



Deteksi Dini Microsleep pada Pengemudi Kendaraan Roda Empat Melalui Citra Mata Menggunakan Algoritma YOLOv8

Rhadis Steffani Saputri^{1*}, Aulia Apriliani², Rizky Syahrul Amar³, Lola Yorita Astri⁴

^{1, 2, 3}Program Studi Teknik Informatika, Universitas Dinamika Bangsa, Jl. Jenderal Sudirman, Thehok, Jambi 36138, Indonesia

⁴Program Studi Sistem Komputer, Universitas Dinamika Bangsa, Jl. Jenderal Sudirman, Thehok, Jambi 36138, Indonesia

* Penulis Korespondensi, Email: rhadisstf@email.com

Abstrak—Microsleep merupakan kondisi tidur singkat yang terjadi secara tiba-tiba dan tidak disadari, terutama saat mengemudi, yang merupakan salah satu penyebab utama tingginya angka kecelakaan di jalan raya. Penelitian ini bertujuan mengembangkan sistem deteksi dini microsleep pada pengemudi kendaraan roda empat menggunakan algoritma YOLO v8. Sistem bekerja dengan mendeteksi citra mata secara real-time melalui kamera, dan memberikan peringatan berupa suara dan visual apabila mata tertutup selama lebih dari 3 detik. Dataset yang digunakan adalah SleepyDetect dari Roboflow Universe, yang terdiri dari dua kelas sesuai kebutuhan sistem yaitu Open-eye dan Closed-eye. Model dilatih menggunakan YOLO v8n. Evaluasi menunjukkan hasil yang sangat baik dengan precision 97.2%, recall 96.7%, F1-score 0.97, dan mAP@0.5 sebesar 97.9%. Sistem juga berhasil diimplementasikan dan diuji dalam berbagai kondisi pencahayaan, serta menunjukkan respons deteksi yang cepat dan akurat. Penelitian ini menunjukkan bahwa kombinasi computer vision dan YOLO v8 dapat diterapkan secara efektif untuk meningkatkan keselamatan pengemudi melalui deteksi dini microsleep.

Kata Kunci: Microsleep; Deep Learning; Computer Vision; Deteksi Objek; YOLOv8; Deteksi Waktu Nyata; Deteksi Citra Mata.

Abstract—Microsleep is a brief and sudden episode of sleep that occurs without conscious awareness, especially during driving, and is recognized as one of the leading causes of road traffic accidents. This study aims to develop an early detection system for microsleep in four-wheeled vehicle drivers using the YOLOv8 algorithm. The system operates by detecting eye conditions in real-time through a camera and issues both audible and visual warnings when the eyes remain closed for more than 3 seconds. The dataset used is SleepyDetect from Roboflow Universe, consisting of two relevant classes: Open-eye and Closed-eye. The model was trained using YOLOv8n. Evaluation results demonstrate excellent performance, with a precision of 97.2%, recall of 96.7%, F1-score of 0.97, and mAP@0.5 of 97.9%. The system was successfully implemented and tested under various lighting conditions, showing fast and accurate detection responses. This research shows that the combination of computer vision and YOLOv8 can be effectively applied to enhance driver safety through early microsleep detection.

Keywords: Microsleep; Deep Learning; Computer Vision; Object Detection; YOLOv8; Real Time Detection; Eye Image Detection.

1. PENDAHULUAN

Tingkat kecelakaan lalu lintas di Indonesia terus meningkat setiap tahun [1]. Pada tahun 2023, jumlah kecelakaan lalu lintas tercatat lebih dari 116.000 kasus, meningkat sebesar 6.8% dibandingkan tahun sebelumnya [2]. Berdasarkan data Kepolisian pada 2020, kecelakaan lalu lintas menyebabkan 23.529 kematian, setara dengan tiga jiwa meninggal setiap hari [3]. Menurut Kepolisian RI, Faktor utama terjadinya kecelakaan adalah kelalaian manusia sebagai pengemudi, yaitu sebesar 61%, diikuti faktor prasarana atau lingkungan sebesar 30%, serta kondisi kendaraan sebesar 9% [4]. Hal ini menunjukkan bahwa pengemudi sebagai pengguna utama jalan raya merupakan faktor risiko terbesar dalam kecelakaan lalu lintas [3]. Salah satu bentuk kelalaian yang sering diremehkan adalah mengemudi dalam kondisi lelah dan mengantuk, yang dapat menimbulkan tidur sesaat atau *microsleep* [1], [5]. Ketika mengemudi dalam keadaan mengantuk, tubuh berada pada kondisi tidak fokus sehingga kemampuan mengemudi menurun drastis [3].

Microsleep sendiri dapat terjadi kapan saja, baik ketika tidak melakukan aktivitas maupun saat beraktivitas, dengan durasi sekitar 3–10 detik hingga seseorang kembali tersadar [6]. Kondisi ini sangat berbahaya bagi keselamatan pengemudi maupun pengguna jalan lain [5]. Oleh karena itu, diperlukan inovasi yang dapat memantau kondisi pengemudi untuk mencegah terjadinya *microsleep* saat berkendara. Salah satu inovasi yang relevan adalah pengembangan sistem deteksi dini *microsleep* berbasis citra mata yang dapat bekerja secara *real-time* menggunakan pendekatan *deep learning*.

Deep learning merupakan cabang kecerdasan buatan yang memungkinkan komputer belajar merepresentasikan data serta mengenali berbagai bentuk data, termasuk gambar, video, dan teks [7]. Dalam

penelitian ini, parameter utama yang digunakan untuk mendeteksi *microsleep* adalah kondisi mata pengemudi. Deteksi dilakukan melalui citra mata yang ditangkap kamera, kemudian dianalisis untuk membedakan mata terbuka atau tertutup. Untuk mendukung proses ini dibutuhkan teknologi *computer vision*, yaitu teknologi yang memungkinkan komputer melihat dan mengenali objek di sekitarnya seperti layaknya manusia [8]. *Computer vision* dapat mengekstrak informasi serta memahami data visual secara otomatis, baik berupa gambar maupun video [9]. Salah satu cabang *computer vision* yang paling populer adalah *object detection*. *Object detection* berfungsi mendeteksi keberadaan objek tertentu pada citra digital atau frame video [10]. Cara kerja *object detection* adalah dengan mengenali fitur-fitur objek dari citra yang diinput, lalu dibandingkan dengan fitur citra pada dataset referensi, sehingga dapat ditentukan apakah objek tersebut termasuk objek yang ingin dideteksi atau tidak [10].

Untuk mewujudkan deteksi objek diperlukan algoritma *deep learning* yang tepat. Algoritma yang banyak digunakan dan relevan dalam penelitian ini adalah YOLO (*You Only Look Once*), sebuah metode berbasis *Convolutional Neural Network (CNN)* [11]. YOLO memiliki keunggulan dalam mendeteksi objek hanya dengan satu kali proses, sehingga jauh lebih cepat dibandingkan metode lain [2]. YOLO (*You Only Look Once*) pertama kali diperkenalkan pada tahun 2015 oleh Joseph Redmon [12]. Versi YOLO yang digunakan pada penelitian ini adalah YOLOv8, yang dibuat dan diterbitkan pada bulan Januari 2023 oleh Ultralytics, developer YOLOv5 [13]. YOLO versi 8 memiliki peningkatan kinerja yang berfokus pada efisiensi model, kecepatan, dan akurasi dalam deteksi objek [12].

Dalam membangun model YOLOv8, dibutuhkan dataset yang relevan dengan tujuan sistem. Sumber dataset pada penelitian ini adalah Roboflow Universe. Roboflow dapat membagikan dataset sekaligus memproses dataset tersebut, melakukan *annotate* atau menandai objek yang akan dideteksi menggunakan *bounding box*, selain itu dapat juga melakukan *pre-processing* pada dataset [14].

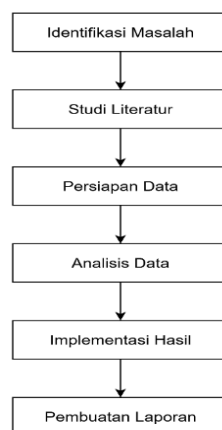
Beberapa penelitian terdahulu menunjukkan relevansi penggunaan YOLOv8 di bidang transportasi dan keselamatan. Ikbaldkk. melakukan penelitian yang menghasilkan inovasi untuk mencegah kecelakaan lalu lintas dengan cara pengenalan rambu menggunakan YOLOv8, sehingga pengemudi tidak lagi bergantung pada kewaspadaan manual dalam membaca rambu jalan [15]. Edmund Ucok Armin dkk. membahas penerapan YOLOv8 untuk mendeteksi kantuk pada pengemudi melalui citra wajah, dengan tingkat keberhasilan tinggi yaitu mAP mencapai 96% [16]. Penelitian-penelitian tersebut menunjukkan bahwa YOLOv8 merupakan algoritma yang tepat untuk deteksi objek melalui kamera.

Namun, hingga saat ini belum ditemukan penelitian yang secara khusus mengembangkan sistem deteksi dini *microsleep* berbasis citra mata sekaligus dilengkapi peringatan *real-time*. Penelitian ini memiliki keterbaruan dan keunggulan dibandingkan studi sebelumnya yang umumnya hanya menilai performa model YOLO v8 dalam deteksi kantuk. Pada penelitian ini, sistem tidak hanya mengklasifikasi kondisi mata, tetapi juga dilengkapi dengan mekanisme peringatan *real-time* berupa audio dan visual menggunakan logika durasi (≥ 3 detik) yang sesuai dengan karakteristik *microsleep*. Hal ini menjadikan penelitian yang tidak hanya mengevaluasi model, namun juga memberikan solusi praktis yang dapat diintegrasikan langsung ke sistem keselamatan pengemudi kendaraan roda empat.

2. METODOLOGI PENELITIAN

2.1 Kerangka Kerja

Kerangka kerja penelitian bertujuan untuk mengatur alur proses selama melakukan penelitian. Kerangka kerja penelitian memperjelas tahap – tahap yang dilakukan untuk mencapai hasil penelitian.



Gambar 2.1. Kerangka Kerja Penelitian

Berdasarkan kerangka kerja di atas, dapat diuraikan setiap tahapannya sebagai berikut :

2.1.1 Identifikasi Masalah

Berdasarkan data Kepolisian pada 2020, kematian yang disebabkan oleh kecelakaan lalu lintas mencapai 23.529 jiwa, setara dengan tiga jiwa meninggal dalam satu hari [3] Menurut Kepolisian RI, faktor utama terjadinya kecelakaan lalu lintas adalah kelalaian manusia sebagai pengendara [4]. Bentuk kelalaian pengemudi dalam berkendara yang sering diremehkan adalah berkendara ketika kelelahan dan mengantuk yang dapat memicu kondisi *Microsleep* [1]. Kondisi *microsleep* bisa membahayakan keselamatan pengemudi dan lingkungan sekitarnya karena penurunan fokus dalam mengemudi [5] *Microsleep* dapat terjadi kapanpun, dalam keadaan tidak melakukan apa-apa atau saat sedang melakukan kegiatan dengan durasi sekitar 3 – 10 detik hingga seseorang kembali tersadar [6]

Oleh karena itu, diperlukan sistem deteksi dini *microsleep* yang dapat mengidentifikasi kondisi mata tertutup secara akurat dan memicu alarm peringatan sebelum kecelakaan terjadi. Sistem ini harus mampu beroperasi dalam berbagai kondisi pencahayaan dan sudut pandang kamera, serta berjalan secara efisien menggunakan perangkat keras sederhana. Penelitian ini bertujuan untuk menjawab permasalahan tersebut dengan melakukan studi literatur untuk menghasilkan sistem deteksi dini *microsleep*.

2.1.2 Studi Literatur

Studi literatur dilakukan sebagai langkah awal untuk memperkuat landasan teoritis serta memahami pendekatan dan teknologi yang relevan dalam mendeteksi *microsleep* secara otomatis. Kajian ini mencakup berbagai sumber ilmiah dan penelitian sebelumnya yang membahas konsep *microsleep*, algoritma deteksi objek, serta penerapan *computer vision* dalam sistem keselamatan berkendara sebagai berikut.

Tabel 2.1. Penelitian Sejenis

Peneliti	Judul	Masalah	Metode	Hasil	Referensi
Edmund Ucok Armin, Anggun Purnama Edra, Fakhri Ikhwanul Alifin, Ikhwanussafa Sadidan, Indri Purwita Sary, Ulinnuha Latifa.	Performa Model YOLO v8 untuk Deteksi Kondisi Mengantuk pada Pengendara Mobil	Kurangnya sistem deteksi kantuk yang akurat tanpa kontak langsung dengan pengemudi.	Algoritma YOLO v8, dataset sebanyak 3708 gambar, pelatihan <i>deep learning</i> , evaluasi mAP dan F1 skor.	Nilai mAP sebesar 96.09%, <i>precision</i> 94.65%, <i>recall</i> , 95.26%, dan F1 skor sebesar 94.96%.	[16]
Novita Ranti Muntari, Indah Chairun Nisa, Ana Sriekaningih, Andri Yogi Adyatma Prasetyo, Muhammad Yusril.	Penerapan YOLO v8 dalam Identifikasi Wajah secara <i>Real-Time</i> Menggunakan CCTV untuk Presensi Siswa	Sistem presensi manual yang tidak efisien dan rawan manipulasi.	Algoritma YOLO v8, dataset wajah sebanyak 120 gambar, roboflow untuk melatih dataset, Python, dan CCTV sebagai media.	Nilai mAP sebesar 88.1%, <i>precision</i> 76.1%, dan <i>recall</i> sebesar 82.8%.	[17]
Muh. Ikbal, R. A. Saputra.	Pengenalan Rambu Lalu Lintas Menggunakan Metode YOLO v8	Sulitnya mengenali rambu lalu lintas secara otomatis dan akurat.	Algoritma YOLO v8, dataset rambu lalu lintas, dan bahasa pemrograman Python.	<i>Precision</i> sebesar 99.3%, <i>recall</i> sebesar 99.9%, dan mAP sebesar 98.4%	[15]

Eben Panja, Hendry, Christine Dewi.	YOLO v8 <i>Analysis for Vehicle Classification Under Various Image Conditions</i>	Keterbatasan untuk melakukan klasifikasi jenis kendaraan di berbagai kondisi cuaca.	YOLO v8 (seri n, s, dan m), augmentasi <i>horizontal flip</i> , dataset dari DAWN.	YOLO v8 seri m unggul dengan mAP sebesar 71.2% (dengan flip), hasil lebih stabil di cuaca ekstrem.	[18]
N. Sujana, M. M. Mutoffar, A. A. Haryanto.	Analisis Kinerja YOLO v8 Optimisasi Roboflow untuk Deteksi Ekspresi Wajah Emosional	Deteksi ekspresi wajah sering tidak akurat karena kualitas data buruk.	Algoritma YOLO v8, Roboflow untuk augmentasi data, pembelajaran <i>machine learning</i> , dataset ekspresi wajah.	YOLO v8 mampu mendeteksi ekspresi dengan hasil optimal setelah dilakukan augmentasi menggunakan Roboflow.	[19]

- Penelitian Edmund Ucok Armin dkk. menggunakan *YOLOv8* untuk deteksi kantuk pada pengemudi dengan mAP 96.09% dan F1 Score 94.96%; fokus pada performa model, berbeda dengan penelitian ini yang menambahkan sistem alarm real-time.
- Penelitian Novita Ranti Muntari dkk. menerapkan *YOLOv8* untuk presensi siswa dengan mAP 88.1%, precision 76.1%, recall 82.8%; meski tujuan berbeda, relevan karena sama-sama mendeteksi wajah secara *real-time*.
- Penelitian Muh. Ikbil dan R.A. Saputra mengenali rambu lalu lintas menggunakan *YOLOv8* dengan hasil akurasi baik; membuktikan efektivitas deteksi visual cepat, sejalan dengan penelitian ini pada deteksi mata.
- Penelitian Eben Panja dkk. menganalisis *YOLOv8* untuk klasifikasi kendaraan, varian terbaik *YOLOv8-m* dengan mAP 71.2%; meski objek berbeda, mendukung performa *YOLOv8* pada kondisi ekstrem.
- Penelitian N. Sujana dkk. menerapkan *YOLOv8* untuk deteksi ekspresi wajah dengan optimasi *Roboflow*; relevan karena sama-sama mendeteksi area wajah, serta menunjukkan pentingnya pengolahan dataset yang baik.

Selanjutnya, studi literatur difokuskan pada teknologi *computer vision* dan algoritma *YOLOv8* (*You Only Look Once* versi 8). Metode deteksi objek menggunakan gambar atau video merupakan penerapan dari *computer vision*, yaitu ilmu komputer yang bekerja dengan cara meniru kemampuan visual manusia [20]. *You Only Look Once* (YOLO) versi 8 dibuat dan diterbitkan pada bulan Januari 2023 oleh Ultralytics, developer *YOLOv5* [13]. YOLO versi 8 memiliki peningkatan kinerja yang berfokus pada efisiensi model, kecepatan, dan akurasi deteksi [12]. Beberapa penelitian terdahulu, seperti oleh Armin dkk. membuktikan bahwa YOLO v8 efektif dalam mendeteksi kondisi mengantuk dengan nilai mAP dan F1-Score tinggi, tetapi pendekatan tersebut masih terbatas pada evaluasi performa model dan belum diterapkan secara langsung dalam sistem peringatan dini *microsleep* [16].

Selain itu, Roboflow sebagai *platform* manajemen dataset dan Visual Studio Code sebagai lingkungan pengembangan turut dikaji dalam literatur untuk mempersiapkan proses *training* dan implementasi. Studi terhadap pustaka OpenCV juga dilakukan untuk memahami integrasi sistem deteksi visual dengan alarm berbasis audio secara *real-time*. Dari keseluruhan studi yang dilakukan, dapat disimpulkan bahwa kombinasi metode *object detection* berbasis *YOLOv8* dengan pendekatan visual terhadap kondisi mata pengemudi memiliki potensi besar untuk dikembangkan sebagai sistem peringatan dini yang praktis dan aplikatif dalam dunia nyata.

2.1.3 Persiapan Data

Dataset yang digunakan adalah SleepyDetect dari Roboflow Universe dengan 1.779 citra beranotasi. Dari 4 kelas awal, hanya digunakan 2 kelas: mata terbuka (*Open-eyes*) dan mata tertutup (*Closed-eyes*). Data dibagi menjadi *train*, *validation*, dan *test* secara *stratified random split*. Untuk meningkatkan variasi, dilakukan *augmentasi* berupa *resize*, *horizontal flip*, serta penyesuaian *brightness* dan *contrast*.

2.1.4 Analisis Data

Model YOLO v8n dilatih menggunakan dataset SleepyDetect pada perangkat CPU dengan bantuan *library* Ultralytics. Evaluasi dilakukan menggunakan metrik *mAP@0.5*, *precision*, *recall*, *F1-score*, serta *loss* (*classification*, *box*, *objectness*) untuk mengukur performa deteksi.

2.1.5 Implementasi Hasil

Model terbaik kemudian diintegrasikan ke sistem deteksi *real-time*. Kamera digunakan untuk menangkap citra mata, lalu sistem memberikan peringatan berupa alarm dan teks jika mata tertutup ≥ 3 detik. Hasil deteksi ditampilkan dengan *bounding box*, label, dan *confidence score*.

2.1.6 Pembuatan Laporan

Tahap akhir adalah penyusunan laporan penelitian secara sistematis, meliputi identifikasi masalah, studi literatur, persiapan data, analisis, implementasi, hingga evaluasi sistem.

2.2 TEKNIK ANALISIS DATA

2.2.1 Analisis Kualitatif

Meliputi pengamatan visual terhadap hasil *bounding box* deteksi mata dari YOLO v8, serta respons sistem alarm terhadap kondisi “mata tertutup” lebih dari 3 detik.

2.2.2 Analisis Kuantitatif

Menggunakan metrik evaluasi performa model untuk menghitung :

a. Presisi

Nilai presisi adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem [16]. Berikut rumus untuk menghitung presisi :

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

b. Recall

Nilai *Recall* adalah persentase sebuah program memprediksi sebuah data ke bukan kelas aktualnya, yang bisa disebut juga dengan sensitifitas [16]. Berikut rumus untuk menghitung *Recall* :

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

c. F1-score

Nilai *F1-Score* merupakan salah satu perhitungan evaluasi yang menunjukkan perbandingan rata-rata recall dan presisi yang dibobotkan [16]. Berikut rumus untuk menghitung *F1-score*:

$$F_1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

d. Mean Average Precicion (mAP)

mAP merupakan presisi rata-rata yang sering digunakan dalam mengukur keakuratan model deteksi [16]. Berikut rumus untuk menghitung mAP :

$$Average\ Precision\ (AP) = \int_0^1 Precision(r)\ dr \quad (4)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (5)$$

3. HASIL DAN PEMBAHASAN

Bab ini membahas hasil dari proses pelatihan dan pengujian model YOLO v8 untuk mendeteksi kondisi mata pengemudi, apakah terbuka atau tertutup, sebagai deteksi awal terhadap gejala *microsleep*. Dataset SleepyDetect digunakan dan telah diproses serta dibagi menjadi tiga bagian, yaitu data latih, validasi, dan uji. Hasil dari pelatihan model ditampilkan dalam bentuk nilai metrik dan juga gambar visualisasi performa model.

3.1 Proses Pembuatan Program

Bagian ini menjelaskan secara rinci bagaimana sistem dibangun, mulai dari tahap awal persiapan data (*preprocessing*), proses pelatihan model, hingga integrasi model ke dalam program deteksi *microsleep* secara *real-time*.

3.1.1 Preprocessing Data

Sebelum model dilatih, data perlu dipersiapkan terlebih dahulu. Proses ini disebut *preprocessing*. Tujuannya adalah untuk memastikan semua gambar memiliki format dan ukuran yang seragam, serta mempermudah model dalam mengenali pola. Langkah-langkah *preprocessing* yang dilakukan adalah:

a. Struktur Folder Dataset

Dataset asli terdiri atas gambar dan label yang disimpan dalam folder `raw/images` dan `raw/labels`. Dataset kemudian dibagi ke dalam tiga folder yaitu `train`, `valid`, dan `test` menggunakan *script* Python. Proses pembagian dilakukan secara acak namun seimbang (*stratified data splitting*), dengan rasio :

- Data latih (*train*) : 1.245 gambar (70%)
- Data validasi (*validation*) : 356 gambar (20%)
- Data uji (*test*) : 178 gambar (10%)

b. Pemilihan dan Pengubahan Label

Dataset awal memiliki empat kelas. Namun, hanya dua kelas yang digunakan dan label dengan kelas lain dihapus dari file `.txt`. Dua kelas tersebut yaitu :

- Kelas 0 → tetap sebagai 0 (Close-eyes)
- Kelas 3 → diubah menjadi 1 (Open-eyes)

c. Augmentasi Data

Untuk memperbanyak data latih dan meningkatkan variasi gambar, dilakukan augmentasi menggunakan *library* `Albumentations`. Proses augmentasi meliputi :

- *Resize* gambar menjadi 320x320 piksel
- *Horizontal flip* (50%)
- Penyesuaian *brightness* dan *contrast* (30%)

Gambar hasil augmentasi disimpan dengan akhiran `_aug.jpg` dan label disalin tanpa perubahan koordinat karena transformasi dilakukan secara aman.

d. Penggabungan Gambar Augmentasi

Setelah semua augmentasi dilakukan, gambar dan label hasil augmentasi dipindahkan kembali ke folder `train/images` dan `train/labels` agar dapat digunakan bersama data asli.

3.1.2 Pelatihan Model YOLOv8n

Pada tahapan augmentasi di data *preprocessing*, total data pelatihan bertambah dua kali lipat. Dari jumlah awal 1.245 gambar asli, dilakukan augmentasi sebanyak 1.245 gambar tambahan. Data hasil augmentasi digabung dengan data asli sehingga total data pelatihan yang digunakan pada model menjadi 2.490 gambar. Dataset dilatih dengan model YOLOv8 versi nano karena ukurannya kecil dan cocok untuk pelatihan di laptop tanpa GPU. Model dilatih dengan menggunakan *library* `Ultralytics` dengan parameter pelatihan sebagai berikut :

```
from ultralytics import YOLO
model = YOLO('yolov8n.pt')

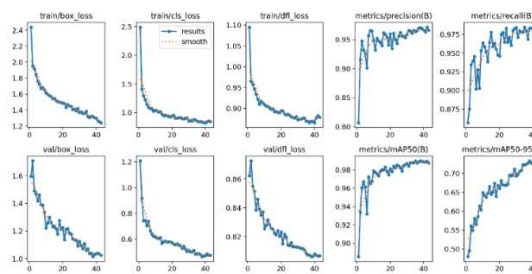
model.train(
    data='D:\PP\Program Microsleep\SleepyDetect.v2i.yolov8-obb\data.yaml',
    epochs=50,
    patience=5,
    imgsz=320,
    batch=4,
    name='yolov8_microsleep_cpu',
    project='runs/train',
    device='cpu'
)
```

Gambar 3.1. Parameter Pelatihan Model

Dengan pengaturan parameter ini, pelatihan model dapat dilakukan secara efisien meskipun menggunakan sumber daya komputasi terbatas, seperti CPU.

3.2 Hasil Pelatihan Data Menggunakan Model YOLOv8n

Grafik pelatihan yang dihasilkan ditunjukkan pada gambar berikut :



Gambar 3.2. Grafik Pelatihan Model

Grafik di atas memperlihatkan penurunan nilai loss dan peningkatan *precision* serta *recall* selama proses pelatihan, yaitu :

- Train/val box_loss, cls_loss, dfl_loss menunjukkan penurunan konsisten pada semua jenis *loss*, menandakan model mempelajari data dengan baik.
- Precision* dan *Recall* (B) meningkat dan stabil di atas 95%, menunjukkan model mampu mendeteksi objek dengan tingkat kesalahan rendah dan sensitivitas tinggi.
- mAP50 dan mAP50-95 (B) mengalami peningkatan tajam di awal pelatihan, lalu stabil, mengindikasikan bahwa model memiliki akurasi tinggi dalam mendeteksi dan mengklasifikasikan objek pada berbagai *threshold* IoU.

3.3 Visualisasi Hasil Evaluasi

Evaluasi model juga dilakukan dengan melihat hasil deteksi secara visual dan grafik performa. Berikut adalah penjelasan dan visualisasi yang dihasilkan dari evaluasi model :

a. Deteksi Gambar Validasi

Model diuji menggunakan gambar validasi. Hasil deteksi ditampilkan dalam bentuk *bounding box* dengan *confidence score*. Gambar berikut adalah contoh hasil deteksi pada *batch* ke-0 data validasi :



Gambar 3.3. Deteksi Validasi Batch ke-0

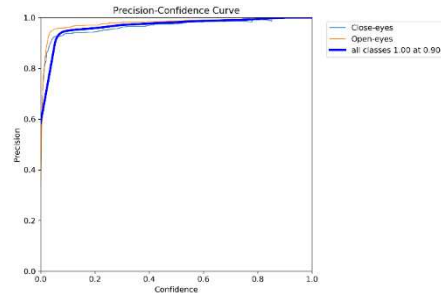
Gambar di atas menunjukkan hasil deteksi objek menggunakan model hasil *training* yang diterapkan pada kondisi mata terbuka atau tertutup, hasilnya sebagai berikut :

- Setiap *frame* menampilkan wajah manusia dengan label hasil deteksi seperti Open-eyes atau Close-eyes, disertai nilai probabilitas *confidence score*.
- Deteksi dilakukan secara *real-time*, terlihat dari beragam pose dan ekspresi dalam kondisi pencahayaan yang berbeda-beda.
- Confidence score* di atas 0.7 menunjukkan tingkat kepercayaan model yang cukup baik terhadap prediksi dengan warna label biru untuk Close-eyes dan hijau muda untuk Open-eyes.

Hasil ini menunjukkan bahwa model hasil training menggunakan YOLO v8n mampu membedakan kondisi mata secara akurat dalam berbagai situasi.

b. **Kurva Precision**

Precision menunjukkan seberapa tepat model dalam mengklasifikasikan kondisi mata. Berikut hasil presisi dari model yang telah dilatih :

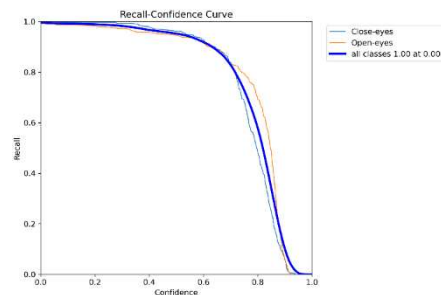


Gambar 3.4. Kurva Precision

Grafik di atas menunjukkan *Precision-Confidence Curve* dari hasil pelatihan model YOLO v8n untuk klasifikasi dua kondisi mata, Close-eyes dan Open-eyes. Model YOLOv8 menunjukkan akurasi tinggi dan stabil dalam mendeteksi kondisi mata terbuka dan tertutup. Dengan *confidence* ≥ 0.9 , model mampu mencapai *precision* sempurna, yang mana sangat penting untuk sistem deteksi *microsleep* agar meminimalkan *false positive*.

c. **Kurva Recall**

Recall menunjukkan seberapa sensitif model terhadap deteksi. Berikut adalah kurva *recall* dari model yang telah dilatih :

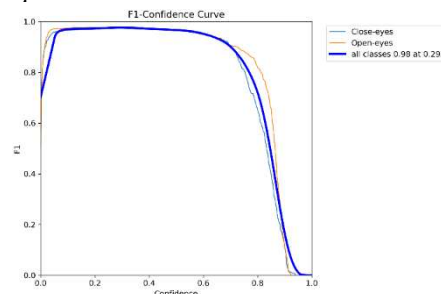


Gambar 3.5. Kurva Recall

Model menunjukkan seiring meningkatnya nilai *threshold*, performa *recall* mengalami penurunan signifikan, yang menandakan peningkatan risiko terjadinya *false negative*. Oleh karena itu, pemilihan *threshold* yang tepat penting untuk menjaga keseimbangan antara sensitivitas dalam kemampuan mendeteksi kondisi *microsleep* dan presisi dalam menghindari alarm palsu.

d. **Kurva F1-Score**

F1-Score adalah gabungan dari *precision* dan *recall*. Berikut adalah F1 score dari model yang telah dilatih :

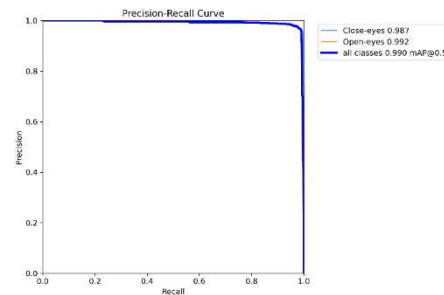


Gambar 3.6. Kurva F1-Score

Model YOLOv8 menunjukkan performa deteksi kondisi mata yang sangat baik dengan F1-score maksimal 0.98, menandakan keseimbangan optimal antara deteksi benar dan minimnya kesalahan. *Threshold confidence* terbaik adalah sekitar 0.29 untuk diterapkan pada sistem deteksi dini *microsleep* agar tetap responsif dan akurat.

e. **Kurva *Precision-Recall* (PR)**

Kurva PR menggambarkan hubungan antara *precision* dan *recall* untuk masing-masing kelas. Berikut adalah *Precision-Recall Curve* dari model yang telah dilatih :

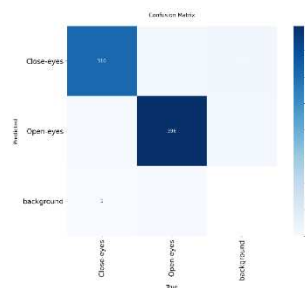


Gambar 3.7. Kurva PR

Precision-Recall Curve ini menunjukkan bahwa model YOLO v8n sangat efektif dan efisien dalam mengklasifikasikan kondisi mata, baik dalam mendeteksi maupun memastikan prediksi benar, dengan kinerja hampir sempurna di kedua metrik.

f. ***Confusion Matrix***

Confusion matrix menunjukkan jumlah prediksi benar dan salah pada masing-masing kelas. Di bawah ini adalah *confusion matrix* dari model yang telah dilatih :



Gambar 3.8. Confusion Matrix

Confusion matrix ini mengonfirmasi bahwa model YOLO v8n bekerja sangat efektif dalam mengenali kondisi mata terbuka dan tertutup serta memisahkannya dari latar belakang dengan kesalahan klasifikasi yang sangat kecil.

3.4 Implementasi Model

Setelah pelatihan, model terbaik (best.pt) digunakan dalam sistem deteksi *real-time*. Berikut adalah komponen utama dalam implementasi model untuk sistem deteksi dini *microsleep* :

a. **Inisialisasi Model**

Berikut adalah inisiasi yang dilakukan untuk membangun sistem :

```
import cv2
import time
import pygame
from ultralytics import YOLO

pygame.mixer.init()
pygame.mixer.music.load("alarm.mpeg")

model = YOLO("D:/PP/Program Microsleep/runs/train/yolov8_microsleep_cpu/weights/best.pt")
DROWSY_THRESHOLD = 3
WAKE_THRESHOLD = 3
cap = cv2.VideoCapture(1)
```

Gambar 3.9. Inisialisasi Model

Kode di atas memuat pemanggilan file model YOLO hasil pelatihan (best.pt) yang digunakan untuk mendeteksi mata terbuka dan tertutup. Selain itu, inisialisasi variabel penting seperti DROWSY_THRESHOLD dan WAKE_THRESHOLD berfungsi untuk menentukan durasi batas waktu (dalam detik) untuk mengaktifkan dan menghentikan alarm *microsleep* secara otomatis.

b. Deteksi *Real-Time*

Setiap *frame* dari kamera kemudian diproses oleh model untuk mendeteksi kondisi mata.

```
results = model.predict(source=frame, conf=0.4, verbose=False)
predictions = results[0].boxes.data.cpu().numpy()

current_time = time.time()
eye_closed_detected = False

for pred in predictions:
    class_id = int(pred[5])
    conf_score = float(pred[4])
    class_name = model.names[class_id]
```

Gambar 3.10. Deteksi Kelas Model

Pada bagian ini, model akan mengidentifikasi objek apakah termasuk kelas “Open-eyes” atau “Close-eyes”.

c. Visualisasi Deteksi pada Layar

Setelah objek berhasil dikenali, sistem menampilkan hasil deteksi ke layar berupa kotak pembatas (*bounding box*), label kelas, dan *confidence score*.

```
cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)

cv2.putText(frame, class_name, (x1, y1 - 15),
            cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)

cv2.putText(frame, f"{conf_score:.2f}", (x1, y1),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
```

Gambar 3.11. Visualisasi Deteksi pada Layar

Kotak pembatas (*bounding box*) dan label kelas membantu pengguna memahami apa yang sedang terdeteksi oleh sistem. Warna hijau menandakan mata terbuka, sedangkan warna merah menunjukkan kondisi mata tertutup. Ini juga menjadi komponen visual yang utama dalam sistem monitoring *microsleep* secara *real-time*.

d. Penghitungan Waktu

Salah satu logika penting dalam sistem ini adalah menghitung durasi ketika mata pengguna terdeteksi tertutup secara terus-menerus.

```
if eye_closed_detected:
    if eye_closed_start_time is None:
        eye_closed_start_time = current_time
    detected_duration = current_time - eye_closed_start_time

    if detected_duration >= DROWSY_THRESHOLD:
        start_alarm()
        show_warning = True
        counting_open_eyes = True

    eye_open_start_time = None
    open_eye_duration = 0.0
```

Gambar 3.12. Penghitungan Waktu

Logika ini bekerja dengan mencatat waktu saat mata tertutup mulai terdeteksi. Jika kondisi ini bertahan selama lebih dari 3 detik (DROWSY_THRESHOLD), maka sistem akan mengaktifkan alarm dan peringatan, serta melakukan perhitungan waktu saat Open-eyes terdeteksi kembali. Dengan pendekatan ini, sistem dapat membedakan antara kedipan mata biasa dan tanda-tanda *microsleep*.

e. Aktivasi dan Penghentian Alarm

Jika sistem mendeteksi mata tertutup melebihi ambang waktu, maka fungsi `start_alarm()` akan dijalankan untuk memutar suara alarm. Sebaliknya, `stop_alarm()` menghentikan suara jika diperlukan.

```
def start_alarm():
    global alarm_playing
    if not alarm_playing:
        pygame.mixer.music.play(-1)
        alarm_playing = True

def stop_alarm():
    global alarm_playing
    if alarm_playing:
        pygame.mixer.music.stop()
        alarm_playing = False
```

Gambar 3.13. Aktivasi dan Penghentian Alarm

Alarm bertujuan untuk memperingatkan pengguna bahwa mereka sedang dalam tertidur sejenak (*microsleep*).

f. Peringatan Visual di Layar

Sebagai pelengkap dari peringatan suara, sistem juga menampilkan pesan peringatan secara visual ketika *microsleep* terdeteksi.

```
if show_warning:
    warning_lines = [
        "ANDA BARU SAJA MENGALAMI MICROSLEEP!",
        "MOHON MENEPI DAN BERTISTIRAHAT DAHULU."
    ]
    for i, line in enumerate(warning_lines):
        y = 30 + i * 30
        cv2.putText(frame, line, (10, y),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0, 255), 2)
```

Gambar 3.14. Peringatan Visual di Layar

Pesan ini membuat sistem lebih komunikatif dan responsif terhadap kondisi pengguna.

g. Logika Penghentian Alarm Otomatis

Alarm yang sudah menyala akibat deteksi *microsleep* akan terus aktif sampai sistem mendeteksi kondisi sebagai berikut :

1. Mata tidak lagi tertutup, dan
2. Mata terbuka selama 3 detik berturut-turut.

```
elif eye_open_detected and counting_open_eyes and alarm_playing:
    if eye_open_start_time is None:
        eye_open_start_time = current_time
        open_eye_duration = current_time - eye_open_start_time

    if open_eye_duration >= WAKE_THRESHOLD:
        stop_alarm()
        show_warning = False
        counting_open_eyes = False
        eye_open_start_time = None
        open_eye_duration = 0.0

    eye_closed_start_time = None
    detected_duration = 0.0
```

Gambar 3.15. Logika Penghentian Alarm Otomatis

Dengan logika ini, sistem memastikan bahwa pengemudi memang sudah terjaga sebelum mematikan alarm dan menghapus pesan peringatan. Selain itu, durasi mata terbuka juga akan ditampilkan saat alarm sedang menyala, sebagai informasi tambahan kepada pengguna

h. Interaksi Pengguna melalui Keyboard

Pengguna dapat mengontrol sistem secara manual melalui *keyboard* untuk menghentikan alarm atau keluar dari program.

```
key = cv2.waitKey(1) & 0xFF
if key == ord('b'):
    break
elif key == ord('s'):
    stop_alarm()
    show_warning = False
    eye_closed_start_time = None
    detected_duration = 0.0

cap.release()
```

Gambar 3.16. Interaksi Pengguna melalui Keyboard

Tombol s digunakan untuk menghentikan alarm, sementara tombol b digunakan untuk keluar dari aplikasi. Fungsi ini memberi pengguna kontrol penuh atas sistem dalam situasi tertentu.

3.5 Hasil Implementasi Model

Setelah implementasi model ke dalam sistem *real-time*, dilakukan pengujian pada beberapa kondisi pencahayaan untuk mengevaluasi kemampuan deteksi dan respons alarm terhadap situasi *microsleep*. Berikut adalah hasil tangkapan layar (*screen capture*) dari deteksi pada sistem yang telah berjalan :

3.5.1 Kondisi Cahaya Redup

Model diuji pada lingkungan dengan pencahayaan yang minim untuk mengevaluasi ketangguhannya dalam mengenali kondisi mata dalam situasi kurang cahaya, seperti dalam ruangan. Hasil deteksi didapat sebagai berikut:



Gambar 3.17. Hasil Deteksi Cahaya Redup

Gambar di atas menampilkan deteksi mata terbuka dalam suasana gelap. Model memberikan label Open-eyes dengan *confidence score* yang tinggi, membuktikan bahwa sistem tetap mampu melakukan identifikasi objek meskipun cahaya sangat redup.

3.5.2 Kondisi Cahaya Natural dari Matahari

Pengujian dilakukan pada siang hari dengan pencahayaan alami dari matahari yang menyinari langsung wajah subjek. Model menunjukkan performa sebagai berikut :



Gambar 3.18. Hasil Deteksi Cahaya Natural B

Gambar di atas menunjukkan mata tertutup namun waktu tutup mata belum melebihi batas 3 detik. Deteksi Close-eyes dilakukan dengan tepat, dan alarm belum aktif, menunjukkan bahwa sistem mengikuti logika waktu dengan benar.



Gambar 3.19. Hasil Deteksi Cahaya Natural Matahari

Pada Gambar di atas, mata tertutup selama lebih dari 3 detik berhasil dikenali oleh sistem. Program memicu alarm dan menampilkan peringatan "ANDA BARU SAJA MENGALAMI MICROSLEEP! MOHON MENEPI DAN BERISTIRAHAT DAHULU" secara otomatis. Hal ini membuktikan bahwa implementasi logika alarm berjalan sesuai harapan dalam skenario nyata.

3.5.3 Kondisi *Back Light* (Cahaya dari Belakang)

Pengujian dilakukan dengan sumber cahaya kuat yang berasal dari belakang subjek (*backlight*), yang biasanya menyulitkan deteksi visual. Hasilnya sebagai berikut :



Gambar 3.20. Hasil Deteksi Kondisi Membelakangi Cahaya A

Pada Gambar di atas mata terbuka berhasil dikenali meskipun wajah subjek berada dalam bayangan karena cahaya dari belakang. Sistem tetap menampilkan label *Open-eyes*, membuktikan bahwa model cukup tangguh terhadap pencahayaan ekstrem.

3.6 Pembahasan

Hasil evaluasi menunjukkan bahwa model YOLO v8n memiliki performa sangat baik dalam mendeteksi kondisi mata pengemudi sebagai indikator dini *microsleep*. Model mencapai *precision* 97.2%, *recall* 96.7%, *F1-score* 0.97, dan *mAP@0.5* sebesar 97.9%. Dari sisi *Computer Vision*, sistem mampu mengenali kondisi mata baik terbuka maupun tertutup meskipun terdapat variasi pencahayaan, sudut pandang, maupun ekspresi wajah. Sementara dari sisi *Deep Learning*, arsitektur CNN pada YOLO v8n terbukti efektif dalam mengekstraksi fitur visual sehingga model dapat melakukan generalisasi dengan baik. Hal ini juga didukung oleh proses augmentasi data yang memperkaya variasi citra. Secara algoritmik, pendekatan *single-shot detection* pada YOLO v8n menjadikan sistem efisien untuk dijalankan secara *real-time* meskipun hanya menggunakan CPU.

Jika dibandingkan dengan penelitian sejenis, hasil penelitian ini lebih baik daripada temuan Edmund Ucok Armin dkk. (mAP 96.09%) dan lebih unggul dibandingkan penelitian Novita Ranti Muntiari dkk. (mAP 88.1%). Hal ini menegaskan bahwa *YOLOv8n* merupakan algoritma yang efektif dalam mendeteksi kondisi mata. Perbedaan utama penelitian ini dengan studi sebelumnya terletak pada implementasi sistem peringatan *real-time* berbasis audio dan visual dengan logika durasi (≥ 3 detik), yang sesuai dengan fenomena *microsleep*. Dengan demikian, penelitian ini tidak hanya menilai kinerja model, tetapi juga menghadirkan solusi praktis yang dapat diintegrasikan langsung ke sistem keselamatan pengemudi.

Pemilihan ambang batas 3 detik dalam sistem ini didasari dengan literatur medis yang menyebutkan bahwa episode *microsleep* umumnya berlangsung 3–10 detik [6]. Dengan demikian, penggunaan batas waktu ini penting untuk membedakan antara kedipan normal (umumnya <1 detik) dan *microsleep* yang berpotensi menyebabkan kecelakaan.

Meskipun hasil yang diperoleh cukup tinggi, terdapat sejumlah keterbatasan. Pertama, akurasi deteksi menurun ketika kepala pengemudi miring sehingga sebagian area mata keluar dari tangkapan kamera. Kedua, penggunaan kacamata dengan lensa gelap atau anti-reflektif dapat mengurangi kualitas citra mata yang terbaca sistem. Ketiga, pengujian pada malam hari dengan pencahayaan rendah menunjukkan penurunan performa meskipun telah dilakukan augmentasi data. Oleh karena itu, sistem masih perlu dikembangkan agar lebih tangguh terhadap berbagai kondisi nyata.

Penelitian lanjutan dapat difokuskan pada implementasi menggunakan perangkat *embedded* seperti Raspberry Pi atau NVIDIA Jetson yang lebih ringan, hemat daya, dan mudah dipasang di kendaraan. Penggunaan kamera infra merah juga berpotensi meningkatkan keandalan deteksi pada kondisi minim cahaya sehingga sistem lebih siap diterapkan dalam situasi berkendara nyata.

4. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem deteksi dini *microsleep* berbasis algoritma YOLO v8n yang ringan, efisien, dan mampu mengenali kondisi mata pengemudi secara *real-time* dengan akurasi tinggi. Hasil evaluasi menunjukkan kinerja sangat baik dengan *mAP@0.5* sebesar 97.9%, *precision* 97.2%, *recall* 96.7%, dan *F1-score* 0.97, membuktikan kemampuan model mendeteksi dengan kesalahan yang sangat rendah pada berbagai kondisi pencahayaan. Sistem dilengkapi alarm otomatis yang aktif saat *microsleep* terdeteksi selama ≥ 3 detik serta fitur kendali manual untuk fleksibilitas pengguna, sehingga mampu memberikan peringatan efektif sekaligus menjawab batasan masalah penelitian terkait pencegahan kecelakaan lalu lintas. Meski demikian, penelitian ini masih memiliki keterbatasan pada kondisi kepala miring, penggunaan kacamata, dan pencahayaan malam. Untuk penelitian selanjutnya, sistem akan diarahkan pada pengembangan berbasis perangkat *embedded* sehingga lebih mudah diterapkan secara langsung di kendaraan.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada Universitas Dinamika Bangsa atas dukungannya selama proses penelitian.

REFERENCES

- [1] R. Maulana, "Deteksi Mata Mengantuk Pada Pengemudi Mobil Menggunakan Metode Viola Jones," *JOINTECS (Journal of Information Technology and Computer Science)*, vol. 4, no. 2, 2019, doi: 10.31328/jo.
- [2] A. I. Pradana, H. Harsanto, and W. Wijianto, "Deteksi Rambu Lalu Lintas Real-Time di Indonesia dengan Penerapan YOLOv11: Solusi Untuk Keamanan Berkendara," *Jurnal Algoritma*, vol. 21, no. 2, pp. 145–155, Nov. 2024, doi: 10.33364/algoritma/v.21-2.2106.
- [3] A. Asvin Mahersatillah Suradi, S. Alam, M. Furqan Rasyid, I. Djafar, U. Dipa Makassar, and J. K. Perintis Kemerdekaan, "Sistem Deteksi Kantuk Pengemudi Mobil Berdasarkan Analisis Rasio Mata Menggunakan Computer Vision," *Jurnal Komputer dan Informatika (JUKI)*, vol. 5, Nov. 2023.
- [4] W. U. Puspoprodo and N. N. Laila, "Studi Pemahaman dan Perilaku Keselamatan Berkendara (Safety Riding) pada Remaja dan Usia Produktif di Pulau Jawa," *Jurnal Ilmiah Kesehatan*, vol. 20, no. 3, pp. 118–126, Dec. 2021, doi: 10.33221/jikes.v20i3.1480.
- [5] G. Suyoso, V. Vestine, B. Prakoso, S. Hartanto, and Rusdiarti, "Edukasi Tentang Microsleep Sebagai Upaya Pengurangan Risiko Kecelakaan Kerja pada TRC BPBD Kabupaten Jember," *National Conference for Community Service (NaCosVi)*, vol. 6, 2024.
- [6] C. A. Saputra, D. Erwanto, P. N. Rahayu, and I. Kadiri, "Deteksi Kantuk Pengendara Roda Empat Menggunakan Haar Cascade Classifier Dan Convolutional Neural Network," *JEECOM*, vol. 3, no. 1, Apr. 2021.
- [7] J. Nurhakiki *et al.*, "Studi Kepustakaan: Pengenalan 4 Algoritma Pada Pembelajaran Deep Learning Beserta Implikasinya," *Jurnal Pendidikan Berkarakter*, vol. 2, no. 1, pp. 270–281, Feb. 2024, doi: 10.51903/pendekar.v2i1.598.
- [8] J. Subur, Suryadhi, M. Taufiqurrohman, and N. R. Al Hafizh, "Pemanfaatan Teknologi Computer Vision untuk Deteksi Ukuran Ikan Bandeng dalam Membantu Proses Sortir Ikan," *CYCLOTRON : Jurnal Teknik Elektro*, vol. 7, Jan. 2024.
- [9] Supiyandi and Rafif Rasendriya, "Penggunaan Visi Komputer untuk Mengidentifikasi Jenis Buah dari Gambar," *Router : Jurnal Teknik Informatika dan Terapan*, vol. 2, no. 4, pp. 94–103, Nov. 2024, doi: 10.62951/router.v2i4.287.
- [10] G. Novandra Rizkatama, A. Nugroho, and S. Alfa Faridh, "Sistem Cerdas Penghitung Jumlah Mobil untuk Mengetahui Ketersediaan Lahan Parkir berbasis Python dan YOLO v4," *Jurnal Edu Komputika*, vol. 8, no. 2, 2021, [Online]. Available: <http://journal.unnes.ac.id/sju/index.php/edukom>
- [11] M. D. R. P. Dio, B. P. Bayu Priyatna, A. L. H. April Lia Hananto, and S. S. H. Shofa Shofiah Hilabi, "Deteksi Objek Kecelakaan Pada Kendaraan Roda Empat Menggunakan Algoritma YOLOv5," *Teknologi: Jurnal Ilmiah Sistem Informasi*, vol. 12, no. 2, pp. 15–26, Dec. 2022, doi: 10.26594/teknologi.v12i2.3260.
- [12] R. Sihombing, W. Harahap, and W. Rahman, "Implementasi Yolo V8 Untuk Mendeteksi Mata Uang Rupiah Emisi Tahun 2022 Ber-Output Audio," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, Aug. 2024.
- [13] J. Jonathan and D. Hermanto, "Penentuan Epochs Hasil Model Terbaik: Studi Kasus Algoritma YOLOv8," *Digital Transformation Technology*, vol. 4, no. 2, pp. 792–798, Sep. 2024, doi: 10.47709/digitech.v4i2.4640.
- [14] N. J. Hayati, D. Singasatia, M. R. Muttaqin, T. Informatika, S. Tinggi, and T. Wastukencana, "Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 Untuk Menghitung Kendaraan," *KOMPUTA : Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 2, 2023, [Online]. Available: <https://universe.roboflow.com/>
- [15] MUH. IKBAL and R. A. Saputra, "Pengenalan Rambu Lalu Lintas Menggunakan Metode YOLOv8," *JIKA (Jurnal Informatika)*, vol. 8, no. 2, p. 204, Apr. 2024, doi: 10.31000/jika.v8i2.10609.
- [16] E. U. Armin, A. Purnama Edra, F. I. Alifin, I. Sadidan, I. P. Sary, and U. Latifa, "Performa Model YOLOv8 untuk Deteksi Kondisi Mengantuk pada pengendara mobil," *BRAHMANA: Jurnal Penerapan Kecerdasan Buatan*, vol. 5, no. 1, pp. 67–76, Dec. 2023.
- [17] N. R. Muntiari, Indah Chairun Nisa, Ana Srikaningih, Andri Yogi Adyatma Prasetyo, and Muhammad Yusril, "Penerapan Algoritma YOLOv8 Dalam Identifikasi Wajah secara Real-Time menggunakan CCTV untuk Presensi Siswa," *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 4, no. 3, pp. 1155–1165, Nov. 2024, doi: 10.51454/decode.v4i3.847.

- [18] E. Panja, H. Hendry, and C. Dewi, "YOLOv8 Analysis for Vehicle Classification Under Various Image Conditions," *Scientific Journal of Informatics*, vol. 11, no. 1, pp. 127–138, Feb. 2024, doi: 10.15294/sji.v11i1.49038.
- [19] N. Sujana, M. Malik Mutoffar, and A. Azzam Haryanto, "Analisis Kinerja YOLOv8 Optimalisasi Roboflow Untuk Deteksi Ekspresi Wajah Emosional Dengan Machine Learning," *NARATIF : Jurnal Ilmiah Nasional Riset Aplikasi dan Teknik Informatika*, vol. 06, Dec. 2024.
- [20] T. Dompeipen and S. Sompie, "Penerapan Computer Vision untuk Pendeteksian dan Penghitung Jumlah Manusia," *Jurnal Teknik Informatika*, vol. 15 no. 4, 2021.