



SISTEM PERINGATAN DINI KANTUK PENGEMUDI MENGGUNAKAN MODEL YOLOV11N BERBASIS CITRA WAJAH

Dedi Triyanto¹⁾, Deny Kurniawan²⁾, Mochamad Wahyudi³⁾, Lise Pujiastuti⁴⁾, Sumanto⁵⁾, Adi Supriyatna⁶⁾

^{1,2}Sistem Informasi, Universitas Bina Sarana Informatika

^{3,5}Informatika, Universitas Bina Sarana Informatika

⁴Sistem Informasi, STMIK Antar Bangsa

⁶Sistem Informasi Akuntansi, Universitas Bina Sarana Informatika

^{1,2,3,4,6}Jl. Kramat Raya No.98, RT.2/RW.9, Kwitang, Kec. Senen, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta 10450

⁴Kawasan Bisnis CBD Ciledug, Jl. HOS Cokroaminoto No.29-35 Blok A5, RT.001/RW.001, Karang Tengah, Kec. Karang Tengah, Kota Tangerang, Banten 15157

Email: ¹dedi.triyanto@bsi.ac.id, ²deny.kurniawan@bsi.ac.id, ³wahyudi@bsi.ac.id, ⁴lise.pujiastuti@gmail.com, ⁵sumanto@bsi.ac.id, ⁶adi.asp@bsi.ac.id

Abstract

Drowsy driving is one of the leading causes of road accidents and presents a serious threat to traffic safety. Studies indicate that 20–30% of traffic accidents are caused by driver fatigue, highlighting the urgent need for an early warning system capable of detecting drowsiness accurately and in real time. This study aims to develop a driver drowsiness detection model using computer vision techniques based on the YOLOv11n algorithm, a lightweight and fast variant of the YOLO family. The model was trained using a facial image dataset processed and augmented via the Roboflow platform, designed to recognize visual indicators of drowsiness. Evaluation results show that the model achieves excellent performance, with a mAP50 score of 0.9710 and a mAP50-95 score of 0.6796. Additionally, the model attained a precision of 0.9382 and a recall of 0.9280, demonstrating high detection capability with minimal false positives. These results confirm that YOLOv11n can be effectively implemented in real-time early warning systems, even on resource-constrained embedded devices. This research successfully addresses the challenges of efficiency and detection accuracy, contributing to the development of AI-based driver safety systems. Future work may focus on integrating this approach with multimodal inputs, such as EOG signals or head movement data, to further enhance detection reliability in real-world driving environments.

Keyword: *Driver Drowsiness, YOLOv11n, Real-Time Detection, Computer Vision, Deep Learning*

Abstrak

Kecelakaan lalu lintas akibat kantuk saat mengemudi merupakan salah satu penyebab utama kematian di jalan raya dan menjadi isu keselamatan yang krusial. Studi menunjukkan bahwa 20–30% kecelakaan disebabkan oleh pengemudi yang mengantuk, sehingga diperlukan sistem peringatan dini yang mampu mendeteksi kondisi ini secara akurat dan real-time. Penelitian ini bertujuan untuk mengembangkan model deteksi kantuk berbasis visi komputer menggunakan algoritma YOLOv11n, yang dikenal sebagai varian ringan dan cepat dari keluarga YOLO. Model dilatih menggunakan dataset citra wajah yang telah diproses dan diaugmentasi melalui platform Roboflow, dengan tujuan untuk mendeteksi tanda-tanda kantuk secara visual. Hasil evaluasi model menunjukkan performa yang sangat baik, dengan nilai mAP50 sebesar 0,9710 dan mAP50-95 sebesar 0,6796. Selain itu, precision mencapai 0,9382 dan recall sebesar 0,9280, yang mengindikasikan kemampuan deteksi yang tinggi serta tingkat kesalahan yang rendah. Temuan ini membuktikan bahwa YOLOv11n dapat diimplementasikan secara efektif dalam sistem peringatan dini untuk meningkatkan keselamatan pengemudi, bahkan pada perangkat dengan sumber daya terbatas. Penelitian ini tidak hanya menjawab tantangan efisiensi dan akurasi deteksi kantuk, tetapi juga memberikan kontribusi nyata bagi pengembangan sistem keselamatan kendaraan berbasis kecerdasan buatan. Ke depan, pengembangan sistem deteksi multimodal yang menggabungkan citra wajah dengan data fisiologis seperti EOG dan detak kepala disarankan untuk meningkatkan keandalan sistem dalam kondisi nyata.

Kata Kunci: *Kantuk Pengemudi, YOLOv11n, Deteksi Real-Time, Visi Komputer, Deep Learning*



1. PENDAHULUAN

Keselamatan berkendara menjadi isu krusial mengingat tingginya angka kecelakaan lalu lintas yang sering disebabkan oleh kantuk saat mengemudi [1]. Studi menunjukkan bahwa 20–30% kecelakaan disebabkan oleh pengemudi yang mengantuk, yang secara signifikan menurunkan kemampuan reaksi, fokus, dan pengambilan keputusan di jalan. Di Indonesia, 79% pengemudi pernah mengemudi dalam keadaan mengantuk, dan 32% hampir terlibat kecelakaan fatal akibatnya. Kantuk pada pengemudi merupakan salah satu penyebab utama kecelakaan di jalan raya, yang menimbulkan risiko serius bagi keselamatan pengemudi maupun publik. Kondisi kelelahan dapat mengganggu konsentrasi, memperlambat waktu reaksi, serta menurunkan kemampuan pengambilan keputusan, sehingga berpotensi menyebabkan kecelakaan yang berujung pada cedera serius bahkan kematian [2].

Mengantuk saat mengemudi dapat disebabkan oleh beberapa faktor utama yang secara signifikan meningkatkan risiko kecelakaan, seperti kurang tidur [1], [3], kelelahan akibat perjalanan jauh tanpa istirahat cukup [4], serta kondisi jalan yang monoton dan minim rangsangan visual [5]. Ketiganya menyebabkan penurunan fokus dan kewaspadaan pengemudi. Oleh karena itu, isu ini menjadi sangat krusial untuk ditangani, khususnya dalam sektor-sektor seperti angkutan truk, transportasi umum, dan perjalanan jarak jauh maupun berkendara secara pribadi [2]. Kondisi ini menunjukkan pentingnya pengembangan sistem peringatan dini deteksi kantuk untuk mencegah risiko kecelakaan fatal.

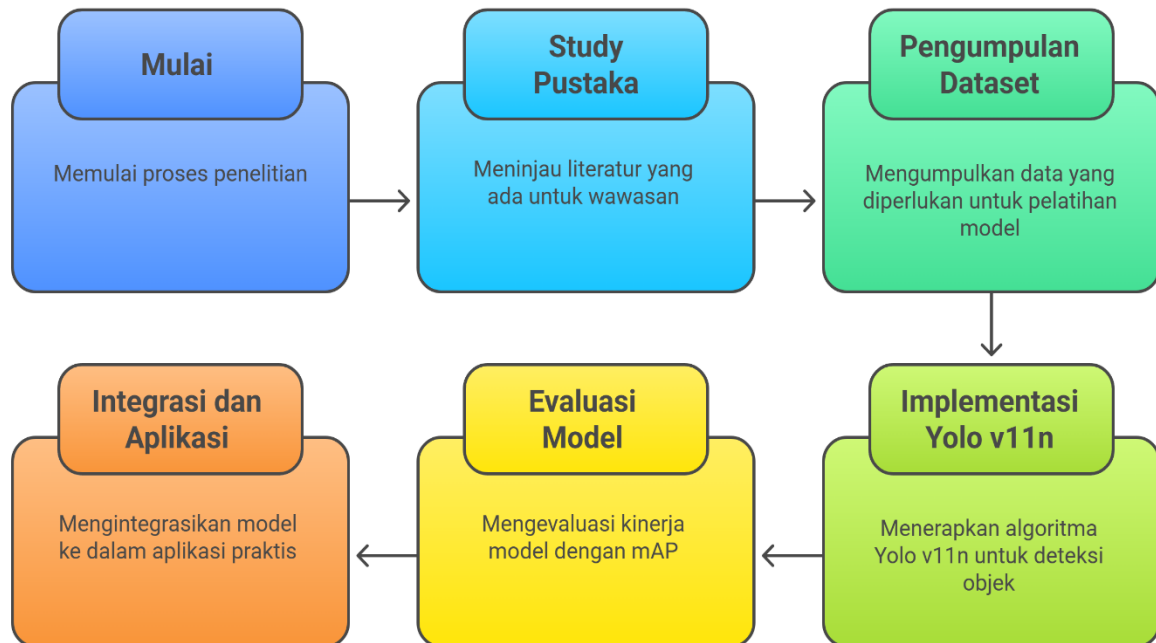
Pendeteksian kantuk secara dini berperan penting dalam memberikan peringatan secara cepat, sehingga pengemudi memiliki kesempatan untuk melakukan tindakan preventif, seperti berhenti sejenak dan beristirahat [6]. Meskipun demikian, sebagian besar teknologi deteksi kantuk yang tersedia saat ini masih menghadapi tantangan dalam hal keakuratan dan kecepatan respons sistem. Salah satu metode yang banyak digunakan dalam bidang pengolahan citra dan dikenal unggul dalam deteksi objek secara real-time adalah You Only Look Once (YOLO) [1], [3], [6].

Meskipun berbagai penelitian sebelumnya telah menerapkan metode deteksi kantuk menggunakan pendekatan berbasis visi komputer, sebagian besar masih mengandalkan kombinasi ekstraksi fitur wajah dengan algoritma klasifikasi seperti Support Vector Machine (SVM) [5], [7]. Beberapa studi menunjukkan bahwa akurasi deteksi menggunakan kombinasi citra dan SVM hanya mencapai kisaran 90% [5], namun menghadapi tantangan dalam hal generalisasi terhadap kondisi pencahayaan yang bervariasi, sudut pandang kamera, dan keterbatasan pemrosesan waktu nyata. Selain itu, penggunaan model YOLO generasi sebelumnya masih belum sepenuhnya optimal dalam mendeteksi ekspresi wajah halus yang menjadi indikator kantuk, terutama ketika digunakan dalam perangkat dengan sumber daya terbatas [8], [9], [10], [11], [12]. Oleh karena itu, dalam penelitian ini diusulkan penerapan YOLOv11n, yaitu varian YOLO yang lebih ringan dan cepat, dengan harapan mampu meningkatkan efisiensi dan akurasi deteksi kantuk secara real-time tanpa mengorbankan performa pada perangkat low-resource [13], [14], [15] seperti sistem tertanam di kendaraan.

Tujuan dari penelitian ini adalah untuk mengembangkan dan menguji model deteksi kantuk pengemudi berbasis YOLOv11n yang mampu bekerja secara real-time dengan akurasi tinggi dan efisiensi komputasi yang lebih baik dibandingkan pendekatan sebelumnya. Melalui penelitian ini, diharapkan dapat tercipta sistem peringatan dini yang lebih responsif dan adaptif terhadap kondisi lapangan, sehingga berkontribusi dalam peningkatan keselamatan berkendara dan pencegahan kecelakaan akibat kantuk.

2. METODE PENELITIAN

Tahapan penelitian yang dirancang secara sistematis untuk mengembangkan sistem peringatan dini kantuk pada pengemudi berbasis YOLOv11n. Penelitian ini diawali dengan identifikasi masalah dan perencanaan, kemudian dilanjutkan dengan studi pustaka untuk memperkuat landasan teori. Tahap berikutnya adalah pengumpulan dataset berupa citra wajah pengemudi dalam berbagai kondisi. Data tersebut digunakan dalam implementasi YOLOv11n untuk mendeteksi tanda-tanda kantuk secara real-time. Setelah model dibangun, dilakukan evaluasi menggunakan metrik seperti mAP untuk mengukur akurasi dan performa. Tahap akhir adalah integrasi model ke dalam sistem aplikasi nyata yang dapat digunakan langsung dalam konteks berkendara. Seluruh tahapan ini digambarkan secara runtut pada Gambar 1.



Gambar 1. Tahapan Penelitian Sistem Peringatan Dini Kantuk Pengemudi dengan YOLOv11n

2.1 Study Pustaka

Setelah proses awal dirumuskan, langkah berikutnya adalah melakukan studi pustaka guna meninjau dan mengkaji berbagai literatur, jurnal, serta penelitian terdahulu yang relevan. Tujuan dari tahap ini adalah untuk memperkuat dasar teori serta memperoleh pemahaman mendalam mengenai metode deteksi kantuk, khususnya penggunaan model deep learning dan arsitektur YOLO dalam visi komputer. Serta untuk mendapatkan gap dari penelitian sebelumnya.

2.2 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini berasal dari platform Roboflow, yang menyediakan data siap pakai untuk pelatihan model pembelajaran mesin, khususnya dalam mendeteksi kantuk pada pengemudi berdasarkan fitur wajah dan perilaku visual. Dataset ini terdiri dari 3.258 gambar untuk pelatihan (83%), 431 gambar untuk validasi (11%), dan 228 gambar untuk pengujian (6%). Seluruh data telah melalui tahap pra-pemrosesan oleh Roboflow, termasuk penyesuaian orientasi dan perubahan ukuran menjadi 640x640 piksel. Selain itu, augmentasi dilakukan dengan menambahkan dua variasi baru pada setiap gambar melalui rotasi ringan sebesar -3° hingga $+3^\circ$, guna meningkatkan keragaman data dan memperkuat kemampuan generalisasi model.

2.3 Implementasi YOLOv11n

Pada tahap ini, algoritma YOLOv11n diterapkan untuk membangun model deteksi kantuk. YOLOv11n dipilih karena merupakan versi ringan dan cepat dari arsitektur YOLO, sehingga cocok untuk digunakan dalam sistem real-time yang dijalankan di perangkat dengan sumber daya terbatas. Proses implementasi mencakup pelabelan data, konfigurasi model, dan pelatihan menggunakan dataset yang telah dikumpulkan. Tools yang digunakan pada tahap ini adalah Roboflow dan Google collabs.

2.4 Evaluasi Model

Setelah model dilatih, langkah selanjutnya adalah mengevaluasi kinerja deteksi menggunakan metrik evaluasi seperti mean Average Precision (mAP), presisi, dan recall. Evaluasi ini bertujuan untuk mengetahui sejauh mana akurasi dan efektivitas model dalam mengenali tanda-tanda kantuk pada pengemudi. Hasil evaluasi juga digunakan sebagai dasar untuk menentukan apakah model sudah siap diterapkan ke tahap selanjutnya atau perlu dilakukan perbaikan.



2.5 Integrasi dan Aplikasi

Tahap terakhir adalah mengintegrasikan model deteksi kantuk ke dalam sistem aplikasi praktis yang dapat digunakan dalam kendaraan. Hasil aplikasi ini adalah berupa aplikasi yang sudah dilengkapi dengan deteksi dini berupa alarm. Model yang sudah dievaluasi dan dinyatakan layak akan diterapkan dalam bentuk sistem peringatan dini yang mampu memberikan notifikasi secara real-time ketika mendeteksi gejala kantuk pada pengemudi. Tahap ini juga mencakup pengujian akhir terhadap kestabilan dan kecepatan sistem di lingkungan nyata.

3. HASIL DAN PEMBAHASAN

3.1 Proses Pelatihan Dataset

1. Instalasi Pustaka Yang Diperlukan

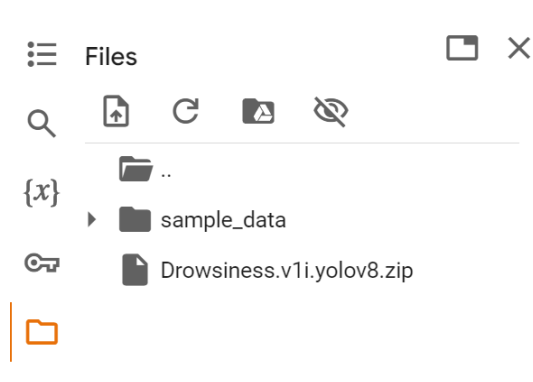
Langkah pertama dalam proses training adalah memastikan semua pustaka yang dibutuhkan telah terinstal pada lingkungan kerja di Google Colab. Dua pustaka utama yang digunakan dalam penelitian ini adalah Ultralytics dan PyTorch, yang mendukung implementasi algoritma YOLOv11n. Instalasi dilakukan dengan menjalankan perintah seperti gambar dibawah ini:

```
[ ] !pip install ultralytics
!pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
```

Gambar 2. Instalasi *library* ultralytics

2. Persiapan Data

Pada tahap awal proses training, data yang akan digunakan harus dipersiapkan dengan baik. Dataset diunduh dari halaman Roboflow, setelah berhasil diunduh, file dataset dalam format ZIP diunggah ke platform Google Colab.

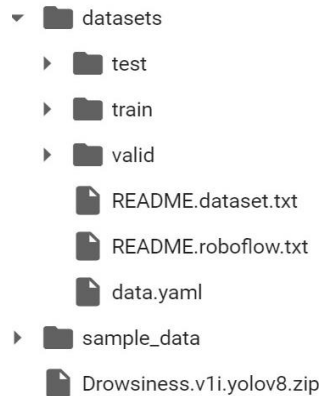


Gambar 3. Dataset telah diupload ke platform Google Colab

Setelah selesai diunduh, file ZIP dataset di-unzip menggunakan perintah Python untuk mengekstrak isinya. Proses ini dilakukan agar file dan struktur direktori dataset dapat diakses oleh algoritma YOLOv11n selama proses training.



```
[ ] !unzip Drowsiness.v1i.yolov8.zip -d datasets
```



Gambar 4. Perintah unzip file Drowsiness.v1i.yolov8.zip kedalam folder "datasets"

3.2. Pengolahan Model Yolov11n

1. Konfigurasi Versi YOLO

Langkah konfigurasi versi YOLO dilakukan untuk mempersiapkan model. YOLOv11n, yang merupakan varian ringan dari algoritma YOLO yang dipilih untuk digunakan dalam proses pelatihan dan deteksi objek.

```
[ ] from ultralytics import YOLO
     model = YOLO('yolo11n.pt') # Gunakan konfigurasi model YOLOv11 nano
```

Gambar 5. Kode konfigurasi model yolo

2. Proses Training

Proses pelatihan melibatkan beberapa parameter utama yang disesuaikan untuk mendapatkan hasil optimal. Kode berikut digunakan untuk menjalankan proses pelatihan model

```
if __name__ == '__main__':
    # Training model
    results = model.train(
        data=f'./datasets/data.yaml', # Path ke file data.yaml
        epochs=50, # Jumlah epoch
        imgsz=640, # Ukuran gambar
        batch=16, # Ukuran batch
        name='drowsiness-detection', # Nama folder penyimpanan hasil
        workers=8 # Jumlah worker untuk dataloader
    )

    # Print notifikasi bahwa training sudah selesai
    print("Training selesai. Hasil tersimpan di:", results)
```

Gambar 6. Kode training pada Google Colab

Beberapa parameter penting yang digunakan dalam proses pelatihan model antara lain: data, yang merujuk pada



jalur file konfigurasi data.yaml berisi informasi lokasi data dan kelas; epochs, yaitu jumlah iterasi penuh terhadap seluruh dataset untuk mendalami pola data; imgsz, ukuran input gambar yang digunakan (misalnya 640x640 piksel); batch, yakni jumlah gambar yang diproses sekaligus dalam satu iterasi—pada penelitian ini digunakan batch size 16 karena seimbang antara kecepatan pelatihan dan efisiensi memori; name, sebagai penamaan folder output hasil pelatihan; serta workers, yaitu jumlah thread paralel untuk mempercepat proses pemuatan data. Kombinasi parameter ini dirancang untuk mengoptimalkan proses pelatihan YOLOv11n secara stabil dan efisien. Setelah konfigurasi sudah selesai akan tampil output dari hasil pelatihan seperti pada gambar 7.

```

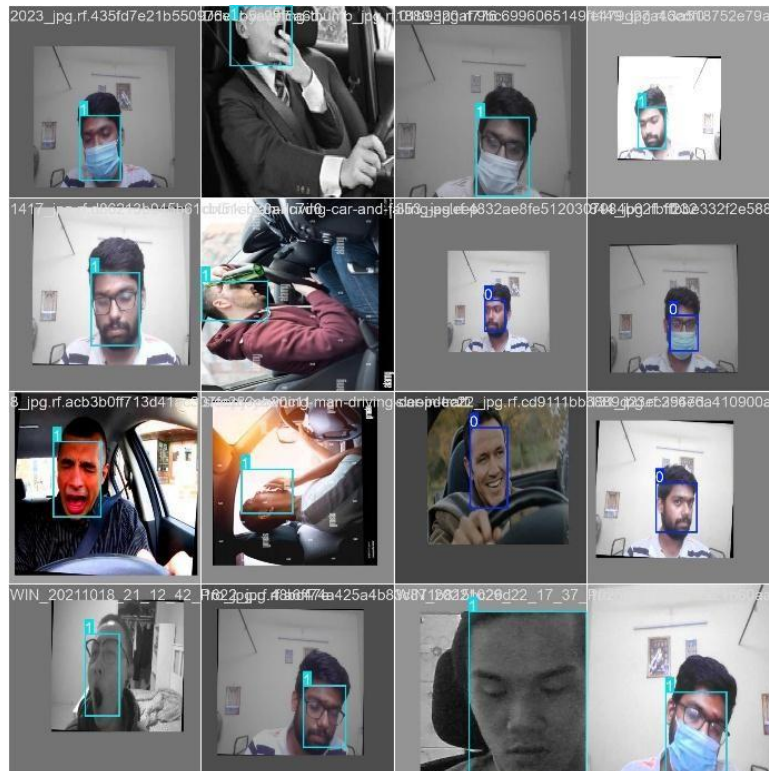
50 Epoch      GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
50/50         2.39G    0.8338   0.3695   1.085     10         640: 100% ██████████ 204/204 [01:0
      Class  Images  Instances  Box(P)   R      mAP50  mAP50-95): 100% ██████████ 14/

50 epochs completed in 1.031 hours.
Optimizer stripped from runs/detect/drowsiness-detection/weights/last.pt, 5.5MB
Optimizer stripped from runs/detect/drowsiness-detection/weights/best.pt, 5.5MB

Validating runs/detect/drowsiness-detection/weights/best.pt...
Ultralytics 8.3.59 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLO11n summary (fused): 238 layers, 2,582,542 parameters, 0 gradients, 6.3 GFLOPs
      Class  Images  Instances  Box(P)   R      mAP50  mAP50-95): 100% ██████████ 14/
      all         431      439     0.932   0.934   0.974   0.702
      Focused    213      217     0.925   0.905   0.968   0.722
      Sleepy     221      222     0.939   0.964   0.979   0.682

Speed: 0.2ms preprocess, 2.4ms inference, 0.0ms loss, 2.1ms postprocess per image
Results saved to runs/detect/drowsiness-detection
Training selesai. Hasil tersimpan di: ultralytics.utils.metrics.DetMetrics object with attributes:
    
```

Gambar 7. Proses pelatihan model dan output lokasi hasil pelatihan





Gambar 8. Hasil Pelatihan pada *batch* ke 8161

```

model = YOLO(model='best.pt')
results = model.val() # Menjalankan evaluasi pada data validation/testing
metrics_dict = results.results_dict # Mengambil dictionary hasil evaluasi

# Menampilkan metrik evaluasi
print("Metrik Evaluasi:")
print(f"mAP50: {metrics_dict['metrics/mAP50(B)']:.4f}") # Mean Average Precision @ IoU
print(f"mAP50-95: {metrics_dict['metrics/mAP50-95(B)']:.4f}") # Mean Average Precision @ IoU
print(f"Precision: {metrics_dict['metrics/precision(B)']:.4f}")
print(f"Recall: {metrics_dict['metrics/recall(B)']:.4f}")

✓ 45.7s

Ultralytics 8.3.56 Python-3.12.8 torch-2.5.1+cu124 CUDA:0 (NVIDIA GeForce RTX 4060 Laptop GPU, 8
YOLOv8m summary (fused): 218 layers, 25,840,918 parameters, 0 gradients, 78.7 GFLOPs
val: Scanning H:\proyek\deteksi-obyek-pengemudi-ngantuk\datasets\valid\labels.cache ... 431 images
WARNING Box and segment counts should be equal, but got len(segments) = 102, len(boxes) = 439. T

Class      Images  Instances  Box(P   R   mAP50  mAP50-95): 100%
  all         431         439    0.938  0.928  0.971   0.68
  Focused    213         217    0.958  0.943  0.971   0.715
  Sleepy     221         222    0.918  0.913  0.971   0.645

Speed: 0.3ms preprocess, 16.5ms inference, 0.0ms loss, 0.6ms postprocess per image
Results saved to runs\detect\val3
Metrik Evaluasi:
mAP50: 0.9710
mAP50-95: 0.6796
Precision: 0.9382
Recall: 0.9280
    
```

Gambar 9. Hasil evaluasi model

Hasil evaluasi model YOLOv11n terhadap dataset pengujian menunjukkan performa yang sangat baik. Model mencapai nilai mAP50 sebesar 0,9710, menandakan tingkat akurasi tinggi dalam mendeteksi objek pada ambang IoU 50%. Sementara itu, nilai mAP50-95 sebesar 0,6796 mencerminkan kemampuan model dalam menghadapi berbagai tingkat ketelitian deteksi, termasuk skenario kompleks. Selain itu, model mencatat precision sebesar 0,9382, yang berarti mayoritas prediksi positif merupakan deteksi yang benar, serta recall sebesar 0,9280, yang menunjukkan bahwa sebagian besar objek yang ada dalam data berhasil dikenali. Secara keseluruhan, hasil ini menegaskan bahwa model memiliki tingkat akurasi, presisi, dan sensitivitas yang tinggi terhadap deteksi kantuk berbasis citra, keterangan lengkap terdapat pada Tabel 1.

Tabel 1. Hasil evaluasi Model

| Metrik | mAP | Analisis dan Interpretasi |
|------------------|------|--|
| mAP50 | 0.97 | Menunjukkan kemampuan model mendeteksi objek dengan sangat akurat saat $IoU \geq 50\%$. Nilai tinggi ini menunjukkan model telah belajar dengan baik dari data latih. |
| mAP50-95 | 0.67 | Meskipun lebih rendah dari mAP50, nilai ini tetap tinggi untuk kondisi pengujian kompleks. Menunjukkan performa realistis dalam berbagai tingkat kecocokan bounding box. |
| Precision | 0.93 | Menandakan tingkat kesalahan false positive sangat rendah. Hal ini penting untuk menghindari alarm palsu dalam aplikasi pendeteksi kantuk pengemudi. |



| | | |
|------------------------|------|--|
| Recall | 0.92 | Menunjukkan bahwa sebagian besar objek yang seharusnya terdeteksi berhasil dikenali oleh model. Penting untuk memastikan bahwa deteksi tidak terlewat (false negative). |
| Kesimpulan Umum | | Kombinasi precision dan recall yang tinggi mengindikasikan performa model yang sangat baik. Nilai mAP50 yang mendekati sempurna mendukung penggunaan dalam sistem deteksi real-time. |

Hasil evaluasi menunjukkan bahwa model memiliki performa sangat baik dengan mAP50 mencapai 0,97 dan precision serta recall yang tinggi (0,93 dan 0,92), menandakan deteksi akurat dengan minim false positive dan false negative. Nilai mAP50-95 sebesar 0,67 mengindikasikan kemampuan yang tetap andal pada berbagai tingkat kecocokan bounding box, sehingga model layak digunakan untuk aplikasi deteksi real-time secara efektif.

4. KESIMPULAN

Penelitian ini dilakukan untuk menjawab permasalahan serius terkait kecelakaan lalu lintas yang disebabkan oleh pengemudi yang mengantuk, dengan merancang dan mengembangkan sistem deteksi kantuk berbasis visi komputer secara real-time. Hasil dari penelitian menunjukkan bahwa model YOLOv11n yang dikembangkan mampu memberikan performa deteksi yang sangat baik pada dataset pengujian. Berdasarkan evaluasi, model menghasilkan nilai mAP50 sebesar 0,9710 dan mAP50-95 sebesar 0,6796. Nilai tersebut menunjukkan bahwa model memiliki kemampuan tinggi dalam mendeteksi objek wajah yang mengindikasikan kondisi kantuk, baik pada tingkat kecocokan minimum hingga kompleks. Selain itu, nilai precision sebesar 0,9382 menunjukkan tingkat kesalahan prediksi positif yang sangat rendah, dan recall sebesar 0,9280 menunjukkan kemampuan model dalam mengenali sebagian besar kondisi kantuk yang ada dalam data uji. Hal ini membuktikan bahwa model tidak hanya efektif tetapi juga andal dalam situasi nyata.

Dengan demikian, seluruh permasalahan utama dalam penelitian ini telah terjawab, yakni bagaimana mengembangkan sistem pendeteksi kantuk pengemudi yang akurat, cepat, dan ringan untuk mendukung implementasi real-time. Sistem ini juga memberikan kontribusi terhadap peningkatan keselamatan berkendara, dengan memberikan peringatan dini terhadap kondisi mengantuk sebelum terjadi risiko yang lebih besar. Penelitian ini memberikan dasar kuat untuk pengembangan lebih lanjut dalam sistem keselamatan kendaraan berbasis AI, khususnya dalam pengawasan kondisi pengemudi. Untuk pengembangan ke depan, disarankan untuk melakukan uji coba langsung dalam kendaraan dan mengombinasikan input dari citra wajah dengan sensor lain seperti EOG atau detak kepala guna meningkatkan akurasi deteksi secara multimodal.

DAFTAR PUSTAKA

- [1] F. C. Yulianto, W. A. Triyanto, and S. Muzid, "Pengembangan Sistem Deteksi Kantuk Menggunakan YOLOv9 untuk Keselamatan dalam Berkendara," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 14, pp. 154–160, 2025.
- [2] T. Malpekar and S. Harne, "DrowsiScan: Early Detection of Driver Drowsiness using Deep Learning," *Int. J. Multidiscip. Res.*, vol. 6, no. 6, pp. 1–7, 2024, doi: 10.36948/ijfmr.2024.v06i06.31576.
- [3] M. Sutrisno, "Deteksi Kantuk Berbasis Jetson Nano Bagi Pengemudi Roda Empat Menggunakan Yolov5," 2022.
- [4] I. Komang Yoga Tri Pranata, Cipta Ramadhani, and G. Wahyu Wiriasto, "Sistem Peringatan Dini Kantuk Pada Pengemudi Malam Hari Menggunakan Metode Facial Landmark Detection Berbasis Raspberry Pi 3 Modul B," *Dielektrika*, vol. 10, no. 2, pp. 100–111, 2023, doi: 10.29303/dielektrika.v10i2.352.
- [5] 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," *JUKI J. Komput. dan Inform.*, vol. 2, pp. 222–230, 2023.
- [6] R. Rahmania, F. Corputty, S. A. Wibowo, D. E. Saputra, and A. Istiqomah, "Exploration of The Impact of Kernel Size for YOLOv5-based Object Detection on Quadcopter," *Int. J. Informatics Vis.*, vol. 6, no. 3, pp. 726–735, 2022, doi: 10.30630/joiv.6.3.898.
- [7] U. D. Maharani, A. S. Handayani, and L. Lindawati, "Analisis Deteksi Mata Kantuk Di Wajah Pengemudi Menggunakan Support Vector Machine (SVM) Berbasis Citra Real-Time," *Build. Informatics, Technol. Sci.*, vol. 6, no. 2, pp. 940–949, 2024, doi: 10.47065/bits.v6i2.5701.
- [8] J. Da Wu and C. H. Chang, "Driver Drowsiness Detection and Alert System Development Using Object Detection," *Trait. du Signal*, vol. 39, no. 2, pp. 493–499, 2022, doi: 10.18280/ts.390211.
- [9] N. I. Vargas-Cuentas and A. Roman-Gonzalez, "Facial image processing for sleepiness estimation," *BioSMART*



- 2017 - *Proc. 2nd Int. Conf. Bio-Engineering Smart Technol.*, 2017, doi: 10.1109/BIOSMART.2017.8095346.
- [10] S. S. Sengar, A. Kumar, and O. Singh, "VigilEye -- Artificial Intelligence-based Real-time Driver Drowsiness Detection," vol. 9332, pp. 0–3, 2024, [Online]. Available: <http://arxiv.org/abs/2406.15646>
- [11] G. Vijendar Reddy, P. Gopala Krishna, L. Sukanya, K. Sri Harsha, N. Harsha Vardhan, and V. Arun, "Image-based Driver Alert System for Prevention of Fatigue-related Accidents," *E3S Web Conf.*, vol. 391, pp. 1–9, 2023, doi: 10.1051/e3sconf/202339101053.
- [12] J. Jose, J. Andrew, K. Raimond, S. Vincent, and R. Jennifer Eunice, "SleepyWheels: An ensemble model for drowsiness detection leading to accident prevention," *J. Auton. Intell.*, vol. 7, no. 3, 2024, doi: 10.32629/jai.v7i3.1117.
- [13] F. Jeribi, A. Siddiqa, H. Kibriya, A. Tahir, and N. Rana, "Efficient Wound Classification Using YOLO n: A Lightweight Deep Learning Approach," *Comput. Mater. Contin.*, 2025.
- [14] Y. J. Zhang, Y. Xu, J. Hou, and Y. H. Song, "LMS-YOLO11n: A Lightweight Multi-Scale Weed Detection Model," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 1, pp. 1291–1300, 2025, doi: 10.14569/IJACSA.2025.01601123.
- [15] L. K. Ramadhani and B. N. Widyaningrum, "Implementation of YOLO v11 for Image-Based Litter Detection and Classification in Environmental Management Efforts," *J. Appl. Informatics Comput.*, vol. 9, no. 3, pp. 617–624, 2025.