena aktualnya memang tidak ada rambu lalu lintas di luar dataset yang yang diujikan, FP sebesar 2 yang artinya terdapat 2 rambu lalu lintas yang diidentifikasi secara salah, dan FN sebesar 39 yang artinya terdapat 39 rambu lalu lintas

yang tidak berhasil dideteksi padahal rambu tersebut termasuk ke dalam jenis rambu yang ada di dataset.

$$Precision = \frac{491}{491 + 2} = \frac{491}{493} = 0,99$$

$$Recall = \frac{491}{491 + 39} = \frac{491}{530} = 0,92$$

$$Accuracy = 0,92$$

Jika parameter tersebut diubah menjadi persentase diperoleh nilai *precision* sebesar 99%, *recall* 92%, dan *accuracy* 92%. Dapat disimpulkan bahwa sistem memiliki kinerja *overall* yang baik dalam mendeteksi rambu lalu lintas.

Tabel 8 Confusion Matrix Data Testing

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP 491	FP 2
	0 (Negative)	FN 39	TN 0

Hasil Pengujian Citra Berbeda Waktu

Pengujian selanjutnya adalah menggunakan data yang baru yang diambil berdasarkan perbedaan waktu yakni siang dan malam. Pengambilan data siang dilakukan pada pukul 09.00-12.00 WIB dan data malam pada pukul 19.00–23.00 WIB. Citra rambu yang diambil merupakan rambu di sepanjang jalan di Daerah Istimewa Yogyakarta.

Hasil pengujian data siang hari ditunjukkan pada Tabel 9. Berdasarkan hasil tersebut diperoleh nilai TP sebesar 1.686 yang artinya terdapat 1.686 rambu lalu lintas yang berhasil diidentifikasi dengan tepat sesuai kelasnya, TN sebesar 0 yang artinya terdapat 0 rambu lalu lintas teridentifikasi salah oleh sistem karena aktualnya memang tidak ada rambu lalu lintas di luar

Tabel 9 Confusion Matrix pada Siang Hari

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP 1686	FP 69
	0 (Negative)	FN 355	TN 0

dataset yang yang diujikan, FP sebesar 69 yang artinya sistem mengidentifikasi secara salah 69 rambu lalu lintas, dan FN sebesar 355 yang artinya terdapat 355 rambu lalu lintas yang tidak berhasil dideteksi meskipun termasuk dalam dataset *training*.

$$Precision = \frac{1686}{1686 + 69} = \frac{1686}{1755} = 0,96$$

$$Recall = \frac{1686}{1686 + 355} = \frac{1686}{2041} = 0,83$$

$$Accuracy = 0,80$$

Dengan mengubah nilai di atas dalam persentase maka disimpulkan bahwa nilai parameter *precision* sebesar 96%, *recall* 83%, dan *accuracy* 80%. Hasil ini mengindikasikan bahwa sistem cukup baik dalam mengidentifikasi jenis rambu lalu lintas pada siang hari.

Selanjutnya pengujian menggunakan data malam hari ditunjukkan pada Tabel 10. Hasil menunjukkan bahwa nilai TP sebesar 1590 yang artinya terdapat 1590 rambu lalu lintas yang berhasil diidentifikasi secara benar sesuai kelasnya, TN sebesar 0 yang artinya terdapat 0 rambu lalu lintas yang teridentifikasi bernilai salah oleh sistem karena aktualnya memang tidak ada rambu lalu lintas diluar dataset yang yang diujikan, FP sebesar 118 yang artinya sistem mengidentifikasi

Tabel 10 Confusion Matrix pada Malam Hari

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP 1590	FP 118
	0 (Negative)	FN 673	TN 0

secara salah 118 rambu lalu lintas, dan FN sebesar 673 yang artinya terdapat 673 rambu lalu lintas yang tidak berhasil dideteksi meskipun ada dalam jenis rambu yang ada di dataset.

$$Precision = \frac{1590}{1590 + 118} = \frac{1590}{1708} = 0,93$$

$$Recall = \frac{1590}{1590 + 673} = \frac{1590}{2263} = 0,70$$

$$Accuracy = 0,67$$

Dalam bentuk persentase maka diperoleh nilai parameter *precision* sebesar 93%, *recall* 70%, dan *accuracy* 67%. Dapat diketahui bahwa kinerja sistem pada malam hari mengalami penurunan dibandingkan

siang hari. Hal tersebut dipengaruhi oleh faktor kondisi cahaya. Pada malam hari sistem kurang dapat bekerja dengan baik karena sulit mengenali rambu lalu lintas dengan kondisi cahaya yang kurang, namun dataset *training* yang telah mengalami *augmentation* berdampak pada cukup tingginya nilai TP pada pengujian ini.

Pada pengujian secara *real-time* kendaraan yang digunakan merupakan sepeda motor. Hasil pengujian pada siang dan malam hari hanya efektif jika kecepatan kendaraan kurang dari 40 km/jam karena menyesuaikan dengan kecepatan pengolahan perangkat Raspberrypi yang digunakan dan kualitas kamera yang hanya berupa *webcam*. Meskipun demikian, dapat disimpulkan bahwa kinerja dari model Yolov8 yang telah didapatkan cukup baik dalam mendeteksi rambu – rambu lalu lintas baik pada siang hari, malam hari, dan citra dengan *augmentation*.

KESIMPULAN

Telah dilakukan perancangan sistem untuk mengidentifikasi tiga puluh jenis rambu lalu lintas di Indonesia menggunakan algoritma Yolov8. Sistem ini dibangun dengan menggunakan data primer berupa lalu lintas sepanjang jalan di Daerah Istimewa Yogyakarta. Pembuatan sistem didahului dengan pemilihan tiga nilai *hyper-parameter* lalu melatih model menggunakan data primer dan *hyper-parameter* tersebut. Sistem ditanamkan dalam *Raspberrypi 4* model B yang diuji secara *real-time* dengan memasangkannya pada kendaraan bermotor roda dua. Hasil pengujian menyimpulkan bahwa deteksi simbol lalu lintas cukup akurat pada siang hari (di atas 80%) dibandingkan pada malam hari (di atas 60%) karena kekurangan cahaya meskipun pada data *training* telah ditambahkan *augmentation* seperti *blur* dan *dark*. Di masa mendatang, perangkat keras yang digunakan dapat ditingkatkan seperti jenis *single board computer* dan kameranya agar model yang telah dikembangkan dapat memberikan hasil optimal.

UCAPAN TERIMA KASIH

Kami mengucapkan terima kasih kepada Laboratorium Instrumentasi dan Kontrol, Departemen Teknik Elektro dan Informatika, Sekolah Vokasi Universitas Gadjah Mada yang telah menyediakan fasilitas agar penelitian ini bisa diselesaikan.

DAFTAR PUSTAKA

- Annamaa, A. (2015). Introducing Thonny, a Python IDE for learning programming. *Proceedings of the 15th Koli Calling Conference on Computing Education Research*, 117–121. <https://doi.org/10.1145/2828959.2828969>
- Auliya, A., Pradani, W., & Haryanto, T. (2022). Kenaf Flower Detection using YOLOv3. *2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, 269–272. <https://doi.org/10.1109/AiDAS56890.2022.9918691>
- BRIN. (2022, February 15). *BRIN Kembangkan Mobil Listrik Tanpa Pengemudi yang Ramah Lingkungan*. <https://www.brin.go.id/news/99513/brin-kembangkan-mobil-listrik-tanpa-pengemudi-yang-ramah-lingkungan>
- DVDVideoSoft. (2023). *Vide to JPG Converter*. <https://www.dvdvideosoft.com/products/dvd/Free-Video-to-JPG-Converter.htm>
- Google. (2024). *Google Colaboratory*. <https://colab.google>
- Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (2017). Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3296–3297. <https://doi.org/10.1109/CVPR.2017.351>
- Lavanya, G., & Pande, S. D. (2023). Enhancing Real-time Object Detection with YOLO Algorithm. *EAI Endorsed Transactions on Internet of Things*, 10, 1–9. <https://doi.org/10.4108/eetiot.4541>
- Mishkin, D. (2012). *Models accuracy on imagenet 2012 val*. <https://github.com/BVLC/Caffe/wiki/Models-Accuracy-on-ImageNet-2012-Val>. <https://github.com/BVLC/caffe/wiki/Models-accuracy-on-ImageNet-2012-val>
- Nacir, O., Amna, M., Imen, W., & Hamdi, B. (2022). Yolo V5 for Traffic Sign Recognition and Detection Using Transfer Learning. *2022 IEEE International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, 1–4. <https://doi.org/10.1109/CISTEM55808.2022.10044022>
- Nugroho, A. A., Wijaya, W., Hendry, J., & Sumanto, B. (2022). Seleksi Fitur Aroma Teh Kombucha menggunakan ANN untuk Optimasi Kinerja Sistem E-nose. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(2), 334. <https://doi.org/10.26760/elkomika.v10i2.334>
- QT Designer. (2024). *QT Designer*. <https://doc.qt.io>
- Rafly Adli. (2023, April 1). *Polri Beberkan Ternyata Pegawai BUMN Paling Banyak Lakukan Pelanggaran Lalu Lintas*. <https://oto.detik.com/catatan-pengendara-mobil/d-6649987/polri-beberkan-ternyata-pegawai-bumn-paling-banyak-lakukan-pelanggaran-lalu-lintas>
- Redmon, J. (2024). *Darknet: Open source neural networks in C*. <http://pjreddie.com/Darknet/>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object

- Detection. *Computer Vision and Pattern Recognition*.
<http://arxiv.org/abs/1506.02640>
- Roboflow. (2024). *Vision Models with Roboflow*.
<https://Roboflow.Com>. <https://roboflow.com>
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. <https://doi.org/10.1007/s11263-015-0816-y>
- Saptohutomo, A. P. (2022). *Korlantas Polri Catat 94.617 Kecelakaan pada Januari-September 2022*. <https://nasional.kompas.com/read/2022/11/20/15200561/korlantas-polri-catat-94617-kecelakaan-pada-januari-september-2022>
- Saputra, D. (2022). *Setelah BSD, Ambisi Kendaraan Otonom akan Dibawa Juga ke IKN Nusantara*. <https://ekonomi.bisnis.com/read/20220521/98/1535416/setelah-bsd-ambisi-kendaraan-otonom-akan-dibawa-juga-ke-ikn-nusantara>
- Sumanto, B., Java, D. R., Wijaya, W., & Hendry, J. (2022). Seleksi Fitur Terhadap Performa Kinerja Sistem E-Nose untuk Klasifikasi Aroma Kopi Gayo. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, *21*(2), 429–438. <https://doi.org/10.30812/matrik.v21i2.1495>
- Wang, C., Guo, J., Wang, S., Wu, Q., Wang, S., & Wu, R. (2022). Research on Gesture Recognition Algorithm Based on Lightweight YOLOv4. *2022 2nd International Conference on Computation, Communication and Engineering (ICCCE)*, 74–78. <https://doi.org/10.1109/ICCCE55785.2022.10036237>
- Yang, G., Wang, J., Nie, Z., Yang, H., & Yu, S. (2023). A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention. *Agronomy*, *13*(7), 1824. <https://doi.org/10.3390/agronomy13071824>
- Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. <https://doi.org/10.1016/j.neucom.2020.07.061>
- Yung, N. D. T., Wong, W. K., Juwono, F. H., & Sim, Z. A. (2022). Safety Helmet Detection Using Deep Learning: Implementation and Comparative Study Using YOLOv5, YOLOv6, and YOLOv7. *2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, 164–170. <https://doi.org/10.1109/GECOST55694.2022.10010490>

Halaman ini sengaja dikosongkan