

Deteksi Pengenalan Rambu Lalu-lintas Berdasarkan Citra Menggunakan Algoritma Yolo di Daerah Istimewa Yogyakarta

Rois Munfarid ¹, Akhmad Fadjeri ²

^{1,2}Universitas Ma'arif Nahdlatul Ulama, Kebumen, Indonesia

roisaa1252@gmail.com

Received: 09/02/2026

Revised: 15/02/2026

Accepted: 14/03/2026

Copyright©20xx by authors. Authors agree that this article remains permanently open access under the terms of the Creative Commons

Abstrak

Tingginya angka kecelakaan lalu lintas secara global dan nasional menunjukkan perlunya inovasi teknologi untuk meningkatkan keselamatan berkendara, khususnya dalam membantu pengemudi mengenali rambu-lalu lintas secara cepat dan akurat. Penelitian ini bertujuan mengembangkan sistem pengenalan rambu-lalu lintas berbasis kecerdasan buatan menggunakan algoritma YOLOv8 yang disesuaikan dengan kondisi lokal di Daerah Istimewa Yogyakarta. Metode yang digunakan adalah model pengembangan perangkat lunak *Waterfall* yang meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian dan validasi, serta tahap implementasi dan pemeliharaan. Dataset penelitian diperoleh dari *Google Street View* sebanyak 984 citra rambu, kemudian dianotasi menggunakan *Roboflow* dan dibagi menjadi data pelatihan, validasi, dan pengujian dengan rasio 80:10:10. Proses pelatihan model dilakukan menggunakan *framework Ultralytics* pada platform *Google Colab* selama 200 *epoch*. Hasil pelatihan menunjukkan nilai *precision* 0,891, *recall* 0,869, *mAP50* 0,973, dan *mAP50-95* 0,889. Pengujian menunjukkan model mampu mendeteksi sebagian besar kelas rambu dengan tingkat *recall* yang sangat tinggi pada berbagai kondisi citra. Model terlatih kemudian diimplementasikan ke dalam aplikasi berbasis Python dengan antarmuka Tkinter yang mendukung deteksi gambar statis serta deteksi *real-time* melalui webcam. Secara keseluruhan, sistem yang dikembangkan memiliki performa baik dan berpotensi mendukung peningkatan keselamatan lalu lintas melalui pengenalan rambu otomatis di lingkungan transportasi cerdas masa depan yang lebih adaptif dan berkelanjutan global.

Kata kunci: Rambu lalu-lintas, YOLOv8, Kecerdasan buatan

Abstract

The high number of traffic accidents globally and nationally highlights the urgent need for technological innovation to improve road safety, particularly in assisting drivers to recognize traffic signs quickly and accurately. This study aims to develop a traffic sign recognition system based on artificial intelligence using the YOLOv8 algorithm, adapted to local conditions in Daerah Istimewa Yogyakarta. The research employed the *Waterfall* software development model, consisting of requirements analysis, system

design, implementation, testing and validation, as well as deployment and maintenance. The dataset was collected from Google Street View, comprising 984 images of traffic signs, which were annotated using Roboflow and divided into training, validation, and testing sets with an 80:10:10 ratio. Model training was conducted using the Ultralytics framework on Google Colab for 200 epochs. The training results achieved a precision of 0.891, recall of 0.869, mAP50 of 0.973, and mAP50-95 of 0.889. Testing results indicate that the model successfully detected most traffic sign classes with very high recall under various image conditions. The trained model was implemented into a Python-based application with a Tkinter graphical user interface, supporting both static image detection and real-time detection via webcam. Overall, the developed system demonstrates strong performance and significant potential to enhance road safety through automatic traffic sign recognition in future intelligent transportation systems.

Keywords: Traffic signs, YOLOv8, Artificial Intelligence

1. Pendahuluan

Keselamatan di jalan raya merupakan isu global yang sangat penting dan menjadi perhatian banyak pihak. Hal ini disebabkan oleh tingginya angka kecelakaan lalu lintas yang terjadi setiap tahunnya di berbagai belahan dunia. Menurut data yang dirilis oleh *World Health Organization* (WHO), kecelakaan lalu lintas menyebabkan sekitar 1,19 juta kematian setiap tahunnya (WHO, 2019), menjadikannya salah satu penyebab utama kematian di seluruh dunia. Selain itu, jutaan orang lainnya mengalami cedera serius yang sering kali berdampak jangka panjang terhadap kualitas hidup mereka. Salah satu faktor yang sering menjadi penyebab tingginya angka kecelakaan adalah kurangnya pemahaman dan kesadaran pengemudi terhadap tanda-tanda lalu lintas. Tanda-tanda lalu lintas dirancang untuk memberikan informasi dan instruksi kepada pengemudi guna menjaga keselamatan di jalan. Namun, sering kali tanda-tanda ini diabaikan, baik karena ketidaktahuan pengemudi maupun tidak terlihat dengan jelas, misalnya dalam kondisi cuaca buruk, pencahayaan yang minim, atau lingkungan jalan yang rumit. Di Indonesia, fenomena ini juga menjadi perhatian serius, mengingat banyaknya kecelakaan lalu lintas yang disebabkan oleh kelalaian dalam mematuhi tanda-tanda jalan.

Korlantas Polri merilis data angka kecelakaan yang terjadi di Indonesia selama tahun 2024. Mereka mencatat sebanyak 152 ribu kasus kecelakaan lalu lintas di Indonesia (Nurhuda, 2024). Sebagian besar insiden ini disebabkan oleh pelanggaran aturan lalu lintas, termasuk ketidakpatuhan terhadap rambu-rambu jalan. Data ini menegaskan perlunya langkah-langkah inovatif untuk meningkatkan keselamatan di jalan, salah satunya melalui pemanfaatan teknologi. Dalam beberapa tahun terakhir, teknologi kecerdasan buatan (*Artificial Intelligence/AI*) telah menunjukkan perkembangan yang sangat pesat. Kecerdasan buatan merupakan bidang ilmu komputer yang fokus dalam meniru manusia yang dapat diterapkan dalam perangkat komputer atau perangkat berbasis teknologi. *Artificial intelligence* terdapat proses dalam pembelajranya selayaknya manusia (Fadjeri et al., 2025). Salah satu penerapan AI yang signifikan adalah dalam bidang pengenalan objek, yang memungkinkan komputer untuk mengidentifikasi dan mengklasifikasikan berbagai benda dalam suatu citra. Citra adalah gambaran yang dikeluarkan dari analisis baik manual dengan mata manusia atau dengan alat rekam optik kamera, citra yang dihasilkan dari pengamatan dapat berupa foto maupun video (Fadjeri, 2024).

Dalam pengenalan objek, algoritma YOLO (*You Only Look Once*) menjadi salah satu teknologi yang menonjol. Algoritma ini dirancang untuk mendeteksi dan mengklasifikasikan objek secara real-time dengan tingkat kecepatan dan akurasi yang tinggi (Riansyah & Mirza, 2023). Seperti penelitian yang dilakukan oleh Muh. Ikbal dan Rizal Adi Saputra yang berjudul “PENGENALAN RAMBU LALU LINTAS MENGGUNAKAN METODE *YOLOV8*” dengan hasil: Hasil pengujian menunjukkan bahwa model ini memiliki kinerja yang baik, dengan nilai *Precision* sebesar 0,993, *Recall* sebesar 0,999, *mAP50* sebesar 0,995, dan *mAP50-95* sebesar 0,984 (IKBAL & Saputra, 2024). Penelitian yang dilakukan oleh Arief dan Rezki Wulandari yang berjudul “SISTEM DETEKSI DAN PENGENALAN RAMBU LALU LINTAS DI INDONESIA MENGGUNAKAN ALGORITMA *YOLOV4*” dengan hasil: sistem yang mendeteksi dan mengenali rambu-rambu lalu lintas yang sudah dites secara realtime. Hasil *mAP* (*mean Average Precision*) pada sistem ini sebanyak 95.15% (Arief, 2021). Penelitian yang dilakukan oleh Riska Pauziah dan Asti herliana yang berjudul “Implementasi Deteksi Rambu Lalu Lintas Menggunakan Metode *Extreme Learning Machine*” dengan hasil: Metode *extreme learning machine* dapat diimplementasikan untuk mendeteksi dan mengklasifikasi gambar rambu-rambu lalu lintas dengan menggunakan proporsi 70% data *training* dan 30% dengan akurasi yang diperoleh sebesar 97% (Pauziah & Herliana, 2021). Penelitian yang dilakukan oleh Namri Fatkhin dan Akhmad Fadjeri yang berjudul “Pembelajaran Mesin Untuk Deteksi Helm Keselamatan Menggunakan Algoritma *YOLOv8*” dengan hasil: Metrik evaluasi presisi mencapai 77,7%, nilai *recall* sebesar 60,7%, dan *mAP* mencapai 68,9% dengan waktu yang berlangsung selama 1 jam 7 menit untuk pelatihan model (Fatkhin & Fadjeri, 2024). Dengan keunggulan tersebut, teknologi YOLO berpotensi diterapkan dalam sistem transportasi untuk meningkatkan keselamatan berkendara, khususnya dalam pengenalan rambu lalu lintas.

Implementasi teknologi ini di Indonesia menghadapi tantangan tersendiri. Salah satu tantangan utama adalah variasi bentuk, warna, dan desain tanda lalu lintas yang tidak selalu seragam di seluruh wilayah. Selain itu, kondisi jalan di Indonesia yang beragam, mulai dari jalan perkotaan hingga pedesaan, juga dapat memengaruhi kinerja algoritma. Untuk mengatasi tantangan ini, diperlukan dataset lokal yang relevan dan pelatihan model yang disesuaikan dengan kondisi spesifik di Indonesia. Salah satu wilayah yang dapat dijadikan studi kasus adalah Daerah Istimewa Yogyakarta (DIY). Berdasarkan *website* Harian Jogja, Yogyakarta merupakan destinasi wisata terfavorit masyarakat Indonesia (Khafid, 2024). Implementasi teknologi ini di Yogyakarta tidak hanya dapat membantu meningkatkan keselamatan berkendara tetapi juga memberikan manfaat bagi wisatawan yang belum tahu dengan kondisi lalu lintas di daerah tersebut.

Agar pengembangan sistem pengenalan rambu lalu lintas ini berjalan secara terstruktur dan sistematis, penelitian ini menggunakan metode pengembangan perangkat lunak *Waterfall*. Metode *Waterfall* adalah metode pengembangan perangkat lunak yang sistematis dan berurutan, dimulai dari tahap analisis kebutuhan, desain sistem, implementasi, pengujian, sampai tahap pemeliharaan (Fadjeri & Hidayat, 2024).

Penerapan teknologi pengenalan citra berbasis algoritma YOLO memiliki potensi besar untuk memberikan solusi nyata dalam meningkatkan keselamatan di jalan raya. Implementasi sistem ini diharapkan berkontribusi dalam pengurangan kecelakaan lalu lintas dan peningkatan kesadaran pengendara, dan membantu navigasi bagi wisatawan yang belum tahu kondisi lalu lintas di suatu daerah, seperti di wilayah seperti Daerah Istimewa Yogyakarta. Tujuan dari penelitian ini yaitu untuk

mengembangkan model dalam mengenali rambu lalu-lintas yang akurat dan relevan dengan kebutuhan lokal. Upaya ini diharapkan memberikan kontribusi positif terhadap keselamatan jalan raya.

2. Metode Penelitian

2.1 Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini yaitu model *waterfall*. Model *waterfall* merupakan model pengembangan yang sering digunakan untuk pengembangan perangkat lunak. Model ini dikembangkan secara berurutan dari tahap awal sampai tahap akhir pengembangan sistem. Tahap berikutnya tidak dapat dilakukan ketika tahap sebelumnya selesai dilaksanakan dan tidak bisa mengulang ke tahap sebelumnya (Pricillia & Zulfachmi, 2021). Berikut adalah tahapan-tahapan pengembangan model *waterfall*.

2.1.1 Analisis Kebutuhan

Pada tahap ini dilakukan analisis terhadap kebutuhan sistem yaitu: jenis rambu lalu-lintas yang akan dikenali, perangkat keras dan perangkat lunak yang digunakan, dan target akurasi.

2.1.2 Perancangan Sistem

Perancangan sistem/desain adalah tahap menciptakan alur atau proses sistem dan cara karakteristiknya muncul (Fitriana & Fadjeri, 2025). Pada tahap ini, perancangan sistem yaitu: menentukan arsitektur sistem, desain dataset menggunakan dataset lokal yang diambil dari *google street view*, dan menggunakan model YOLOV8.

2.1.3 Implementasi

Pada tahap ini, implementasi yaitu: akuisisi dan persiapan dataset, *preprocessing data*, pelatihan model yolo, evaluasi model, dan implementasi deteksi rambu.

2.1.4 Pengujian dan Validasi

Pada tahap ini meliputi uji model dengan berbagai kondisi pencahayaan dan jarak, bandingkan hasil prediksi dengan hasil sebelumnya, dan jika akurasi rendah ulangi dengan dataset yang lebih besar atau teknik tuning model.

2.1.5 Implentasi dan Pemeliharaan

Pada tahap ini yang dilakukan yaitu *deploy* model ke sistem *real-time* misalnya berbasis python dan pantau performa model setelah implementasi dan lakukan *update* model jika diperlukan.

2.2 Teknik Pengumpulan Data

Teknik pengumpulan data merupakan cara yang digunakan oleh peneliti untuk mengumpulkan data-data penelitian dari sumber data (Putri & Murhayati, 2025). Teknik pengumpulan data merupakan langkah paling penting dalam penelitian, karena tujuan utama dalam penelitian yaitu memperoleh data. Tanpa teknik pengumpulan data yang tepat, peneliti tidak akan mendapatkan data yang sesuai dengan standar yang ditetapkan. Teknik pengumpulan data yang digunakan penulis yaitu dengan mengambil citra gambar secara langsung untuk digunakan sebagai dataset. Dalam penelitian ini, dataset yang diperlukan berbentuk foto rambu lalu lintas yang diperoleh

dari *Google Maps (Street View)* sebanyak 984 gambar. Dataset yang telah dikumpulkan akan diproses *training, validation, dan testing* dengan rasio pembagian sebesar 80:10:10.

3. Hasil dan Pembahasan

3.1 Hasil Analisis Kebutuhan

Pada tahap awal pengembangan sistem, perlu dilakukan analisis kebutuhan untuk memahami dan menentukan spesifikasi dasar yang diperlukan dalam mengenali rambu lalu-lintas.

3.1.1 Jenis Rambu Lalu-lintas

Sistem dirancang untuk mengenali jenis rambu lalulintas yang umum ditemui di Yogyakarta, yaitu: Rambu peringatan seperti hati-hati, persimpangan, dan tempat penyebrangan anak-anak. Rambu larangan seperti dilarang parkir, dilarang putar balik, dan dilarang masuk. Rambu perintah seperti wajib mengikuti arah jalan.

3.1.2 Perangkat Keras

Untuk menjalankan proses pelatihan dan pengujian model, gunakan sebuah perangkat laptop ASUS Vivobook M513UA yang memiliki spesifikasi cukup memadai untuk menjalankan proses komputasi.

3.1.3 Perangkat Lunak

Perangkat lunak yang dipakai dalam pembuatan model yaitu: python sebagai bahasa pemrograman, *roboflow* untuk labeling atau anotasi pada dataset, YOLOv8 sebagai algoritma untuk deteksi objek, OpenCV untuk proses pengolahan citra dan akses kamera, dan *Google Colab* untuk pelatihan model.

3.1.4 Target Akurasi

Target performa sistem adalah memiliki akurasi diatas 91% berdasarkan evaluasi terhadap dataset uji, agar dapat diandalkan dalam aplikasi nyata.

3.2 Hasil Perancangan Sistem

Hasil desain dibuat untuk menggambarkan aplikasi deteksi pengenalan rambu lalu lintas. Berikut adalah hasil desain sistem yang dibuat oleh peneliti ditampilkan dalam bentuk *flowchart* dan desain *user interface*.

3.2.1 Flowchart



Gambar 1. *flowchart*

3.2.2 *User Interface*



Gambar 2. Halaman Awal

Gambar di atas merupakan sketsa antarmuka pengguna untuk sebuah aplikasi pengenalan rambu lalu lintas berbasis yolo.

3.3 Hasil Implementasi

3.3.1 Akuisisi dan Persiapan Dataset

3.3.1.1 Mengumpulkan gambar rambu lalu lintas dari berbagai sudut pandang dan berbagai kondisi pencahayaan di *google street view*.



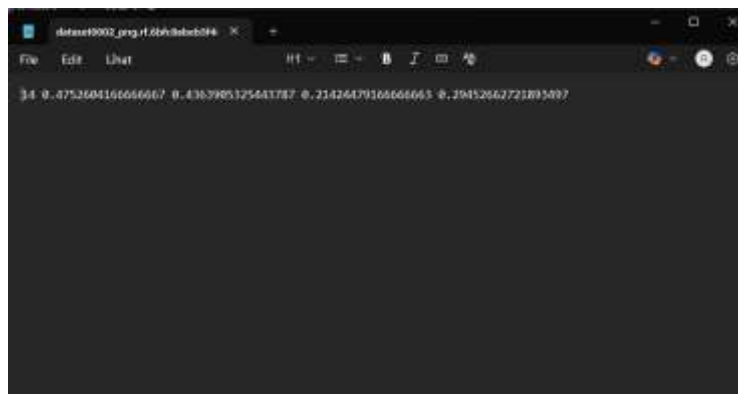
Gambar 3. Gambar rambu lalu-lintas

3.3.1.2 Melakukan anotasi gambar menggunakan *Roboflow* untuk menandai posisi rambu dan kelasnya.



Gambar 4. Proses anotasi gambar

3.3.1.3 Menyimpan anotasi gambar dalam format YOLO (.txt).



Gambar 5. Hasil anotasi dalam format .txt

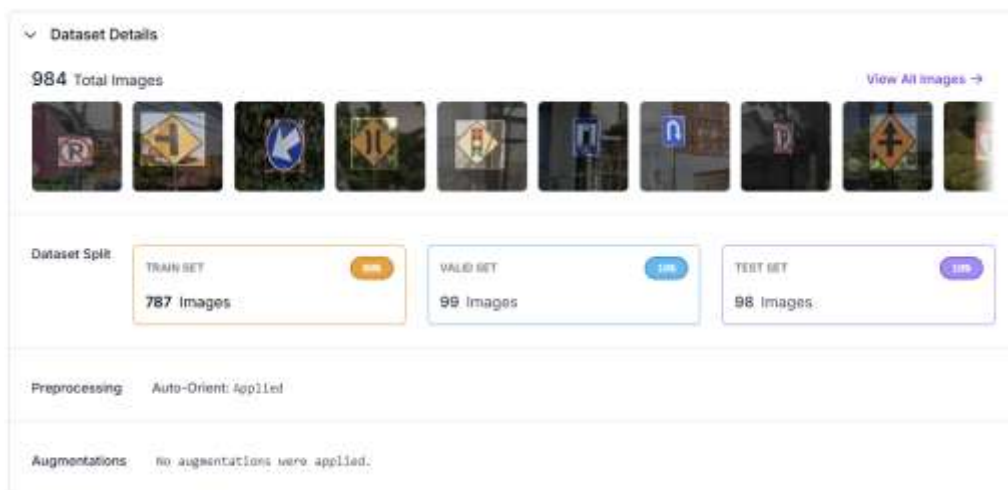
3.3.1.4 Membagi dataset ke dalam 3 bagian yaitu *training*, *validation*, dan *testing*, dengan perbandingan 80%, 10%, dan 10%.



Gambar 6. Pembagian dataset

3.3.2 Preprocessing Data

Pada tahap ini dataset tidak dilakukan *resize* maupun augmentasi dikarenakan data yang dipakai sudah cukup.



Gambar 7. Preprocessing data

3.3.3 Pelatihan Model Yolo

Pada tahap ini dataset yang telah di anotasikan dan dibagi menjadi 3 bagian yaitu *training*, *validation*, dan *testing* kemudian dilakukan proses pelatihan model yolo di *Google Colab* menggunakan *Framework Ultralytics* dan *epoch* sebanyak 200 kali. Hasil dari *training* model menggunakan *Google Colab* memperoleh hasil *Pressicion* sebesar 0,891, *Recall* sebesar 0,869, *mAP50* sebesar 0,973, dan *mAP50-95* sebesar 0,889.

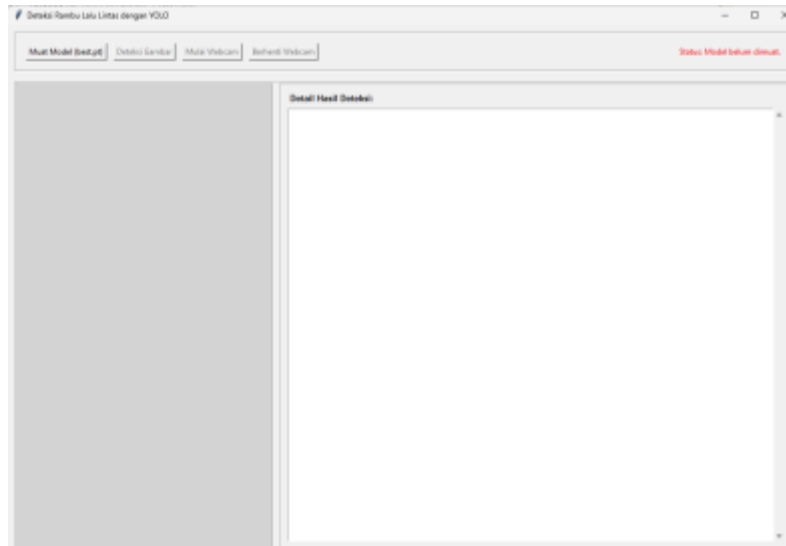
3.4 Peengujian dan Validasi

Secara umum model menunjukkan kinerja yang sangat kuat di sebagian besar kelas. *Recall* (R) untuk hampir semua kelas adalah 1.0 atau sangat mendekati 1.0 (misalnya, dilarang belok kanan, dilarang berhenti, dilarang parkir, hati-hati, dan banyak lagi). Ini mengindikasikan bahwa model sangat efektif dalam mengidentifikasi semua instance rambu lalu lintas yang sebenarnya ada dalam gambar.

3.5 Implementasi dan Pemeliharaan

Pada tahap ini peneliti mengimplementasikan hasil *training* model (*best.pt*) ke dalam aplikasi berbasis python dengan menggunakan tkinter. Tkinter merupakan pustaka standar Python untuk membuat antarmuka pengguna grafis (GUI).

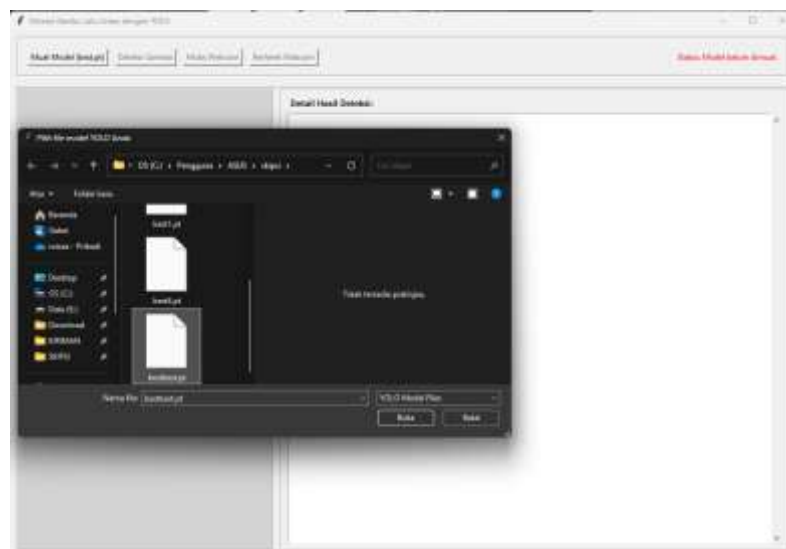
3.5.1 Tampilan Awal



Gambar 8. Tampilan halaman awal

Dari gambar di atas merupakan tampilan awal saat membuka aplikasi deteksi rambu lalu lintas menggunakan yolo.

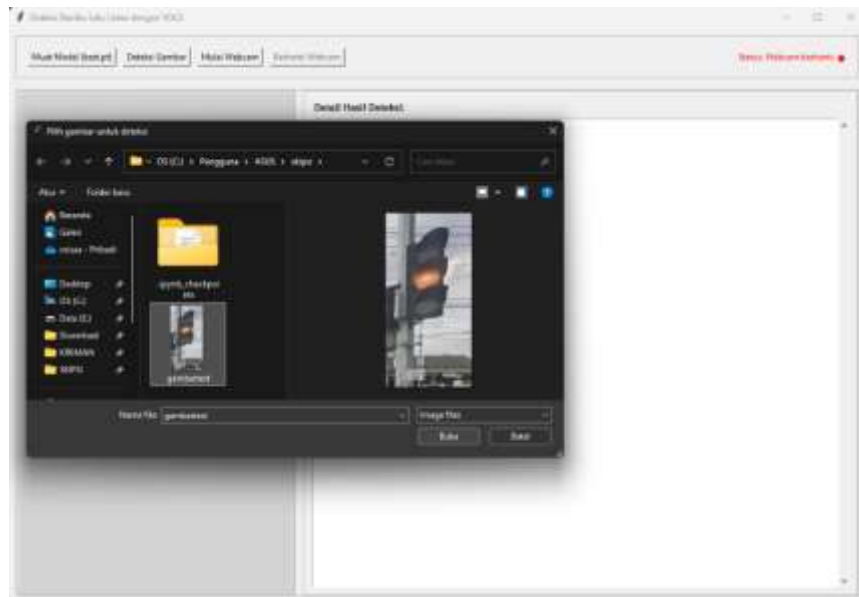
3.5.2 Memasukkan model (*best.pt*)



Gambar 9. mengunggah model (*best.pt*)

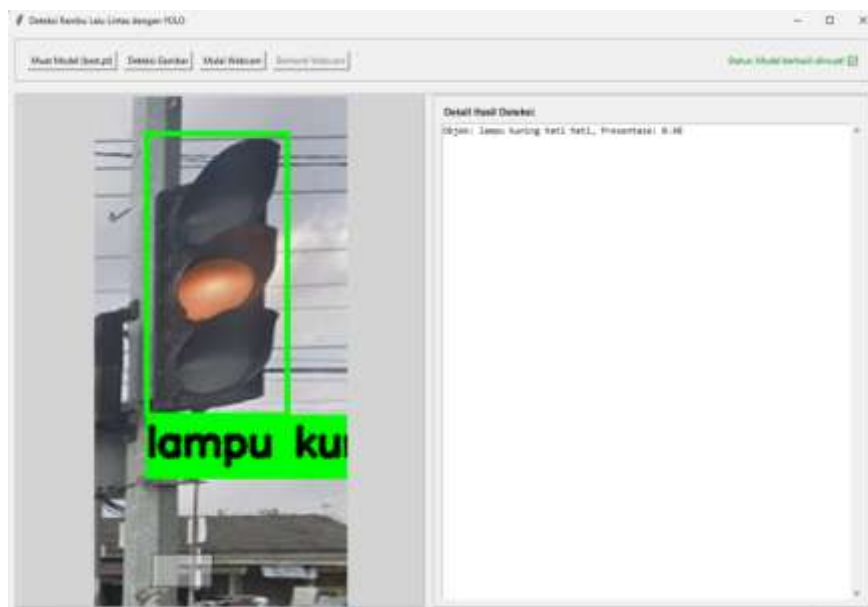
Dari gambar di atas dapat dijelaskan untuk memasukkan model yolo (*best.pt*) kita klik tombol “Muat Model (*best.pt*)”, tombol ini berfungsi untuk memasukkan model yolo dari file direktori penyimpanan. Klik “Buka” jika sudah memilih model maka model sudah terunggah ke dalam aplikasi.

3.5.3 Deteksi Gambar



Gambar 10. Deteksi Gambar

Dari gambar di atas dapat dijelaskan untuk memasukkan gambar yang akan dideteksi kita klik tombol “Deteksi Gambar”, tombol ini berfungsi untuk memasukkan gambar yang akan dideteksi dari file direktori penyimpanan. Klik “Buka” jika sudah memilih model maka gambar sudah terunggah ke dalam aplikasi.



Gambar 11. Hasil deteksi gambar

Gambar di atas merupakan tampilan dari hasil deteksi gambar dengan gambar di sebelah kiri dan hasil deteksi di sebelah kanan. Pada kolom hasil deteksi menampilkan “Objek berupa lampu kuning hati hati dengan presentase sebesar 0,98”.

3.5.4 Deteksi *real-time*



Gambar 12. Deteksi *real-time*

Gambar di atas merupakan tampilan deteksi secara *real-time* menggunakan *webcam*. Cara melakukan deteksi menggunakan *webcam* kita hanya perlu klik tombol “Mulai Webcam” maka akan tampil seperti gambar di atas. Jika kita telah selesai melakukan deteksi menggunakan *webcam* maka kita klik tombol “Berhenti Webcam” maka *webcam* akan berhenti.

4. Kesimpulan

Penelitian ini berhasil mengembangkan dan mengimplementasikan sistem pengenalan rambu lalu lintas berbasis kecerdasan buatan menggunakan algoritma YOLOv8 yang disesuaikan dengan kondisi rambu di Daerah Istimewa Yogyakarta. Proses pengembangan dimulai dari analisis kebutuhan, perancangan sistem, pengumpulan serta anotasi dataset, hingga pelatihan model menggunakan platform *Google Colab*. Dataset dibagi menjadi data *training*, *validation*, dan *testing* dengan rasio perbandingan 80:10:10, kemudian dilatih selama 200 *epoch* menggunakan *framework Ultralytics*. Hasil pelatihan menunjukkan performa model dengan nilai *precision* sebesar 0,891, *recall* 0,869, *mAP50* sebesar 0,973, dan *mAP50-95* sebesar 0,889, yang secara umum telah mendekati dan sebagian memenuhi target performa sistem. Pada tahap pengujian, model mampu mengenali sebagian besar rambu dengan tingkat *recall* yang sangat tinggi pada berbagai kelas, menandakan kemampuan deteksi objek yang baik dalam berbagai kondisi citra. Model terlatih kemudian berhasil diimplementasikan ke dalam aplikasi berbasis Python dengan antarmuka GUI menggunakan Tkinter, yang mendukung deteksi melalui gambar statis maupun secara *real-time* menggunakan *webcam*. Secara keseluruhan, penelitian ini menunjukkan bahwa penerapan algoritma YOLOv8 untuk pengenalan rambu lalu lintas lokal memiliki potensi yang kuat dalam mendukung sistem keselamatan berkendara, serta dapat menjadi dasar pengembangan lebih lanjut menuju sistem transportasi cerdas yang adaptif dan aplikatif di lingkungan nyata.

Daftar Pustaka

Arief, R. W. (2021). *Tesis Sistem Deteksi Dan Pengenalan Rambu Lalu Lintas Di Indonesia Menggunakan Algoritma Yolov4*. 1–68.

- Fadjeri, A. (2024). *Identifikasi Teks dari Citra Menggunakan Optical Character Recognition*. 2, 13–24.
- Fadjeri, A., Burhaein, E., Widiyono, I. P., Tri, R., & Khudaedi, D. (2025). *Pengembangan Teknologi AI berbasis Computer Vision dan Robotika untuk Latihan Akurasi Olahraga Tennis Lapangan Duduk bagi Atlet Disabilitas Fisik*. 1, 65–76.
- Fadjeri, A., & Hidayat, T. (2024). *Implementasi Sistem Informasi Absensi dan Nilai Berbasis Web di SMA Islam Al-Kahfi Somalangu Kebumen dengan Metode Waterfall*. 6(02), 595–611.
- Fatkhin, N., & Fadjeri, A. (2024). *Pembelajaran Mesin Untuk Deteksi Helm Keselamatan Menggunakan Algoritma YOLOv8*. 2, 77–86.
- Fitriana, A. D., & Fadjeri, A. (2025). *PENGEMBANGAN CHATBOT MENGGUNAKAN ALGORITMA NEURAL NETWORK UNTUK MENINGKATKAN PELAYANAN PELANGGAN DI TOKO*. 6(1), 298–311. <https://doi.org/10.46576/djtechno>
- IKBAL, M., & Saputra, R. A. (2024). *Pengenalan Rambu Lalu Lintas Menggunakan Metode YOLOv8*. *JIKA (Jurnal Informatika)*, 8(2), 204. <https://doi.org/10.31000/jika.v8i2.10609>
- Khafid, S. (2024). *Kota Jogja Masih Jadi Magnet Wisatawan*. *Harianjogja.Com*. <https://wisata.harianjogja.com/read/2024/09/08/504/1186974/kota-jogja-masih-jadi-magnet-wisatawan>
- Nurhuda, S. F. (2024). *Ada 152 Ribu Kecelakaan Lalu Lintas Selama Tahun 2*. *Detikcom*. https://oto.detik.com/berita/d-7688470/ada-152-ribu-kecelakaan-lalu-lintas-selama-tahun-2024-27-ribu-orang-tewas?utm_source=chatgpt.com
- Pauziah, R., & Herliana, A. (2021). *Implementasi Deteksi Rambu Lalu Lintas Menggunakan Metode Extreme Learning Machine*. *E-Prosiding Sistem Informasi*, x(x), 36–41.
- Pricillia, T., & Zulfachmi. (2021). *Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD)*. *Jurnal Bangkit Indonesia*, 10(1), 6–12. <https://doi.org/10.52771/bangkitindonesia.v10i1.153>
- Putri, H. J., & Murhayati, S. (2025). *Metode Pengumpulan Data Kualitatif*. 9, 13074–13086.
- Riansyah, A., & Mirza, A. H. (2023). *Pendeteksi Mobil Berdasarkan Merek dan Tipe Menggunakan Algoritma YOLO*. *Smatika Jurnal*, 13(01), 43–52. <https://doi.org/10.32664/smatika.v13i01.719>
- WHO, W. H. O. (2019). *Road safety*. World Health Organization: WHO. https://www.who.int/health-topics/road-safety#tab=tab_1