

© Jurnal Nasional Teknik Elektro dan Teknologi Informasi
Karya ini berada di bawah Lisensi Creative Commons Atribusi-BerbagiSerupa 4.0 Internasional
Terjemahan artikel 10.22146/jnteti.v14i2.18701

Pengembangan Sistem Deteksi Kantuk Menggunakan YOLOv9 untuk Keselamatan dalam Berkendara

Fernando Candra Yulianto¹, Wiwit Agus Triyanto¹, Syafiul Muzid¹

¹ Program Studi Sistem Informatika, Fakultas Teknik, Universitas Muria Kudus, Kudus, Jawa Tengah 59352, Indonesia

[Diserahkan: 22 Januari 2025, Revisi: 5 Maret 2025, Diterima: 17 April 2025]

Penulis Korespondensi: Fernando Candra Yulianto (email: 202153090@std.umk.ac.id)

INTISARI — Sistem deteksi kantuk memiliki peranan penting dalam keselamatan berkendara, mengingat tingginya angka kecelakaan lalu lintas yang banyak disebabkan oleh rasa kantuk. Beberapa sistem deteksi kantuk yang dibuat menggunakan metode *eye aspect ratio* (EAR), *percentage of eyelid closure* (PERCLOS), dan *convolutional neural network* (CNN) masih memiliki keterbatasan dalam hal akurasi dan waktu respons. Penelitian ini bertujuan untuk mengatasi permasalahan tersebut dengan menerapkan metode You Only Look Once versi 9 (YOLOv9). Metode ini memiliki keunggulan dalam hal kecepatan dan akurasi karena dapat mendeteksi objek secara *real-time* dalam satu tahap pemrosesan. *Dataset* dikumpulkan secara independen dari beberapa sumber yang merepresentasikan kondisi nyata di dalam kendaraan, mencakup berbagai kondisi pencahayaan dan sudut pandang. Setelah itu, dilakukan proses pelabelan, prapemrosesan, dan pemodelan. Kinerja model dievaluasi berdasarkan metrik presisi, *recall*, *F1-score*, dan *mean average precision* (mAP). Model terbaik dioptimasi menggunakan beberapa teknik optimasi guna mendapatkan hasil yang paling optimal. Hasil penelitian menunjukkan bahwa model YOLOv9 yang dilatih menggunakan optimasi *Nesterov-accelerated adaptive moment estimation* (Nadam) memiliki kecepatan pemrosesan citra yang lebih baik dibandingkan model lainnya. Model ini menghasilkan presisi, *recall*, *F1-score*, mAP@50, mAP@50, mAP@50-95, dan kecepatan pemrosesan masing-masing sebesar 99,4%, 99,6%, 99,5%, 99,5%, 85,5%, dan 52,08 FPS. Model yang dikembangkan dapat mendeteksi tanda-tanda kantuk pada pengemudi secara *real-time*, seperti mata tertutup, menguap, posisi kepala yang tidak normal, dan gerakan tangan yang tidak wajar. Namun, model ini masih memiliki keterbatasan dalam mendeteksi pengemudi yang mengenakan kacamata hitam, sehingga diperlukan pengembangan lebih lanjut untuk meningkatkan kinerjanya dalam kondisi tersebut.

KATA KUNCI — Keselamatan Berkendara, Sistem Deteksi Kantuk, YOLOv9, Deteksi Objek *Real-Time*.

I. PENDAHULUAN

Mengingat tingginya angka kecelakaan lalu lintas di berbagai negara yang tidak jarang mengakibatkan cedera serius atau bahkan kematian, keselamatan dalam berkendara menjadi salah satu aspek penting yang harus diperhatikan dalam kehidupan sehari-hari [1]. Salah satu faktor utama penyebab kecelakaan adalah rasa kantuk saat berkendara [2]. Berdasarkan data yang dilansir oleh National Highway Traffic Safety Administration, total kecelakaan fatal yang terjadi di Amerika Serikat pada tahun 2022 mencapai 693 kecelakaan [3]. Di Indonesia, sebuah studi tahun 2023 mengungkapkan bahwa 79% responden pernah setidaknya satu kali mengemudi dalam keadaan mengantuk dan 32% responden menyatakan hampir terlibat dalam kecelakaan fatal karena mengantuk saat mengemudi [4]. Data dari berbagai studi menunjukkan bahwa 20–30% kecelakaan lalu lintas disebabkan oleh pengemudi yang mengantuk [5]. Mengemudi dalam kondisi mengantuk sangat berbahaya karena rasa kantuk secara signifikan mengurangi kemampuan pengemudi untuk bereaksi dengan cepat dan tepat terhadap situasi darurat di jalan, seperti pengereman mendadak atau perubahan arah kendaraan yang tidak terduga. Selain itu, rasa kantuk dapat menyebabkan pengemudi kehilangan fokus dan konsentrasi, sehingga meningkatkan risiko kecelakaan.

Beberapa faktor dapat menjadi penyebab pengemudi mengantuk saat berkendara, sehingga meningkatkan risiko kecelakaan secara signifikan. Faktor pertama adalah kurang tidur, yang menyebabkan tubuh tidak dapat beristirahat dengan cukup. Hal ini mengakibatkan kelelahan pada otak dan tubuh, sehingga secara drastis menurunkan kemampuan untuk tetap

fokus dan waspada saat berkendara [6]. Faktor kedua adalah kelelahan akibat perjalanan jauh, terutama jika pengemudi melakukan perjalanan tanpa istirahat dengan cukup. Dalam kondisi seperti ini, otot menjadi tegang dan energi terkuras, sehingga meningkatkan kemungkinan timbulnya rasa kantuk [7]. Faktor terakhir adalah kondisi jalan yang monoton, seperti jalan raya lurus dan panjang dengan variasi yang minim, yang sering kali menyebabkan pengemudi kehilangan konsentrasi dan mengantuk karena kurangnya rangsangan visual [8]. Dengan kemajuan teknologi, berbagai upaya telah dilakukan untuk meminimalkan terjadinya kecelakaan yang disebabkan oleh rasa kantuk, salah satunya adalah pengembangan sistem deteksi kantuk pada pengemudi [9]. Deteksi dini kantuk dapat memberikan peringatan tepat waktu, sehingga pengemudi dapat mengambil tindakan pencegahan, seperti beristirahat sejenak untuk menghilangkan kantuknya. Namun, sebagian besar sistem deteksi kantuk yang ada masih memiliki keterbatasan dalam hal akurasi dan waktu respons. Metode You Look Only Once (YOLO) adalah pendekatan yang sangat populer di bidang visi komputer, terutama untuk deteksi objek waktu nyata [10].

Metode YOLO menawarkan kelebihan dalam hal kecepatan dan akurasi [11] karena dapat mendeteksi objek secara efisien dalam satu tahap, dibandingkan dengan metode deteksi lain yang biasanya memerlukan beberapa tahap pemrosesan [12]. Penelitian sebelumnya telah mengembangkan sistem berbasis visi komputer yang mengintegrasikan metrik *eye aspect ratio* (EAR) dan *percentage of eyelid closure* (PERCLOS) untuk mendeteksi kantuk. Sistem ini menggunakan kamera untuk memantau posisi mata pengemudi dan memicu alarm jika mata

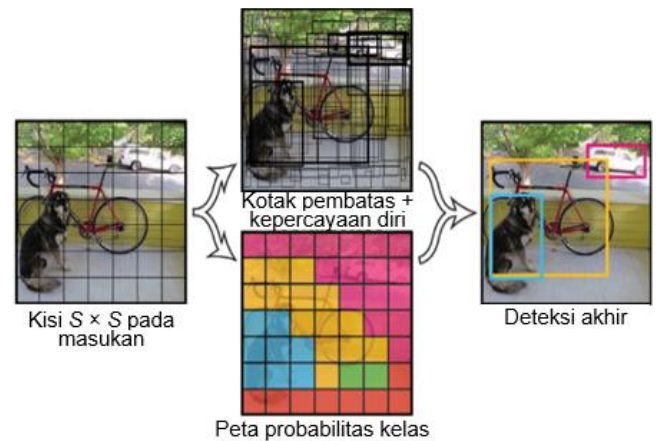
tertutup melebihi durasi yang telah ditentukan sebelumnya. Meskipun metode ini mencapai akurasi hingga 80%, kinerjanya dipengaruhi oleh kondisi pencahayaan. Selain itu, metode ini kurang adaptif terhadap variasi ambang batas EAR untuk masing-masing pengemudi, sehingga diperlukan penyesuaian manual untuk setiap pengemudi [13]. Selain itu, penelitian lain telah menggunakan metode *convolutional neural network* (CNN) dan menghasilkan akurasi sebesar 93% [14]. Akan tetapi, hasil deteksi ini sangat bergantung pada kualitas pencahayaan dan orientasi kamera ke arah wajah pengemudi.

Penelitian lain yang menggunakan metode YOLOv8 menunjukkan peningkatan yang signifikan. Dengan menggunakan *dataset* yang mencakup berbagai perilaku yang tidak terfokus dan kelelahan, YOLOv8 menunjukkan kinerja yang luar biasa dengan menghasilkan presisi sebesar 97,5%, *recall* sebesar 92,8%, *F1-score* sebesar 95,1%, dan *mean average precision* (mAP) sebesar 96,7% [15]. Sistem ini juga mampu mendeteksi kondisi secara *real-time* dengan konsistensi tinggi di berbagai kondisi pencahayaan. Oleh karena itu, metode YOLO dianggap sebagai pilihan ideal untuk diterapkan pada sistem deteksi kantuk yang diusulkan. Dalam penelitian ini, sistem dikembangkan menggunakan YOLOv9, yang menggabungkan beberapa versi, termasuk YOLOv9t, YOLOv9s, YOLOv9m, YOLOv9c, dan YOLOv9e, yang merupakan iterasi lanjutan dari YOLOv8. Model yang dibuat menggunakan YOLOv9 tersebut dievaluasi berdasarkan metrik presisi, *recall*, *F1-score*, dan mAP. Guna menentukan hasil yang paling optimal, model terbaik kemudian dioptimalkan menggunakan beberapa teknik optimasi, seperti AdamW, *stochastic gradient descent* (SGD), Adam, Adamax, *Nesterov-accelerated adaptive moment estimation* (Nadam), RAdam, dan RMSProp.

Model terbaik diimplementasikan menggunakan Streamlit [16], sehingga dapat diakses melalui platform web dan seluler. Sistem dilengkapi dengan fitur peringatan berupa suara peringatan yang akan aktif saat pengemudi menunjukkan tanda-tanda kantuk, seperti mata tertutup, menguap, posisi kepala tidak normal, dan gerakan tangan yang tidak biasa. Dengan peringatan ini, pengemudi diharapkan segera menghentikan kendaraannya dan beristirahat sejenak untuk mengurangi risiko kecelakaan. Sistem ini diharapkan dapat memberikan kontribusi yang signifikan untuk meningkatkan keselamatan dalam berkendara, terutama bagi pengemudi mobil pribadi yang rentan terhadap kelelahan dalam perjalanan jarak jauh.

II. METODOLOGI

YOLO merupakan salah satu metode deteksi objek *real-time* yang paling populer di bidang visi komputer [17]. Metode ini bekerja dengan membagi citra ke dalam kisi (kotak kecil) dan secara simultan mengidentifikasi objek di dalam kisi tersebut dalam satu tahap pemrosesan [18]. Ilustrasi cara kerja metode YOLO ditunjukkan pada Gambar 1. Gambar ini memberikan ilustrasi sederhana tentang cara kerja metode YOLO. Dalam metode ini, citra masukan dibagi menjadi kisi (misalnya 5×5). Setiap sel bertugas mendeteksi objek jika pusat objek berada di dalam sel tersebut. Untuk setiap sel, YOLO memprediksi beberapa kotak pembatas (*bounding box*) yang mencakup koordinat posisi (x, y), lebar, tinggi, dan skor kepercayaan diri (*confidence score*) yang menunjukkan kemungkinan kotak tersebut benar-benar mencakup objek. Selain itu, YOLO menghasilkan peta probabilitas kelas, yang mewakili probabilitas objek di dalam kotak tersebut termasuk



Gambar 1. Model membagi citra menjadi kisi $S \times S$, dengan setiap sel kisi memprediksi kotak pembatas, skor keyakinan untuk kotak tersebut, dan probabilitas kelas. Prediksi ini dikodekan sebagai tensor dengan dimensi $(S \times S \times (B \times 5 + C))$

dalam kategori tertentu (misalnya, anjing, sepeda, atau lainnya). Prediksi dari kotak pembatas dan probabilitas kelas kemudian digabungkan untuk mengidentifikasi objek yang terdeteksi. Kotak-kotak yang tidak relevan dieliminasi dengan menggunakan teknik *non-maximum suppression* (NMS) guna menghindari deteksi ganda, sehingga menghasilkan deteksi akhir yang lebih akurat. Dengan mengintegrasikan deteksi dan klasifikasi ke dalam satu tahap, YOLO menawarkan kecepatan dan efisiensi tinggi, menjadikannya ideal untuk aplikasi *real-time* [19]. Bagian selanjutnya menguraikan langkah-langkah dalam metodologi penelitian. Berikut adalah tahapan yang dilakukan dalam penelitian ini.

A. PENGUMPULAN DATASET

Dataset sebanyak 1.400 sampel dikumpulkan dari beberapa sumber. *Dataset* tersebut terdiri atas citra wajah pengemudi yang menunjukkan tanda-tanda mengantuk dan kondisi terjaga. Citra diambil pada berbagai kondisi pencahayaan dan dari berbagai sudut. Gambar 2 menunjukkan ilustrasi tanda-tanda pengemudi dalam kondisi mengantuk, yang meliputi mata tertutup, menguap, perubahan posisi kepala yang tidak normal, dan gerakan tangan yang tidak biasa. Sementara itu, Gambar 3 menunjukkan tanda-tanda pengemudi dalam kondisi terjaga.

B. PELABELAN DATASET

Pelabelan adalah proses berbasis *graphical user interface* (GUI) yang dilakukan untuk memberikan anotasi pada citra melalui penggunaan kotak pembatas, sehingga mempermudah pembuatan *dataset* berlabel dalam proyek visi komputer [20]. Dalam proses ini, *dataset* yang dikumpulkan diberi label berdasarkan dua kelas utama, yaitu mengantuk dan terjaga. Gambar 4 memperlihatkan hasil proses pelabelan *dataset* yang dikategorikan pada kelas terjaga, yang menunjukkan bahwa pengemudi tidak dalam keadaan mengantuk. Sementara itu, Gambar 5 menampilkan kelas mengantuk, yang menunjukkan tanda-tanda rasa kantuk pada pengemudi.

C. PRAPEMROSESAN DATASET

Pada tahap ini, ukuran citra diubah menjadi 320 piksel. Algoritma YOLO digunakan untuk mendeteksi dan mengisolasi area wajah utama, seperti mata, mulut, dan kepala. Proses ini menghasilkan kotak pembatas yang fokus pada area-area penting yang dapat menunjukkan tanda-tanda adanya rasa kantuk. Algoritma YOLO diterapkan pada tahap awal ini untuk membantu deteksi dini objek tertentu yang menunjukkan rasa kantuk. Oleh karena itu, pada fase pelatihan selanjutnya, model



Gambar 2. Tanda-tanda kondisi mengantuk.



Gambar 3. Tanda-tanda kondisi terjaga.

dapat fokus pada bagian-bagian yang relevan untuk proses klasifikasi.

D. PEMODELAN YOLOV9

Algoritma YOLOv9, yang meliputi YOLOv9t, YOLOv9s, YOLOv9m, YOLOv9c, dan YOLOv9e, digunakan pada tahap pengembangan model ini [21]. Setiap versi dilatih menggunakan *dataset* yang sama dan diuji pada data uji untuk mendapatkan hasil evaluasi.

E. EVALUASI MODEL

Model yang telah dilatih dievaluasi menggunakan metrik seperti presisi, *recall*, *F1-score*, mAP, dan kecepatan pemrosesan. Pengujian dilakukan pada berbagai kondisi pencahayaan, posisi kamera, dan perspektif pengemudi. Berikut ini adalah penjelasan dari setiap metrik yang digunakan.

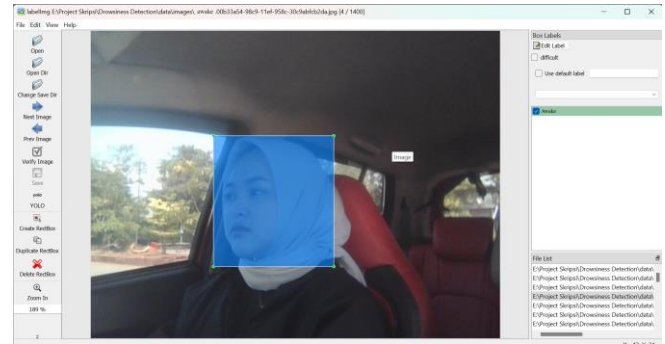
1) PRESISI

Presisi mengukur akurasi prediksi positif yang benar (*true positive*) relatif terhadap semua prediksi positif (*true positive + false positive*). Metrik ini digunakan untuk mengevaluasi akurasi sistem dalam mendeteksi objek yang benar [22].

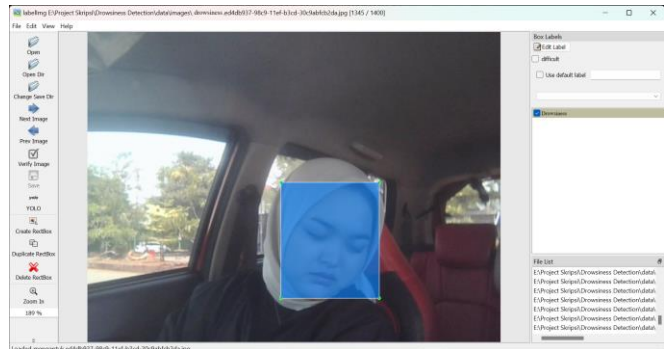
$$P = \frac{TP}{TP+FP} \quad (1)$$

2) RECALL

Recall mengukur tingkat kemampuan sistem dalam mendeteksi semua objek yang harus diidentifikasi dengan membandingkan jumlah positif benar dengan jumlah total objek relevan (*true positive + false negative*) [23].



Gambar 4. Proses pelabelan kondisi terjaga.



Gambar 5. Proses pelabelan kondisi mengantuk.

$$R = \frac{TP}{TP+FN} \quad (2)$$

3) F1-SCORE

F1-score adalah rata-rata antara presisi dan *recall*, yang menyeimbangkan keduanya [24].

$$F1 = \frac{2 \times P \times R}{P+R} \quad (3)$$

4) MEAN AVERAGE PRECISION

mAP adalah nilai rata-rata *average precision* (AP) untuk setiap kelas dalam *dataset*, dengan Q mewakili jumlah kelas dan $AP(q)$ mewakili AP untuk kelas q . Metrik ini mengevaluasi kinerja keseluruhan model di semua kelas [25].

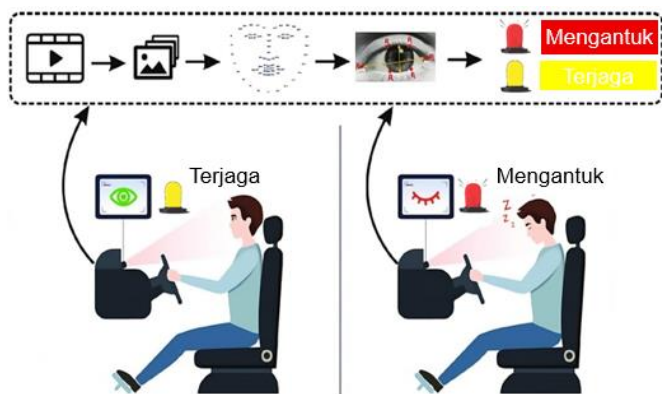
$$mAP = \frac{1}{Q} \sum_{q=1}^Q AP(q) \quad (4)$$

F. IMPLEMENTASI MODEL

Pada tahap akhir, model dipilih berdasarkan hasil metrik terbaik dari berbagai versi YOLOv9 yang telah diterapkan sebelumnya. Model yang dipilih diimplementasikan ke dalam sistem deteksi kantuk sederhana menggunakan Streamlit. Sistem dilengkapi dengan fitur peringatan berupa bunyi alarm saat pengemudi terdeteksi mengantuk.

III. HASIL DAN PEMBAHASAN

Gambar 6 menunjukkan desain sistem deteksi kantuk yang dikembangkan. Sistem ini dirancang untuk mengidentifikasi tanda-tanda kantuk pada pengemudi, seperti mata tertutup, menguap, posisi kepala tidak normal, dan gerakan tangan yang tidak biasa. Ketika sistem mendeteksi adanya tanda-tanda tersebut pada pengemudi, secara otomatis sistem mengeluarkan suara peringatan. Sebaliknya, jika tidak ada tanda-tanda kantuk yang terdeteksi, sistem tidak akan mengeluarkan peringatan apa pun. Sistem ini dirancang untuk diimplementasikan menggunakan kerangka kerja Streamlit, yang memungkinkan pengembangan antarmuka yang interaktif dan ramah pengguna. Keluaran akhir dari sistem ini adalah aplikasi yang



Gambar 6. Desain sistem deteksi kantuk.

dapat diakses melalui perangkat seluler dan situs web, sehingga memungkinkan pengguna memanfaatkan sistem tanpa perlu membeli perangkat keras tambahan. Hal ini menjadikannya solusi praktis dan hemat biaya untuk berbagai macam pengguna. Pembahasan berikutnya menjelaskan hasil evaluasi model, yang sebelumnya dilatih menggunakan metode YOLOv9. Evaluasi ini mencakup analisis kinerja deteksi dalam berbagai kondisi, seperti kondisi pencahayaan dan sudut kamera yang beragam, untuk memastikan efektivitas sistem dalam mendeteksi tanda-tanda kantuk secara *real-time*.

A. EVALUASI MODEL

Setelah melalui serangkaian tahapan sebelumnya, proses pelatihan dilakukan dengan menerapkan algoritma YOLOv9 selama 20 *epoch* menggunakan optimasi AdamW. Hal ini dilakukan karena, secara *default*, metode YOLO menggunakan optimasi ini, yaitu nilai laju belajar (*learning rate*) sebesar 0,001667 dan momentum sebesar 0,9. Proses pelatihan dilakukan pada laptop AMD Ryzen 5 4500U dengan Radeon Graphics. Hasil proses pelatihan disajikan pada Tabel I dan Tabel II.

Berdasarkan hasil evaluasi model YOLOv9 yang disajikan pada Tabel I dan Tabel II, dapat disimpulkan bahwa model YOLOv9t merupakan pilihan terbaik untuk penelitian ini. Pemilihan ini mempertimbangkan keterbatasan perangkat keras yang digunakan, yaitu laptop yang ditenagai prosesor AMD Ryzen 5 4500U dengan Radeon Graphics. Model YOLOv9t menunjukkan kinerja deteksi yang sangat baik, dengan nilai *presisi*, *recall*, dan *F1-score* sebesar 99,4%, *mAP@50* sebesar 99,5%, dan *mAP@50-95* sebesar 84,8%. Hasil ini menunjukkan bahwa kinerja YOLOv9t sebanding dengan model yang lebih kompleks, sambil tetap mempertahankan efisiensi yang jauh lebih baik. Salah satu kelebihan model ini adalah efisiensi komputasinya, yang ditunjukkan oleh nilai GFLOPs terendah sebesar 6,4 dan jumlah parameter yang relatif ringan, yaitu sebanyak 1.730.214.

Hal tersebut menjadikan YOLOv9t sebagai model hemat energi yang dapat meminimalkan kebutuhan komputasi, sehingga sangat sesuai untuk diaplikasikan pada perangkat dengan spesifikasi terbatas. Selain itu, kecepatan inferensi model ini mencapai 52,08 FPS, yang mendukung penggunaannya pada inferensi *real-time*, terutama pada perangkat keras dengan daya komputasi terbatas, seperti laptop. Dengan demikian, model YOLOv9t menawarkan keseimbangan optimal antara kinerja deteksi dan efisiensi komputasi, sehingga menjadikannya pilihan ideal untuk digunakan pada sistem yang dikembangkan. Pada tahap

TABEL I
PERBANDINGAN BERBAGAI MODEL YOLOv9

Model	Presisi	Recall	F1-Score	mAP@50
YOLOv9t	99,4	99,4	99,4	99,5
YOLOv9s	99,6	99,4	99,5	99,5
YOLOv9m	99,6	99,5	99,6	99,5
YOLOv9c	99,4	99,5	99,4	99,5
YOLOv9e	99,2	99,5	99,3	99,4

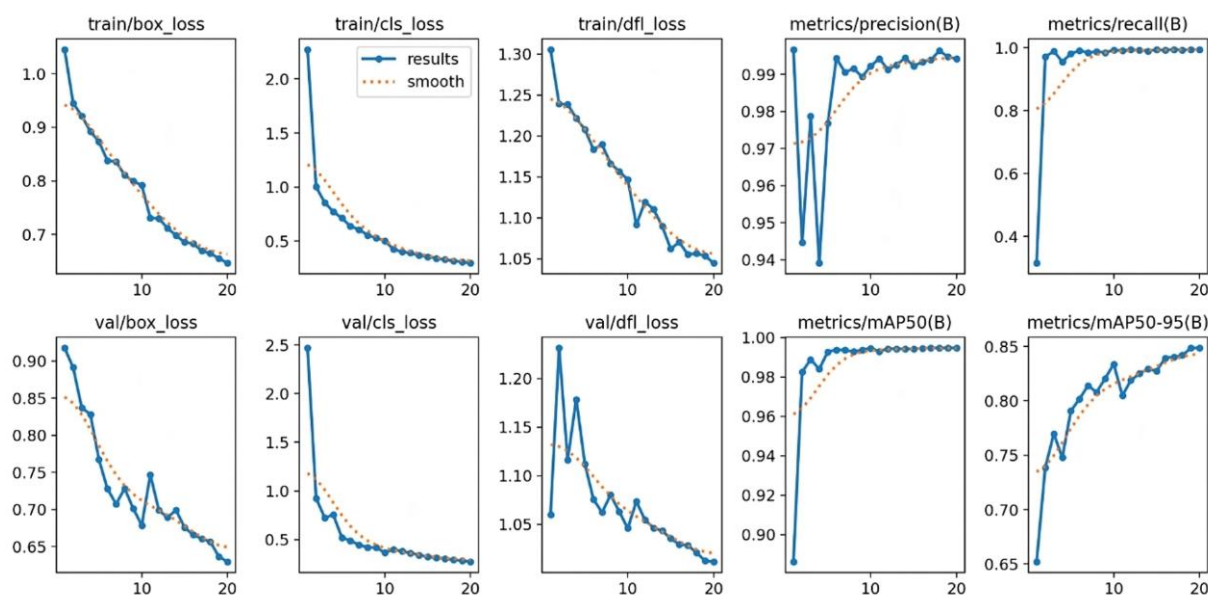
TABEL II
PERBANDINGAN LANJUTAN BERBAGAI MODEL YOLOv9

Model	mAP50-95	GFLOPs	Parameter	FPS
YOLOv9t	84,8	6,4	1.730.214	52,08
YOLOv9s	85,2	22,1	6.194.422	17,73
YOLOv9m	85,5	60,0	16.576.438	7,41
YOLOv9c	84,8	82,7	21.146.966	7,33
YOLOv9e	85,5	169,5	53.204.118	2,96

berikutnya, penelitian ini difokuskan pada implementasi model YOLOv9t ke dalam sistem yang dirancang untuk mendukung deteksi kantuk secara *real-time*. Gambar 7 menyajikan kinerja pelatihan dan validasi model YOLOv9t. Berdasarkan hasil grafik pelatihan dan validasi model YOLOv9t menggunakan optimasi AdamW yang ditunjukkan pada Gambar 7, tren kerugian secara konsisten menurun seiring dengan bertambahnya jumlah *epoch*, baik pada saat pelatihan maupun validasi. Hal ini menunjukkan bahwa model YOLOv9t secara progresif meningkatkan kemampuan pembelajarannya dan secara bertahap dapat meminimalkan kesalahan prediksi. Grafik metrik evaluasi, seperti *presisi*, *recall*, *mAP@0.5*, dan *mAP@0.5-0.95*, menunjukkan adanya peningkatan stabil yang mendekati nilai maksimum. Hasil ini menegaskan bahwa model tersebut memiliki kemampuan deteksi objek yang sangat baik, baik dalam hal akurasi maupun generalisasi ke data validasi. Namun, terjadi sedikit fluktuasi selama tahap awal pelatihan metrik validasi, termasuk *presisi*, *recall*, *mAP@0.5*, dan *mAP@0.5-0.95*. Fluktuasi yang terjadi merupakan hal alami yang mencerminkan proses adaptasi model terhadap pola data yang kompleks sebelum akhirnya mencapai konvergensi yang stabil. Secara keseluruhan, grafik menunjukkan bahwa model YOLOv9t yang digunakan dalam studi ini mampu menunjukkan kinerja yang optimal, dengan keseimbangan antara tingkat akurasi tinggi dan kemampuan generalisasi yang baik. Temuan ini mengindikasikan bahwa model tersebut dapat diimplementasikan lebih lanjut pada sistem deteksi kantuk *real-time*.

Pada tahap selanjutnya, hasil evaluasi model YOLOv9t yang sebelumnya dilatih menggunakan algoritma optimasi AdamW dibandingkan dengan beberapa algoritma optimasi lainnya, seperti SGD, Adam, Adamax, Nadam, Radam, dan RMSProp. Perbandingan ini bertujuan untuk mengidentifikasi metode optimasi yang paling efektif dalam meningkatkan akurasi dan efisiensi model, khususnya dalam proses deteksi kantuk.

Hasil evaluasi pada Tabel III menunjukkan bahwa teknik optimasi yang digunakan dalam pelatihan model YOLOv9 berdampak signifikan pada kinerjanya, khususnya dalam hal kecepatan konvergensi dan kemampuan generalisasi. Berdasarkan hasil evaluasi, *optimizer* seperti AdamW, Adam, Adamax, Nadam, dan RAdam menunjukkan kinerja yang lebih unggul dibandingkan dengan *optimizer* lain seperti SGD dan RMSProp. Hal ini disebabkan oleh kemampuan Adam dan variannya yang secara otomatis dapat menyesuaikan laju



Gambar 7. Pelatihan dan kinerja validasi model YOLOv9t menggunakan optimisasi AdamW.

TABEL III
PERBANDINGAN YOLOV9T DENGAN OPTIMIZER BERBEDA

Optimizer	Presisi	Recall	F1-Score	mAP @0.5	mAP @0.5-0.9
AdamW	99,4	99,4	99,4	99,5	84,8
SGD	99,4	99,4	99,4	99,4	83,6
Adam	99,6	99,4	99,5	99,5	85,4
Adamax	99,5	99,5	99,5	99,5	85,4
NAdam	99,4	99,6	99,5	99,5	85,5
RAdam	99,6	99,4	99,5	99,5	84,3
RMSProp	98,8	98,1	98,5	99,4	82,5

belajar berdasarkan momen pertama (rata-rata) dan momen kedua (varians) gradien. Adaptasi ini memungkinkan pembaruan parameter yang lebih stabil dan lebih cepat, sehingga model dapat mencapai konvergensi secara efisien, bahkan dalam jumlah periode yang terbatas seperti dalam proses pelatihan ini.

Sebaliknya, *optimizer* seperti SGD, yang hanya mengandalkan gradien langsung tanpa mekanisme momentum atau laju belajar adaptatif, cenderung memerlukan lebih banyak *epoch* untuk mencapai hasil optimal. Hal ini menyebabkan kinerjanya kurang memadai jika digunakan pada *dataset* ini. Meskipun RMSProp dirancang untuk menangani gradien yang tidak stabil, RMSProp menunjukkan kemampuan generalisasi yang lebih rendah pada model ini. Hal ini dapat disebabkan karena penyetelan *hyperparameter* yang kurang optimal atau ketergantungannya pada kondisi data tertentu. Di sisi lain, Adam dan variannya unggul karena stabilitas, kemampuan beradaptasi, dan efektivitasnya dalam mengelola gradien kompleks, sehingga menjadikannya pilihan yang paling sesuai untuk melatih model YOLOv9 pada *dataset* deteksi objek dengan dua kelas utama, yaitu mengantuk dan terjaga. Pembahasan berikutnya difokuskan pada penggunaan model YOLOv9t dengan *optimizer* Nadam. Tabel III menunjukkan bahwa Nadam mencapai akurasi yang sedikit lebih baik dibandingkan dengan *optimizer* turunan Adam lainnya. Gambar 8 mengilustrasikan kinerja pelatihan dan validasi model YOLOv9t menggunakan *optimizer* Nadam.

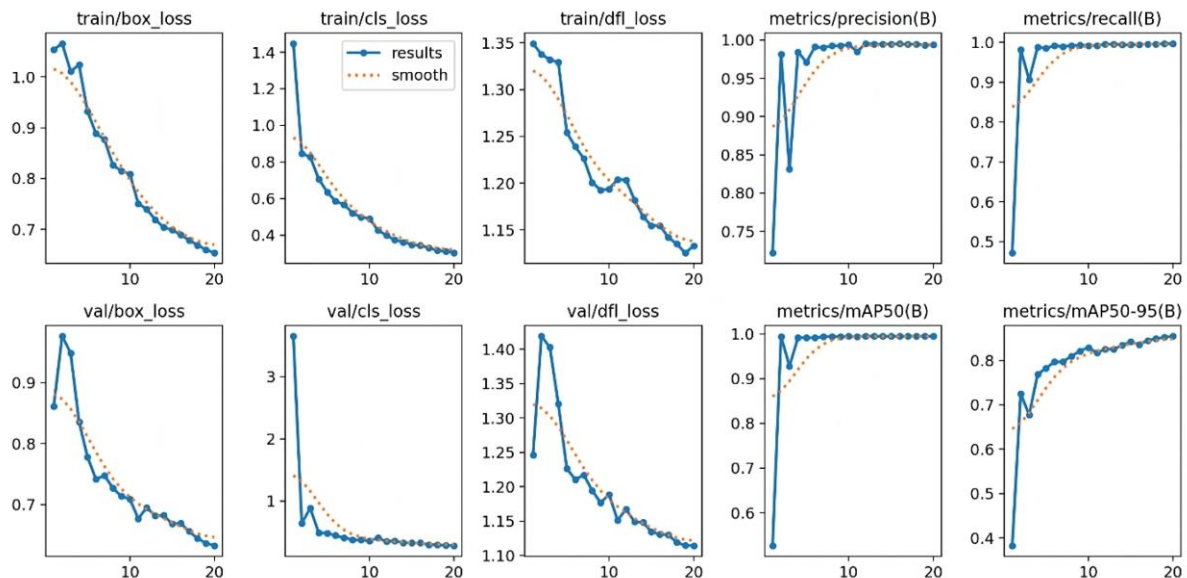
Hasil evaluasi model YOLOv9t dengan optimisasi Nadam (Gambar 8) menunjukkan peningkatan kinerja dibandingkan

dengan hasil yang diperoleh menggunakan optimisasi AdamW (Gambar 7). Peningkatan kinerja ini terjadi karena Nadam menggabungkan momentum Nesterov dengan Adam, yang memungkinkan model melihat estimasi gradien masa depan sebelum pembaruan parameter, sehingga dapat mempercepat konvergensi dan meningkatkan stabilitas pelatihan. AdamW lebih berfokus pada regularisasi dengan peluruhan bobot (*weight decay*) guna mengurangi *overfitting*, sedangkan Nadam menunjukkan keunggulannya dalam eksplorasi optimisasi karena momentum Nesterov membantu menghindari jebakan lokal. Oleh karena itu, Nadam menghasilkan nilai mAP@0.5-0.9 yang lebih tinggi (85,5) daripada AdamW (84,8), yang mengindikasikan bahwa deteksi ini memiliki tingkat akurasi lebih tinggi dan generalisasi yang lebih baik dalam berbagai tingkat kesulitan objek. Meskipun demikian, kedua grafik optimisasi masih menunjukkan terjadinya fluktuasi pada tahap awal pelatihan. Dalam pembahasan berikut, model ini akan diimplementasikan menggunakan Streamlit untuk pengujian lebih lanjut.

B. IMPLEMENTASI MODEL

Pada tahap ini, model YOLOv9t yang dilatih menggunakan algoritma optimisasi Nadam dipilih untuk diimplementasikan ke dalam sistem dengan memanfaatkan kerangka Streamlit. Penggunaan kerangka Streamlit didasarkan pada sifatnya yang *open-source* serta kemudahannya dalam membangun aplikasi web yang interaktif dan intuitif. Streamlit secara khusus dirancang untuk mendukung kebutuhan di bidang ilmu data dan pembelajaran mesin, dengan kemampuan untuk membuat antarmuka pengguna yang menarik dan informatif hanya dengan beberapa baris kode Python. Salah satu keunggulan utamanya adalah kemampuannya untuk mendukung visualisasi data, eksplorasi model pembelajaran mesin, serta penyajian hasil analisis dalam format yang dapat dengan mudah dipahami oleh pengguna teknis dan nonteknis. Dengan memanfaatkan Streamlit, proses penyampaian hasil analisis menjadi lebih efisien dan efektif [26].

Dalam konteks sistem deteksi kantuk ini, Streamlit merupakan pilihan yang ideal karena memungkinkan integrasi model pembelajaran mesin yang kompleks dengan antarmuka pengguna yang sederhana tetapi fungsional. Hal ini



Gambar 8. Pelatihan dan kinerja validasi model YOLOv9t menggunakan optimisasi Nadam.



Gambar 9. Implementasi model YOLOv9t dengan optimisasi Nadam.

memudahkan *end user* untuk menggunakan sistem tanpa perlu memahami detail teknis implementasi model. Streamlit juga mendukung fleksibilitas pengembangan, seperti penyesuaian antarmuka atau penambahan fitur lain sesuai dengan kebutuhan pengguna. Berdasarkan keunggulan-keunggulan tersebut, kerangka kerja ini dipilih sebagai komponen inti dalam pengembangan sistem.

Gambar 9 menyajikan hasil implementasi model YOLOv9t yang dilatih dengan optimisasi Nadam. Hasil tersebut menunjukkan bahwa model mampu mendeteksi tanda-tanda kantuk dan terjaga dengan kinerja yang sangat baik. Model tersebut menunjukkan akurasi yang konsisten di berbagai pengujian, termasuk pada kondisi pencahayaan yang bervariasi. Setiap prediksi menghasilkan skor kepercayaan diri dengan rata-rata sekitar 0,88. Skor ini dianggap sangat baik karena skor kepercayaan diri yang mendekati 1 mengindikasikan kepastian yang lebih tinggi dalam prediksi model. Skor kepercayaan diri yang tinggi menunjukkan bahwa model tersebut dapat memberikan prediksi yang andal dalam skenario dunia nyata. Model tersebut juga menunjukkan kecepatan pemrosesan citra yang sangat tinggi, yaitu sebesar 52,08 FPS. Dengan kemampuannya untuk secara konsisten mendeteksi tanda-tanda kantuk dan terjaga dalam berbagai kondisi pencahayaan, model tersebut menunjukkan kinerja

yang unggul. Namun, model ini masih memiliki keterbatasan dalam mendeteksi pengemudi yang mengenakan kacamata hitam, sehingga pengembangan lebih lanjut diperlukan untuk meningkatkan kinerjanya dalam kondisi ini.

IV. KESIMPULAN

Hasil penelitian menunjukkan bahwa model YOLOv9t yang dilatih menggunakan optimisasi Nadam dapat bekerja secara efisien dan akurat pada perangkat keras dengan daya komputasi terbatas. Berdasarkan hasil evaluasi, model ini terbukti menjadi pilihan yang optimal karena keseimbangan antara kinerja dan efisiensi yang sangat baik. Model tersebut mencapai metrik deteksi yang tinggi, dengan presisi sebesar 99,4%, *recall* sebesar 99,6%, *F1-score* dan nilai mAP@50 sebesar 99,5%, dan nilai mAP@50-95 sebesar 85,5%, GFLOP terendah 6,4, dan jumlah parameter paling ringan adalah 1.730.214. Kecepatan inferensi sebesar 52,08 FPS makin menggarisbawahi kesesuaiannya untuk aplikasi *real-time*. Selama pengujian, model tersebut menunjukkan kinerja yang kuat dalam mendeteksi tanda-tanda kantuk dan terjaga dalam berbagai kondisi pencahayaan.

Namun, hasil grafik kinerja pelatihan dan validasi pada model tersebut masih memiliki kendala yang sama, yaitu masih menunjukkan fluktuasi pada tahap awal pelatihan, yang mengindikasikan adanya ruang untuk penyesuaian *hyperparameter*. Diperlukan *dataset* yang lebih beragam, penerapan augmentasi data yang tepat, dan proses pelatihan yang lebih lama untuk mendapatkan kinerja model yang lebih baik. Selain itu, model ini masih memiliki keterbatasan dalam mendeteksi pengemudi yang memakai kacamata hitam, sehingga diperlukan pengembangan lebih lanjut guna meningkatkan kinerjanya pada kondisi tersebut. Secara keseluruhan, penelitian ini difokuskan pada pentingnya keseimbangan antara akurasi, efisiensi komputasi, dan kinerja, terutama untuk aplikasi pada perangkat dengan sumber daya terbatas. Penelitian selanjutnya dapat difokuskan pada pengoptimalan generalisasi model yang telah dibuat.

KONFLIK KEPENTINGAN

Penulis menyatakan bahwa tidak ada konflik kepentingan dalam pelaksanaan penelitian ini atau dalam persiapan makalah ini.

KONTRIBUSI PENULIS

Konseptualisasi, Fernando Candra Yulianto, Wiwit Agus Triyanto, dan Syaiful Muzid; metodologi, Fernando Candra Yulianto dan Wiwit Agus Triyanto; perangkat lunak, Fernando Candra Yulianto and Syaiful Muzid; validasi, Fernando Candra Yulianto; analisis formal, Wiwit Agus Triyanto and Syaiful Muzid; investigasi, Fernando Candra Yulianto; sumber daya, Fernando Candra Yulianto; kurasi data, Fernando Candra Yulianto; penulisan—persiapan draf asli, Fernando Candra Yulianto; penulisan—peninjauan dan penyuntingan, Fernando Candra Yulianto, Wiwit Agus Triyanto, dan Syaiful Muzid; visualisasi, Fernando Candra Yulianto; pengawasan, Wiwit Agus Triyanto dan Syaiful Muzid; administrasi proyek, Wiwit Agus Triyanto.

REFERENSI

- [1] W.U. Puspoprodo dan N.N. Laila, "Studi pemahaman dan perilaku keselamatan berkendara (safety riding) pada remaja dan usia produktif di Pulau Jawa," *J. Ilm. Kesehat.*, vol. 20, no. 3, hal. 118–126, Des. 2021, doi: 10.33221/jikes.v20i3.1480.
- [2] A.A.M. Suradi dkk., "Sistem deteksi kantuk pengemudi mobil berdasarkan analisis rasio mata menggunakan computer vision," *JUKI J. Komput. Inform.*, vol. 5, no. 2, hal. 222–230, Nov. 2023, doi: 10.53842/juki.v5i2.269.
- [3] National Center for Statistics and Analysis, "Overview of motor vehicle traffic crashes in 2022," Nat. Highw. Traffic Saf. Adm., Washington, D.C., USA, DOT HS 813 560, 2024.
- [4] R. Rahmadiyah dan A. Widyanti, "Prevalence of drowsy driving and modeling its intention: An Indonesian case study," *Transp. Res. Interdiscip. Perspect.*, vol. 19, Mei 2023, Art. no 100824, doi: 10.1016/j.trip.2023.100824.
- [5] S.S. Soleimanloo dkk., "The association of schedule characteristics of heavy vehicle drivers with continuous eye-blink parameters of drowsiness," *Transp. Res. Part F, Traffic Psychol. Behaviour*, vol. 90, hal. 485–499, Okt. 2022, doi: <https://doi.org/10.1016/j.trf.2022.08.019>.
- [6] B. Subekti, "Penanganan kecelakaan lalu lintas dalam perspektif kebijakan publik," *Indones. J. Edu. Humanit.*, vol. 2, no. 4, hal. 144–151, Nov. 2022.
- [7] R.S. Kamila, "Faktor yang berhubungan dengan kelelahan kerja pada pengemudi bus di PT X tahun 2022," Skripsi, UIN Syarif Hidayatullah, Jakarta, Indonesia, 2022.
- [8] R. Aprianto, A. Rokhim, A. Basuki, dan S. Sugiyarto, "Pengaruh karakteristik pengemudi dan pemanfaatan rest area terhadap kelelahan pengemudi studi kasus Ruas Jalan Tol Pejagan-Solo," *J. Keselam. Transp. Jalan (Indones. J. Road Saf.)*, vol. 8, no. 1, hal. 92–103, Jun. 2021, doi: 10.46447/ktj.v8i1.310.
- [9] M.Y. Petra dan L. Lina, "Sistem pendeteksi kantuk pada mata pengemudi roda empat dengan algoritme convolutional neural network," *Computatio J. Comp. Sci. Inf. Syst.*, vol. 8, no. 1, hal. 20–28, Apr. 2024, doi: 10.24912/computatio.v8i1.16220.
- [10] R. Rahmania dkk., "Exploration of the impact of kernel size for YOLOv5-based object detection on quadcopter," *JOIV, Int. J. Inform. Visualization*, vol. 6, no. 3, hal. 726–735, Sep. 2022, doi: 10.30630/joiv.6.3.898.
- [11] B.G. Amira, M.M. Zoulikha, dan P. Hector, "Driver drowsiness detection and tracking based on YOLO with Haar cascades and ERNN," *Int. J. Saf. Secur. Eng.*, vol. 11, no. 1, hal. 35–42, Feb. 2021, doi: 10.18280/ijssse.110104.
- [12] A.Y. Alin, K. Kusri, dan K.A. Yuana, "The effect of data augmentation in deep learning with drone object detection," *IJCCS (Indones. J. Comput. Cybern. Syst.)*, vol. 17, no. 3, hal. 237–248, Jul. 2023, doi: 10.22146/ijccs.84785.
- [13] R. Thakur, Shivam, S. Raj, dan S. Pandey, "Driver drowsiness detection system using machine learning," dalam *Advances in Transdisciplinary Engineering*, R.M. Singari, P.K. Kankar, Eds., Amsterdam, Belanda: IOS Press BV, 2022, hal. 31–38.
- [14] H.V. Chand dan J. Karthikeyan, "CNN based driver drowsiness detection system using emotion analysis," *Intell. Automat. Soft Comput.*, vol. 31, no. 2, hal. 717–728, Sep. 2022, doi: 10.32604/iasc.2022.020008.
- [15] A. Debsi dkk., "Driver distraction and fatigue detection in images using ME-YOLOv8 algorithm," *IET Intell. Transp. Syst.*, vol. 18, no. 10, hal. 1910–1930, Okt. 2024, doi: 10.1049/itr2.12560.
- [16] J. Surya dan F.H. Aminuddin, *Pemrograman MYSQL Database With Streamlit Python*. Jambi, Indonesia: PT. Sonpedia Publishing Indonesia, 2024.
- [17] D.N. Alfarizi dkk., "Penggunaan metode YOLO pada deteksi objek: Sebuah tinjauan literatur sistematis," *J. Artif. Intel. Sist. Penunjang Keputusan*, vol. 1, no. 1, hal. 54–63, Jun. 2023.
- [18] A. Nurhopipah, I.R. Azziz, dan J. Suhaman, "Behind the mask: Detection and recognition based-on deep learning," *IJCCS (Indones. J. Comput. Cybern. Syst.)*, vol. 16, no. 1, hal. 67–78, Jan. 2022, doi: 10.22146/ijccs.72075.
- [19] W.A. Triyanto, K. Adi, dan J.E. Suseno, "Detection and tracking of broiler flock movements in the chicken coop using YOLO," dalam *8th Int. Conf. Energy Environ. Epidemiol. Inf. Syst. (ICENIS 2023)*, dalam E3S Web of Conferences, vol. 448, 2023, hal. 1–13.
- [20] M.T.A. Dipu, S.S. Hossain, Y. Arafat, dan F.B. Rafiq, "Real-time driver drowsiness detection using deep learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 7, hal. 844–850, Jul. 2021, doi: 10.14569/IJACSA.2021.0120794.
- [21] Ultralytics, "YOLOv9: A Leap Forward in Object Detection Technology." Tanggal akses: 14-Jan-2025. [Online]. Tersedia: <https://docs.ultralytics.com/models/yolov9/>
- [22] M.F. Ridho, F. Panca, W. Yandi, dan A.A. Rachmani, "Drowsiness detection in the advanced driver-assistance system using YOLO V5 detection model," *Electron. J. Ilm. Tek. Elekt.*, vol. 5, no. 1, hal. 130–137, Mei 2024, doi: 10.33019/electron.v5i1.136.
- [23] Z. Wang, K. Yao, dan F. Guo, "Driver attention detection based on improved YOLOv5," *Appl. Sci.*, vol. 13, no. 11, Jun.-1 2023, Art. no 6645, doi: 10.3390/app13116645.
- [24] S. Liu dkk., "A driver fatigue detection algorithm based on dynamic tracking of small facial targets using YOLOv7," *IEICE Trans. Inf. Syst.*, vol. E106-D, no. 11, hal. 1881–1890, 2023, doi: 10.1587/transinf.2023EDP7093.
- [25] Y. Du, X. Liu, Y. Yi, dan K. Wei, "Optimizing road safety: Advancements in lightweight YOLOv8 models and GhostC2f design for real-time distracted driving detection," *Sensors*, vol. 23, no. 21, Nov.-1 2023, Art. no 8844, doi: 10.3390/s23218844.
- [26] A.A. Permana, M.R. Muttaqin, dan M.A. Sunandar, "Sistem deteksi api secara real time menggunakan algoritme You Only Look Once (YOLO) versi 8," *Jati (J. Mhs. Tek. Inform.)*, vol. 8, no. 5, hal. 10395–10400, Okt. 2024, doi: 10.36040/jati.v8i5.10847.