

COMPARISON OF YOLOv11, YOLOv12, YOLOv26, OBJECT DETECTION ALGORITHMS AND ResNet50 DAN MobileNetV2 CLASSIFICATION MODELS FOR DRIVER DROWSINESS DETECTION

PERBANDINGAN ALGORITMA DETEKSI OBJEK YOLOv11, YOLOv12, YOLOv26, DAN MODEL KLASIFIKASI ResNet50 DAN MobileNetV2 PADA DETEKSI KANTUK PENGEMUDI

Kevin Anugrah Putra Hasibuan¹, Risanto Darwaman², Eka Ari Endrawati³

Program Studi Teknik Informatika, Universitas Krisnadwipayana^{1,2,3}

Kevinputrahasibuan86@gmail.com¹

ABSTRACT

The high number of traffic accidents caused by human error—specifically drowsiness or microsleep while driving—poses a serious challenge to driving safety. This study aims to analyze and compare the performance of YOLOv11, YOLOv12, and YOLOv26 algorithms, as well as ResNet50 and MobileNetV2 classification models, in detecting driver drowsiness via real-time facial image analysis. Performance evaluation was conducted using precision, recall, accuracy, and mean Average Precision (mAP) metrics. The study employed a quantitative approach with an experimental method utilizing computer vision and deep learning technologies. The dataset was obtained from Roboflow Universe and comprised 2,649 images annotated into three classes: open, close, and yawning. Research stages included dataset collection and preprocessing, model training, performance testing, evaluation based on established metrics, and system implementation using a camera to facilitate real-time detection. The results indicate that all models were capable of detecting driver conditions, albeit with varying performance levels across the evaluation metrics. YOLOv11 demonstrated superiority in supporting real-time detection, while ResNet50 and MobileNetV2 delivered strong classification performance with differing levels of efficiency. This research is expected to support the development of effective and practical drowsiness detection systems to enhance driving safety.

Keywords: Driver Drowsiness, YOLOv11, YOLOv12, YOLOv26, Computer Vision

ABSTRAK

Tingginya angka kecelakaan lalu lintas yang disebabkan oleh *human error*, khususnya kantuk atau *microsleep* saat mengemudi, menjadi permasalahan serius dalam keselamatan berkendara. Penelitian ini bertujuan menganalisis dan membandingkan kinerja algoritma YOLOv11, YOLOv12, dan YOLOv26 serta model klasifikasi ResNet50 dan MobileNetV2 dalam mendeteksi kondisi kantuk pengemudi berbasis citra wajah secara *real-time*. Evaluasi kinerja dilakukan menggunakan metrik presisi, *recall*, akurasi, dan *mean Average Precision* (mAP). Penelitian menggunakan pendekatan kuantitatif dengan metode eksperimental yang menerapkan teknologi *computer vision* dan *deep learning*. Dataset diperoleh dari Roboflow Universe dan terdiri atas 2.649 citra yang telah dianotasi ke dalam tiga kelas, yaitu *open*, *close*, dan *yawning*. Tahapan penelitian mencakup pengumpulan dan prapemrosesan dataset, pelatihan setiap model, pengujian kinerja, evaluasi berdasarkan metrik yang ditetapkan, serta implementasi sistem menggunakan kamera untuk mendukung deteksi secara *real-time*. Hasil penelitian menunjukkan bahwa seluruh model mampu mendeteksi kondisi pengemudi dengan tingkat kinerja yang berbeda pada setiap metrik evaluasi. YOLOv11 menunjukkan keunggulan dalam mendukung deteksi *real-time*, sedangkan ResNet50 dan MobileNetV2 memberikan kinerja yang baik dalam proses klasifikasi dengan tingkat efisiensi yang berbeda. Penelitian ini diharapkan dapat mendukung pengembangan sistem deteksi kantuk yang efektif dan aplikatif untuk meningkatkan keselamatan berkendara.

Kata kunci: Kantuk Pengemudi, YOLOv11, YOLOv12, YOLOv26, Computer Vision

PENDAHULUAN

Kondisi mengantuk ketika berkendara menjadi salah satu faktor yang dapat mengurangi tingkat kewaspadaan, menurunkan konsentrasi, serta meningkatkan kemungkinan terjadinya kecelakaan lalu lintas. Salah satu kondisi yang perlu mendapat perhatian adalah

microsleep, yaitu keadaan ketika pengemudi mengalami kehilangan kesadaran dalam waktu yang sangat singkat. Meskipun berlangsung sesaat, kondisi tersebut dapat menyebabkan pengemudi kehilangan kendali terhadap kendaraan dan meningkatkan risiko terhadap keselamatan. Oleh sebab itu,

dibutuhkan suatu sistem yang dapat mendeteksi tanda-tanda kantuk secara otomatis dan *real-time* sehingga peringatan dini dapat diberikan sebelum kondisi tersebut berkembang menjadi situasi berbahaya. Kemajuan teknologi *deep learning* dan *computer vision* memberikan peluang untuk menganalisis kondisi pengemudi melalui karakteristik visual pada wajah, terutama perubahan atau pola pada area mata yang dapat menjadi indikator terjadinya kantuk [1], [2].

Sejumlah pendekatan telah dikembangkan untuk mengidentifikasi kondisi kantuk pada pengemudi. Metode berbasis *Haar Cascade* yang dikombinasikan dengan *Convolutional Neural Network* (CNN), misalnya, telah dimanfaatkan untuk mengenali kejadian *microsleep* pada pengemudi [3]. Penelitian lainnya menggunakan algoritma YOLO untuk mendeteksi kondisi mata, sementara arsitektur berbasis *MobileNet* diterapkan dalam mengklasifikasikan karakteristik wajah yang menunjukkan kondisi mengantuk [4], [5]. Seiring dengan perkembangan teknologi deteksi objek, keluarga YOLO juga mengalami pengembangan melalui berbagai versi dan arsitektur yang dirancang untuk menghasilkan proses deteksi yang semakin efektif dalam lingkungan *real-time*, termasuk YOLOv11 dan YOLOv12 [6], [7].

Walaupun sejumlah penelitian telah membuktikan potensi *deep learning* dalam mendukung deteksi kantuk pengemudi, kajian yang membandingkan model deteksi objek dan model klasifikasi dengan karakteristik arsitektur yang berbeda masih diperlukan. YOLOv11, YOLOv12, dan YOLOv26 merupakan pendekatan berbasis *object detection*, sedangkan ResNet50 dan MobileNetV2 lebih berorientasi pada proses klasifikasi citra. Perbedaan pendekatan dan struktur arsitektur tersebut dapat menghasilkan kemampuan yang berbeda dalam mengenali indikator visual kantuk. Berdasarkan pertimbangan tersebut, penelitian ini berfokus pada

perbandingan kinerja YOLOv11, YOLOv12, YOLOv26, ResNet50, dan MobileNetV2 dalam mengidentifikasi kondisi kantuk pengemudi berdasarkan citra wajah.

Dataset penelitian diperoleh dari *Roboflow Universe* dan mencakup tiga kategori kondisi, yaitu *open*, *close*, dan *yawning*. YOLOv11, YOLOv12, dan YOLOv26 diterapkan untuk melakukan proses *object detection*, sedangkan ResNet50 dan MobileNetV2 digunakan untuk mengklasifikasikan citra berdasarkan kondisi yang telah ditentukan. Selanjutnya, setiap model diuji dan dievaluasi menggunakan metrik kinerja yang relevan untuk memperoleh perbandingan kemampuan secara objektif. Hasil perbandingan tersebut diharapkan dapat mengidentifikasi model yang memiliki kinerja paling optimal serta dapat dijadikan dasar dalam pengembangan sistem deteksi kantuk pengemudi secara *real-time* menggunakan kamera *webcam*.

TINJAUAN PUSTAKA

Deep Learning

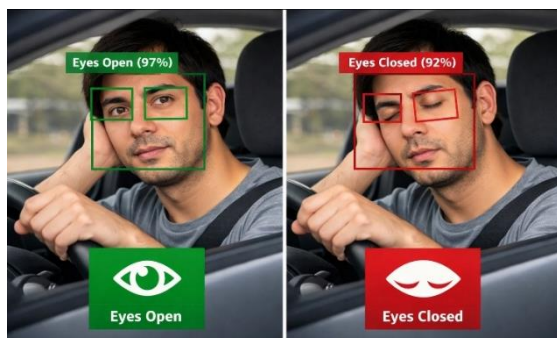
Deep Learning merupakan salah satu cabang dari *machine learning* yang memanfaatkan jaringan saraf tiruan dengan beberapa lapisan untuk mempelajari pola, fitur, dan karakteristik data secara otomatis. Pendekatan ini mampu melakukan ekstraksi fitur secara mandiri tanpa memerlukan proses perancangan fitur secara manual. Kemampuan tersebut menjadikan *deep learning* banyak digunakan dalam pengolahan data visual, khususnya citra dan video, serta berbagai aplikasi pengenalan pola dan deteksi objek. Dalam konteks keselamatan berkendara, teknologi ini dapat dimanfaatkan untuk mengidentifikasi karakteristik visual yang berkaitan dengan kondisi kantuk pengemudi [1].

Computer Vision

Computer Vision merupakan bidang teknologi yang memungkinkan komputer memperoleh, menginterpretasikan, dan

menganalisis informasi yang terkandung dalam citra maupun video secara otomatis [2]. Teknologi ini memungkinkan sistem mengenali berbagai objek dan karakteristik visual berdasarkan informasi yang terdapat pada gambar. Dalam penerapan deteksi kantuk pengemudi, *computer vision* berperan dalam mengidentifikasi wajah, menganalisis kondisi mata, serta mengenali perubahan ekspresi atau karakteristik wajah yang dapat menjadi indikator kantuk secara *real-time*. Integrasi *computer vision* dengan metode *deep learning* memungkinkan proses pemantauan kondisi pengemudi dilakukan secara lebih otomatis dan mendukung pemberian peringatan dini ketika terindikasi adanya penurunan kewaspadaan.

Saya juga dapat menyusun bagian 2.3 YOLOv11, 2.4 YOLOv12, 2.5 YOLOv26, 2.6 ResNet50, dan 2.7 MobileNetV2 dengan gaya penulisan yang sama agar landasan teori penelitian konsisten.



Gambar 1. Object Detection

Gambar 1 menunjukkan mekanisme *object detection* dalam mengidentifikasi keberadaan objek pada sebuah citra melalui penggunaan area pembatas atau *bounding box*. Dalam penelitian ini, konsep tersebut diterapkan untuk menentukan posisi objek visual yang berkaitan dengan kondisi pengemudi, sehingga informasi yang diperoleh dapat diproses dan dianalisis menggunakan model *deep learning*.

Object Detection

Object detection merupakan metode dalam *computer vision* yang bertujuan mengidentifikasi objek sekaligus menentukan lokasi dan kelasnya pada suatu

citra. Proses identifikasi umumnya direpresentasikan menggunakan *bounding box* yang mengelilingi objek yang terdeteksi. Pada sistem deteksi kantuk pengemudi, pendekatan ini dapat dimanfaatkan untuk menemukan area wajah dan mata sehingga analisis terhadap karakteristik visual pengemudi dapat dilakukan secara lebih terfokus.

Salah satu metode konvensional yang dapat digunakan adalah *Haar Cascade*, yang memiliki kemampuan mendeteksi objek dengan waktu pemrosesan relatif cepat. Namun, tingkat keberhasilannya dapat dipengaruhi oleh berbagai kondisi, seperti intensitas pencahayaan, orientasi wajah, serta jarak antara wajah dan kamera [3]. Perkembangan teknologi *deep learning* selanjutnya menghasilkan pendekatan deteksi objek yang lebih adaptif terhadap variasi kondisi citra serta mampu mendukung pemrosesan secara *real-time*.

YOLOv11

YOLOv11 merupakan salah satu arsitektur *object detection* yang dikembangkan untuk melakukan identifikasi objek secara cepat dan mendukung kebutuhan pemrosesan *real-time*. Model ini memiliki kemampuan inferensi yang efisien sehingga berpotensi diterapkan pada sistem pemantauan kondisi pengemudi yang membutuhkan respons secara cepat [11].

Dalam penelitian ini, YOLOv11 digunakan untuk mendeteksi kondisi mata berdasarkan citra wajah pengemudi. Kemampuan model dalam melakukan deteksi secara langsung dalam satu proses memungkinkan YOLOv11 digunakan pada sistem yang membutuhkan kecepatan respons dalam mengidentifikasi perubahan kondisi mata sebagai salah satu indikator kantuk.

YOLOv12

YOLOv12 merupakan salah satu pengembangan dari keluarga algoritma YOLO yang ditujukan untuk meningkatkan kemampuan dan efisiensi dalam proses

deteksi objek. Model ini dapat diterapkan dalam berbagai permasalahan *computer vision*, termasuk kebutuhan deteksi objek yang memerlukan pemrosesan secara *real-time* [12].

Pada penelitian ini, YOLOv12 digunakan sebagai salah satu model pembanding untuk mengevaluasi kemampuan deteksi kondisi mata pengemudi. Penggunaan YOLOv12 memungkinkan penelitian mengidentifikasi perbedaan kinerja dibandingkan dengan arsitektur YOLO lainnya dalam mengenali karakteristik visual yang berkaitan dengan kondisi kantuk.

YOLOv26

YOLOv26 merupakan model *object detection* yang dikembangkan dengan orientasi pada efisiensi proses inferensi dan penerapan pada sistem *real-time*, termasuk perangkat dengan keterbatasan sumber daya komputasi. Model ini menggunakan pendekatan *end-to-end* sehingga proses deteksi dapat dilakukan tanpa memerlukan tahapan *Non-Maximum Suppression* (NMS) tambahan [13].

Selain pendekatan *end-to-end*, YOLOv26 dilengkapi dengan berbagai pengembangan pada proses pelatihan dan optimasi model yang ditujukan untuk meningkatkan efisiensi serta kemampuan deteksi. Oleh karena itu, model ini digunakan sebagai salah satu alternatif pembanding dalam penelitian untuk mengetahui kemampuan YOLOv26 dalam mengidentifikasi kondisi mata pengemudi sebagai indikator deteksi kantuk.

MobileNetV2

MobileNetV2 merupakan arsitektur *Convolutional Neural Network* (CNN) yang dirancang dengan struktur jaringan ringan sehingga membutuhkan sumber daya komputasi yang relatif rendah. Karakteristik tersebut menjadikan MobileNetV2 sesuai untuk diterapkan pada berbagai aplikasi *computer vision*, khususnya sistem yang membutuhkan pemrosesan cepat dan dapat dijalankan

pada perangkat dengan keterbatasan sumber daya [14].

Dalam penelitian ini, MobileNetV2 digunakan untuk melakukan klasifikasi kondisi mata pengemudi ke dalam dua kategori, yaitu *Open* dan *Close*. Penerapan metode *transfer learning* dapat membantu meningkatkan efektivitas proses pelatihan, terutama ketika jumlah dataset yang tersedia relatif terbatas. Dengan demikian, MobileNetV2 dapat menjadi alternatif model yang efisien untuk mendukung sistem klasifikasi kondisi mata secara *real-time*.

ResNet50

ResNet50 merupakan arsitektur *Convolutional Neural Network* (CNN) yang menerapkan konsep *residual learning* melalui mekanisme *residual connection* atau *skip connection*. Mekanisme tersebut memungkinkan informasi dan gradien diteruskan melewati beberapa lapisan jaringan, sehingga membantu mengurangi permasalahan *vanishing gradient* pada jaringan dengan kedalaman yang tinggi [15].

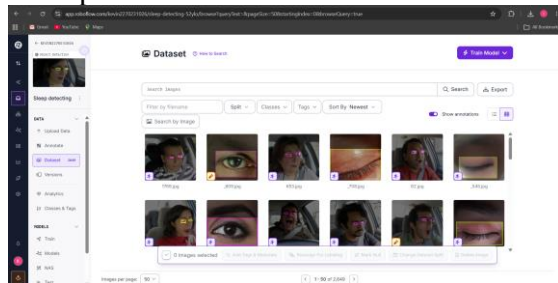
Kemampuan ResNet50 dalam melakukan ekstraksi fitur visual yang kompleks menjadikannya sesuai untuk berbagai tugas klasifikasi citra. Dalam penelitian ini, ResNet50 digunakan sebagai model klasifikasi kondisi mata pengemudi dan menjadi model pembanding terhadap MobileNetV2. Perbandingan kedua model dilakukan untuk mengetahui perbedaan kemampuan dalam mengenali kondisi mata yang dikategorikan sebagai *Open* dan *Close*.

Dataset dan Roboflow Universe

Dataset merupakan salah satu komponen fundamental dalam pengembangan model *deep learning* karena karakteristik, kualitas, jumlah, dan keragaman data dapat memengaruhi kemampuan model dalam mempelajari serta mengenali pola tertentu [9]. Dataset yang memiliki representasi kondisi objek yang beragam dapat membantu model

menghasilkan kemampuan generalisasi yang lebih baik terhadap data baru.

Dalam penelitian ini, dataset diperoleh melalui *Roboflow Universe* dan digunakan sebagai sumber data untuk pengembangan model deteksi dan klasifikasi. Dataset tersebut selanjutnya diproses dan dibagi untuk kebutuhan pelatihan, validasi, serta pengujian. Penggunaan dataset yang sama sebagai dasar evaluasi memungkinkan kinerja YOLOv11, YOLOv12, YOLOv26, ResNet50, dan MobileNetV2 dibandingkan secara lebih terstruktur berdasarkan karakteristik dan metrik evaluasi yang telah ditetapkan.



Gambar 2. Dataset Roboflow Universe

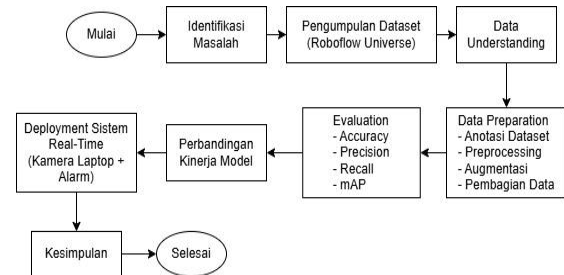
Gambar 2 memperlihatkan dataset yang digunakan sebagai sumber data dalam penelitian ini. Dataset tersebut terdiri atas citra wajah pengemudi yang telah melalui proses anotasi sehingga setiap objek atau kategori yang menjadi target deteksi dan klasifikasi dapat dikenali dengan baik oleh model *deep learning*. Proses anotasi dilakukan untuk memberikan informasi mengenai lokasi maupun label objek yang terdapat pada citra sehingga model dapat mempelajari pola visual secara lebih efektif.

Selanjutnya, dataset dibagi ke dalam tiga kelompok data, yaitu *training*, *validation*, dan *testing*. Data *training* digunakan untuk melatih model dalam mengenali karakteristik objek, data *validation* dimanfaatkan untuk memantau dan mengoptimalkan kinerja model selama proses pelatihan, sedangkan data *testing* digunakan untuk mengevaluasi kemampuan model terhadap data yang belum pernah dipelajari sebelumnya. Pembagian dataset tersebut bertujuan untuk

menghasilkan proses pengembangan model yang lebih objektif serta memastikan kemampuan generalisasi model dalam mendeteksi kondisi pengemudi secara akurat.

METODE

Metodologi Penelitian Menggunakan CRISP-DM



Gambar 3. Flowchart CRISP-DM

Penelitian pada gambar 3 menggunakan pendekatan *Cross-Industry Standard Process for Data Mining* (CRISP-DM) sebagai kerangka metodologis dalam proses pengembangan dan evaluasi sistem deteksi kantuk pengemudi. CRISP-DM dipilih karena menyediakan tahapan penelitian yang sistematis, mulai dari pemahaman permasalahan hingga penerapan model. Kerangka ini memungkinkan proses pengolahan data, pembangunan model, evaluasi, dan implementasi dilakukan secara terstruktur sehingga hasil perbandingan kinerja model dapat diperoleh secara lebih objektif.

Dalam penelitian ini, CRISP-DM terdiri atas enam tahapan utama, yaitu (1) *Business Understanding*, (2) *Data Understanding*, (3) *Data Preparation*, (4) *Modeling*, (5) *Evaluation*, dan (6) *Deployment*. Keenam tahapan tersebut diterapkan untuk menganalisis dan membandingkan kemampuan algoritma YOLOv11, YOLOv12, YOLOv26, ResNet50, dan MobileNetV2 dalam mengenali kondisi kantuk pengemudi berdasarkan citra wajah. Pendekatan ini diharapkan dapat menghasilkan proses penelitian yang terarah serta mendukung pengembangan sistem deteksi kantuk secara *real-time*.

Business Understanding

Tahap *Business Understanding* merupakan tahap awal yang bertujuan mengidentifikasi permasalahan dan menentukan tujuan penelitian berdasarkan kebutuhan yang hendak diselesaikan. Permasalahan utama yang menjadi fokus penelitian adalah risiko kecelakaan lalu lintas akibat menurunnya kewaspadaan pengemudi, khususnya karena kondisi mengantuk atau *microsleep*. Kondisi tersebut dapat menyebabkan penurunan konsentrasi, keterlambatan respons, bahkan hilangnya kendali terhadap kendaraannya sehingga berpotensi menimbulkan kecelakaan.

Berdasarkan permasalahan tersebut, penelitian ini mengembangkan dan membandingkan beberapa model *deep learning*, yaitu YOLOv11, YOLOv12, YOLOv26, ResNet50, dan MobileNetV2. Sistem yang dikembangkan diarahkan untuk mampu menganalisis kondisi mata pengemudi melalui kamera laptop atau *webcam* secara *real-time*. Kondisi mata terbuka dan tertutup digunakan sebagai salah satu indikator visual dalam mengidentifikasi tingkat kewaspadaan pengemudi.

Untuk menentukan model yang memiliki kinerja paling sesuai, dilakukan perbandingan terhadap performa masing-masing algoritma berdasarkan sejumlah metrik evaluasi. Metrik yang digunakan meliputi akurasi, presisi, *recall*, dan *mean Average Precision* (mAP). Hasil evaluasi tersebut menjadi dasar dalam menentukan model yang memiliki kemampuan terbaik untuk mendukung sistem deteksi kantuk pengemudi secara *real-time*.

Data Understanding

Tahap *Data Understanding* dilakukan untuk memahami karakteristik, struktur, kualitas, serta distribusi dataset yang digunakan dalam penelitian. Dataset diperoleh melalui platform *Roboflow Universe* dan berisi citra wajah pengemudi dengan berbagai kondisi mata. Dataset yang digunakan terdiri atas sekitar 2.649

citra yang dikelompokkan ke dalam dua kelas utama, yaitu:

1. Open, yaitu kondisi ketika mata pengemudi terbuka.
2. Close, yaitu kondisi ketika mata pengemudi tertutup.

Dataset memiliki variasi karakteristik visual yang cukup beragam untuk merepresentasikan kondisi pengemudi dalam situasi yang berbeda. Variasi tersebut meliputi perbedaan tingkat pencahayaan, sudut pengambilan gambar, jarak wajah terhadap kamera, penggunaan aksesoris seperti kacamata, serta perbedaan ekspresi wajah. Keberagaman karakteristik tersebut penting diperhatikan karena dapat memengaruhi kemampuan model dalam mengenali kondisi mata secara akurat.

Pada tahap ini juga dilakukan pemeriksaan terhadap format dataset, jumlah citra pada setiap kelas, kualitas gambar, serta kelengkapan anotasi. Proses tersebut bertujuan memastikan bahwa dataset memiliki karakteristik yang sesuai dengan kebutuhan pelatihan, validasi, dan pengujian model.

Data Preparation

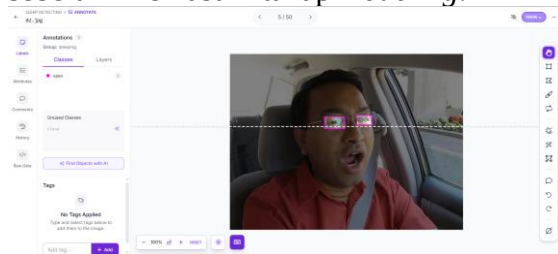
Tahap *Data Preparation* merupakan proses mempersiapkan dataset agar dapat digunakan secara optimal dalam proses pelatihan dan pengujian model. Tahap ini mencakup beberapa kegiatan, salah satunya adalah proses anotasi dataset. Anotasi diperlukan untuk memberikan informasi mengenai objek dan kelas yang terdapat pada setiap citra sehingga model dapat mempelajari karakteristik visual dari masing-masing kategori.

Anotasi Dataset

Dataset terlebih dahulu dipastikan memiliki anotasi yang sesuai dengan objek yang menjadi fokus penelitian. Citra wajah pengemudi dengan kondisi mata terbuka diberikan label Open, sedangkan citra dengan kondisi mata tertutup diberikan label Close. Anotasi tersebut menjadi informasi dasar bagi model *deep learning* dalam mempelajari perbedaan karakteristik

visual antara kondisi mata terbuka dan tertutup.

Selain proses anotasi, data juga dipersiapkan sesuai dengan kebutuhan masing-masing model. Dataset kemudian disusun ke dalam kelompok data *training*, *validation*, dan *testing*. Data *training* digunakan untuk proses pembelajaran model, data *validation* digunakan untuk memantau performa model selama pelatihan, sedangkan data *testing* digunakan untuk mengukur kemampuan model terhadap data yang belum digunakan selama proses pembelajaran. Tahap persiapan ini dilakukan untuk memastikan dataset berada dalam kondisi yang sesuai sebelum memasuki tahap *modeling*.



Gambar 4. Proses Anotasi Dataset

Gambar 4 memperlihatkan proses anotasi dataset menggunakan platform *Roboflow Universe*. Platform tersebut dimanfaatkan untuk melakukan pengelolaan, anotasi, dan penyiapan dataset sehingga data dapat digunakan dalam proses pelatihan dan pengujian model. Setelah proses anotasi selesai, dataset diekspor dalam format dan ukuran yang sesuai dengan kebutuhan masing-masing algoritma yang digunakan dalam penelitian.

Preprocessing Data

Sebelum memasuki tahap pelatihan (*training*), dataset terlebih dahulu melalui proses *preprocessing* untuk memastikan data berada dalam kondisi yang sesuai dengan kebutuhan model. Tahapan *preprocessing* meliputi:

1. Pengubahan ukuran (*resize*) gambar, yaitu menyesuaikan dimensi citra dengan ukuran masukan yang ditetapkan pada masing-masing model.
2. Normalisasi data, yaitu menyesuaikan nilai piksel agar berada pada rentang

yang sesuai sehingga proses pembelajaran model dapat berlangsung secara lebih optimal.

3. Augmentasi gambar, yaitu melakukan variasi terhadap citra melalui transformasi tertentu untuk meningkatkan keragaman data dan membantu mengurangi risiko *overfitting*.
4. Konversi format dataset, yaitu menyesuaikan struktur dan format data dengan kebutuhan algoritma yang digunakan pada tahap *modeling*.
- 5.

Pembagian Dataset

Setelah melalui proses *preprocessing*, dataset dibagi menjadi tiga kelompok, yaitu data *training*, *validation*, dan *testing*. Data *training* digunakan untuk melatih model, data *validation* digunakan untuk memantau performa model selama proses pelatihan, sedangkan data *testing* digunakan untuk mengukur kemampuan model terhadap data yang belum digunakan dalam proses pembelajaran.

Tabel 1. Pembagian Data Training, Validation, dan Testing

Jenis Data	Proporsi	Jumlah Data
Training	80%	2.119
Validation	8%	212
Testing	12%	318
Total	100%	2.649

Catatan: Jumlah data dihitung berdasarkan total 2.649 citra dan dibulatkan ke bilangan terdekat sehingga jumlah keseluruhan tetap 2.649 citra.

Export Dataset

Dataset yang telah dipersiapkan selanjutnya diekspor melalui *Roboflow Universe* dalam format yang sesuai dengan masing-masing model. Untuk YOLOv11, YOLOv12, dan YOLOv26, dataset diekspor menggunakan format YOLO sehingga dapat digunakan dalam proses pelatihan melalui *Google Colab*. Sementara itu, dataset untuk ResNet50 dan MobileNetV2 disesuaikan dengan kebutuhan klasifikasi citra menggunakan *TensorFlow* dan *Keras*.

Modeling

Tahap *Modeling* merupakan proses pembangunan dan pelatihan model *deep learning* yang digunakan untuk membandingkan kemampuan dalam mendeteksi kondisi kantuk pengemudi berdasarkan citra wajah. Penelitian ini menggunakan lima model, yaitu YOLOv11, YOLOv12, YOLOv26, ResNet50, dan MobileNetV2. Proses pelatihan seluruh model dilakukan menggunakan *Google Colab* dengan dukungan GPU Tesla T4 untuk mempercepat proses komputasi.

1. YOLOv11, YOLOv12, dan YOLOv26

YOLOv11, YOLOv12, dan YOLOv26 digunakan sebagai model *object detection* untuk mengidentifikasi lokasi dan kondisi area mata pada citra wajah pengemudi. Ketiga model tersebut dipilih karena memiliki kemampuan melakukan deteksi objek secara cepat sehingga sesuai untuk dikembangkan pada sistem pemantauan kondisi pengemudi secara *real-time*.

Dataset yang telah dianotasi melalui *Roboflow Universe* digunakan sebagai masukan dalam proses pelatihan. Parameter pelatihan yang digunakan pada ketiga model tersebut adalah:

- Epoch: 50
- Image Size: 640 × 640 piksel
- Batch Size: 16

Proses pelatihan dilakukan menggunakan framework *Ultralytics YOLO* pada *Google Colab*. Setelah proses *training* selesai, bobot model dengan performa terbaik disimpan dalam berkas *best.pt*. Bobot tersebut selanjutnya digunakan pada tahap evaluasi dan implementasi sistem deteksi secara *real-time*.

ResNet50

ResNet50 digunakan sebagai model klasifikasi citra dengan pendekatan *transfer learning*. Model ini dimanfaatkan untuk mengidentifikasi kondisi mata pengemudi ke dalam kategori *Open* dan *Close*. Citra yang telah dipersiapkan sebelumnya

digunakan sebagai masukan dalam proses pelatihan ResNet50.

Parameter pelatihan ResNet50 yang digunakan dalam penelitian ini meliputi:

- Epoch: 20
- Image Size: 224 × 224 piksel
- Batch Size: 32

Proses pelatihan dilakukan menggunakan *TensorFlow* dan *Keras* pada *Google Colab*. Sebelum proses pelatihan, citra melalui tahapan *preprocessing*, seperti pengubahan ukuran, penskalaan nilai piksel, dan augmentasi. Pendekatan *transfer learning* digunakan dengan memanfaatkan kemampuan ekstraksi fitur dari model yang telah dilatih sebelumnya. Setelah pelatihan selesai, model disimpan dalam format *model.h5* untuk digunakan pada tahap evaluasi dan pengujian.

MobileNetV2

MobileNetV2 digunakan sebagai model klasifikasi untuk mengidentifikasi kondisi mata pengemudi ke dalam dua kelas, yaitu *Open* dan *Close*. Model ini dipilih karena memiliki arsitektur yang relatif ringan sehingga berpotensi mendukung penerapan sistem dengan kebutuhan komputasi yang lebih rendah. Parameter pelatihan MobileNetV2 yang digunakan adalah:

- Epoch: 30
- Image Size: 224 × 224 piksel
- Batch Size: 32

Pelatihan dilakukan menggunakan *TensorFlow* dan *Keras* pada *Google Colab* dengan pendekatan *transfer learning*. Dataset yang telah melalui tahap *preprocessing* digunakan sebagai masukan untuk melatih model. Model yang telah selesai dilatih kemudian digunakan untuk proses evaluasi dan pengujian kemampuan klasifikasi kondisi mata pengemudi.

Evaluation

Tahap *Evaluation* dilakukan untuk mengukur dan membandingkan kemampuan masing-masing model berdasarkan data *testing* yang telah dipisahkan sebelumnya. Evaluasi bertujuan

mengetahui sejauh mana model mampu mengenali kondisi mata pengemudi secara tepat serta menentukan model yang memberikan kinerja paling optimal.

Metrik evaluasi yang digunakan dalam penelitian ini meliputi:

1. Akurasi, digunakan untuk mengukur proporsi prediksi model yang benar terhadap keseluruhan data pengujian.
2. Presisi, digunakan untuk mengetahui tingkat ketepatan model ketika memberikan prediksi terhadap suatu kelas.
3. Recall, digunakan untuk mengukur kemampuan model dalam menemukan seluruh data yang termasuk ke dalam kelas yang sebenarnya.
4. Mean Average Precision (mAP), digunakan terutama untuk mengevaluasi kemampuan model *object detection*, khususnya YOLOv11, YOLOv12, dan YOLOv26, dalam mendeteksi objek berdasarkan lokasi dan kelasnya.

Hasil dari setiap metrik kemudian dibandingkan untuk mengetahui karakteristik dan keunggulan masing-masing model. Perbandingan tersebut menjadi dasar dalam menentukan model yang memiliki kinerja paling sesuai untuk mendukung sistem deteksi kantuk pengemudi.

Deployment

Tahap *Deployment* merupakan tahap akhir dalam kerangka CRISP-DM, yaitu menerapkan model yang telah dilatih dan dievaluasi ke dalam sistem deteksi kantuk pengemudi secara *real-time*. Model dengan bobot terbaik digunakan untuk melakukan proses prediksi terhadap citra yang diperoleh secara langsung dari kamera.

Dalam penelitian ini, sistem diimplementasikan menggunakan webcam laptop sebagai perangkat untuk menangkap citra wajah pengemudi secara langsung. Citra yang diperoleh dari kamera kemudian diproses oleh model untuk mendeteksi kondisi mata. Sistem dapat mengidentifikasi kondisi mata berdasarkan kelas yang telah ditentukan, seperti *Open*

dan *Close*. Hasil deteksi tersebut selanjutnya digunakan sebagai dasar dalam mengenali indikasi kantuk pengemudi. Implementasi ini bertujuan menguji kemampuan model dalam kondisi penggunaan secara langsung sekaligus mengetahui kesesuaian hasil deteksi dengan kebutuhan sistem pemantauan kantuk secara *real-time*.

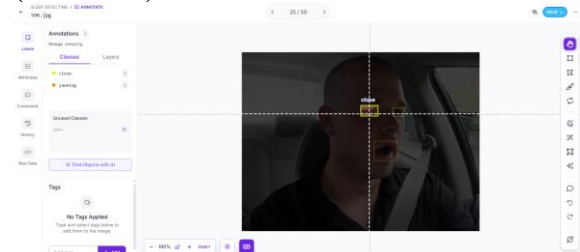
HASIL DAN PEMBAHASAN

Business Understanding

Tujuan utama penelitian ini adalah mengembangkan sistem deteksi kantuk pengemudi secara *real-time* berbasis citra wajah untuk meminimalkan risiko kecelakaan lalu lintas. Sistem memanfaatkan teknik *deep learning* dan *computer vision* untuk mengenali tiga kondisi utama pengemudi: Open (mata terbuka), Close (mata tertutup/indikator *microsleep*), dan Yawning (menguap/indikator awal kelelahan). Sistem terintegrasi dengan kamera web laptop dan dilengkapi mekanisme alarm peringatan dini yang aktif saat kondisi bahaya terdeteksi secara berkelanjutan.

Data Understanding

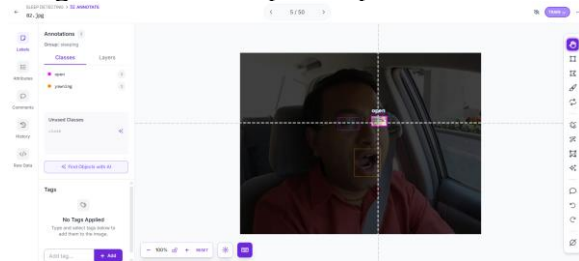
Data diperoleh dari platform Roboflow Universe dengan total 2.649 citra wajah pengemudi yang telah diberi anotasi *bounding box*. Dataset ini mencakup keragaman kondisi lingkungan seperti variasi pencahayaan (terang dan gelap), sudut pengambilan gambar, orientasi wajah, serta penggunaan aksesoris (kacamata).



Gambar 5. Anotasi_mata_tertutup

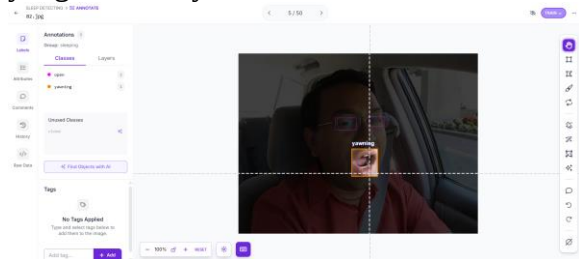
Pada Gambar 5 menunjukkan proses anotasi pada citra dengan kondisi mata tertutup (*Close*). Pada gambar tersebut, kedua mata diberikan bounding box yang

mengelilingi area mata kemudian diberi label Close. Pemberian bounding box dilakukan secara presisi mengikuti posisi mata agar model mampu mempelajari bentuk, ukuran, dan lokasi mata tertutup dengan baik selama proses pelatihan



Gambar 6. Anotasi mata_terbuka

Prosedur anotasi untuk gambar dengan kondisi mata terbuka ditunjukkan pada Gambar 6 Pada tahap ini, platform Roboflow digunakan untuk menetapkan label *Open* (Terbuka) dan membuat kotak pembatas (bounding box) di sekeliling mata yang terbuka. Untuk memastikan model dapat mempelajari fitur kondisi mata terbuka dengan tepat, kotak-kotak pembatas tersebut diposisikan secara cermat agar sesuai dengan bentuk mata yang sebenarnya.



Gambar 7. Anotasi Menguap

Prosedur anotasi untuk gambar yang menampilkan aktivitas menguap ditunjukkan pada Gambar 7 Pada tahap ini, platform Roboflow digunakan untuk membuat kotak pembatas *bounding box* di sekeliling area mulut yang terbuka lebar akibat menguap, lalu melabelinya sebagai *Yawning* (menguap). Model dapat mempelajari karakteristik tindakan menguap secara optimal karena kotak pembatas tersebut memiliki skala dan posisi yang presisi mengikuti bentuk mulut.

Menguap merupakan salah satu tanda awal kelelahan atau rasa kantuk pada pengemudi. Aktivitas menguap sering kali dikaitkan dengan penurunan tingkat

kewaspadaan saat mengemudi, meskipun hal tersebut tidak selalu berarti pengemudi telah tertidur. Kelas "Yawning" ditambahkan ke dalam dataset untuk membantu model mengenali indikator awal kelelahan sebelum terjadinya kondisi mata tertutup dalam waktu lama.

Data dibagi secara proporsional menjadi tiga subset:

- Data Pelatihan (Training): 80% (2.119 citra)
- Data Validasi (Validation): 8% (212 citra)
- Data Pengujian (Testing): 12% (318 citra)

Data Preparation

Tahap penyiapan dataset untuk digunakan dalam proses pelatihan model dikenal sebagai tahap persiapan data. Untuk memastikan dataset lengkap siap digunakan bagi model YOLOv11, YOLOv12, YOLOv26, ResNet50, dan MobileNetV2, data yang telah dianotasi dalam penelitian ini diproses menggunakan platform Roboflow. Pada tahap ini, dataset dibagi menjadi set pelatihan, validasi, dan pengujian agar setiap model dapat dilatih dan dievaluasi menggunakan data yang berbeda. Tujuan dari pembagian dataset yang tepat adalah untuk mengurangi kemungkinan *overfitting* dan menghasilkan model yang mampu melakukan generalisasi dengan baik terhadap data baru.

Dataset Split	TRAIN SET 2116 Images	VALID SET 225 Images	TEST SET 308 Images
---------------	--------------------------	-------------------------	------------------------

Gambar 8. Pembagian Dataset

Gambar 8 Pembagian dataset dilakukan pada model YOLOv11, YOLOv12, YOLOv26 ResNet50, dan MobileNetV2 secara otomatis oleh Roboflow sehingga distribusi data pada masing-masing kelas dapat dibagi secara merata. Tahap ini sangat penting karena kualitas pembagian dataset dapat mempengaruhi hasil evaluasi dan tingkat akurasi model pada saat pengujian sistem deteksi kantuk pengemudi.

Tahap persiapan data mencakup penyesuaian format anotasi dan

penyelarasan citra agar kompatibel dengan arsitektur model yang diuji:

- Format Object Detection (YOLOv11, YOLOv12, YOLOv26): Ekspor anotasi *bounding box* (*Open*, *Close*, *Yawning*) dalam standar format Ultralytics YOLO dengan resolusi masukan 640×640 piksel.
- Format Klasifikasi (ResNet50, MobileNetV2): Pra-pemrosesan data citra dengan *resizing* ke ukuran 224×224 piksel serta pengelompokan ke dalam struktur direktori berbasis kelas.

Modeling

Proses pelatihan dilakukan di lingkungan Google Colab memanfaatkan akselerasi GPU Tesla T4. Penelitian ini membandingkan 5 algoritma *deep learning*:

1. Object Detection Models:
 - *YOLOv11*, *YOLOv12*, *YOLOv26* dilatih dengan konfigurasi 50 *epochs*, *batch size* 16, dan *image size* 640.
2. Image Classification Models:
 - *ResNet50*: Dilatih selama 20 *epochs*, *batch size* 32, resolusi 224×224 .
 - *MobileNetV2*: Dilatih selama 30 *epochs*, *batch size* 32, resolusi 224×224 .

Evaluation

Pada tahap ini, semua model yang digunakan dalam penelitian yaitu *YOLOv11*, *YOLOv12*, *YOLOv26*, *ResNet50*, dan *MobileNetV2* dibandingkan satu sama lain. Tujuan perbandingan ini adalah untuk menentukan model mana yang memiliki kinerja terbaik untuk sistem deteksi kantuk pengemudi berbasis citra wajah. Evaluasi dilakukan menggunakan metrik yang sesuai dengan jenis modelnya: Akurasi, Presisi, *Recall*, dan *F1-Score* untuk model klasifikasi, serta Presisi, *Recall*, *mAP@0,5*, dan *mAP@0,5:0,95* untuk model deteksi objek.

Tabel 2. Model Deteksi Objek

Model	<i>Precisi on</i>	<i>Reca ll</i>	<i>mAP@ 0.5</i>	<i>mAP@0.5: 0.95</i>
YOLOv11	95%	98%	98%	64%

Model	<i>Precisi on</i>	<i>Reca ll</i>	<i>mAP@ 0.5</i>	<i>mAP@0.5: 0.95</i>
YOLOv12	93%	95%	97%	63%
YOLOv26	94%	95%	96%	62%

Model *YOLOv11* menunjukkan kinerja yang lebih unggul dibandingkan model deteksi objek lainnya, sebagaimana terlihat pada Tabel 4.2. Nilai Presisi mencapai 94,9%, *Recall* 98,0%, *mAP@0,5* sebesar 97,5%, dan *mAP@0,5:0,95* sebesar 63,6%. Hasil ini menunjukkan bahwa *YOLOv11* cukup efektif dalam mengidentifikasi objek pada kelas *Open* (Mata Terbuka), *Close* (Mata Tertutup), dan *Yawning* (Menguap).

Sebaliknya, *YOLOv26* mencatatkan Presisi 94,2%, *Recall* 95,4%, *mAP@0,5* sebesar 96,2%, dan *mAP@0,5:0,95* sebesar 62,1%; sementara itu, *YOLOv12* mencapai Presisi 93,1%, *Recall* 94,9%, *mAP@0,5* sebesar 96,6%, dan *mAP@0,5:0,95* sebesar 62,5%. Hasil evaluasi menunjukkan bahwa kedua model tersebut memiliki kinerja yang baik, namun masih berada di bawah *YOLOv11*, khususnya dalam hal nilai *Recall* dan *mAP*. Mengingat *YOLOv11* memberikan kinerja deteksi yang lebih akurat dan stabil untuk sistem deteksi kantuk pengemudi secara *real-time*, model ini dipilih sebagai model deteksi objek terbaik dan digunakan untuk tahap penerapan *deployment*.

Tabel 3. Model Klasifikasi Citra

Model	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
<i>ResNet50</i>	77%	74%	76%	75%
<i>MobileNetV2</i>	94%	95%	91%	92%

Tabel 3 menunjukkan bahwa model *MobileNetV2* mengungguli *ResNet50* dalam setiap metrik evaluasi. *MobileNetV2* mencapai akurasi 94%, presisi 95%, recall 91%, dan skor F1 92%. Temuan ini menunjukkan bahwa *MobileNetV2* lebih baik daripada *ResNet50* dalam mengklasifikasikan kondisi *Open* (Mata Terbuka), *Close* (Mata Tertutup), dan *Yawning* (Menguap). Di sisi lain, *ResNet50*

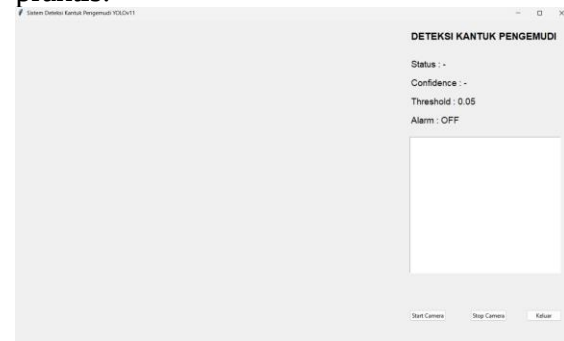
memperoleh skor F1 75%, recall 76%, akurasi 77%, dan presisi 74%. Angka-angka ini menunjukkan bahwa kinerja klasifikasi ResNet50 lebih rendah dibandingkan MobileNetV2 akibatnya, model tersebut tidak memberikan performa terbaik pada dataset penelitian ini. Berdasarkan perbandingan seluruh model, YOLOv11 merupakan model terbaik untuk pengenalan objek, sedangkan MobileNetV2 adalah model terbaik untuk klasifikasi gambar. Dibandingkan dengan YOLOv12 dan YOLOv26, YOLOv11 menunjukkan kemampuan yang lebih unggul dalam mengenali objek secara tepat dan konsisten pada kondisi Open, Close, dan Yawning, serta mencapai nilai presisi, recall, dan mAP tertinggi. Sementara itu, metrik akurasi, presisi, recall, dan skor F1 menunjukkan bahwa MobileNetV2 memiliki kinerja klasifikasi yang lebih baik daripada ResNet50. Mengingat temuan tersebut, YOLOv11 dipilih sebagai model utama untuk tahap penerapan karena memiliki kinerja deteksi terbaik dan mampu mendeteksi kantuk pengemudi secara akurat dan *real-time*. Model klasifikasi dengan kinerja terbaik dalam penelitian ini adalah MobileNetV2, yang mengungguli ResNet50 dalam mengidentifikasi fitur-fitur pada foto wajah pengemudi.

Deployment

Pada tahap *deployment*, model YOLOv11 yang telah melalui proses pelatihan dan evaluasi diintegrasikan ke dalam aplikasi desktop yang dikembangkan menggunakan bahasa pemrograman Python. Aplikasi tersebut memanfaatkan pustaka Tkinter sebagai antarmuka pengguna (*User Interface* atau UI), sehingga pengguna dapat berinteraksi dengan sistem secara lebih mudah.

Sistem memanfaatkan kamera atau *webcam* laptop sebagai perangkat untuk menangkap citra wajah pengemudi secara langsung. Citra yang diperoleh kemudian diproses oleh model YOLOv11 untuk mengidentifikasi kondisi mata sebagai

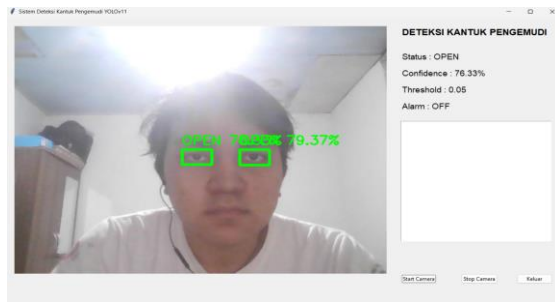
salah satu indikator kantuk. Sistem dirancang untuk melakukan proses deteksi secara *real-time* dan memberikan peringatan kepada pengemudi ketika kondisi yang mengindikasikan kantuk berhasil terdeteksi. Implementasi ini bertujuan menguji kemampuan model dalam lingkungan penggunaan secara langsung serta menunjukkan penerapan hasil pelatihan model ke dalam sistem deteksi kantuk yang dapat digunakan secara praktis.



Gambar 9. Tampilan Awal Sistem

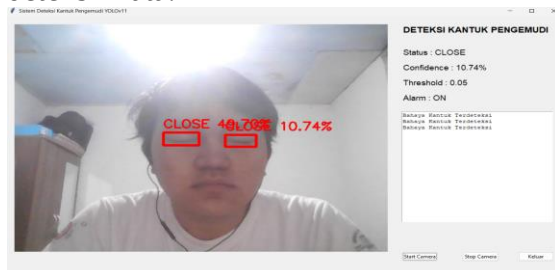
Gambar 9 memperlihatkan tampilan awal antarmuka sistem deteksi kantuk pengemudi yang dikembangkan menggunakan model YOLOv11. Antarmuka sistem dirancang untuk menampilkan berbagai informasi yang berkaitan dengan proses deteksi, meliputi status hasil deteksi, tingkat kepercayaan (*confidence score*), nilai ambang kepercayaan (*confidence threshold*), serta status peringatan. Selain panel informasi, sistem menyediakan area tampilan kamera yang digunakan untuk menampilkan hasil deteksi secara *real-time*.

Untuk memudahkan pengoperasian, antarmuka dilengkapi dengan beberapa tombol kendali, yaitu Mulai Kamera, Hentikan Kamera, dan Keluar. Tombol tersebut digunakan untuk menjalankan kamera, menghentikan proses pemantauan, dan menutup aplikasi. Pada kondisi awal sebagaimana ditunjukkan pada Gambar 4, area tampilan video masih dalam keadaan kosong karena kamera belum diaktifkan, sehingga proses deteksi belum berlangsung.



Gambar 10. Deteksi Kondisi Mata Terbuka (OPEN)

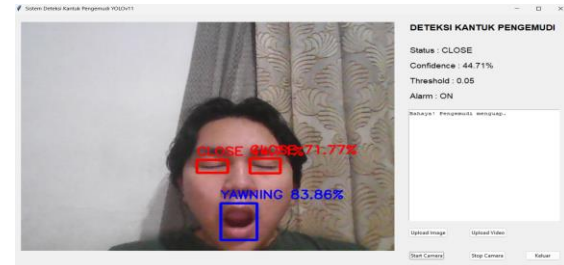
Gambar 10 menunjukkan hasil deteksi ketika sistem mengidentifikasi kondisi mata pengemudi dalam keadaan terbuka (*OPEN*). Pada tampilan tersebut, sistem menampilkan nilai tingkat keyakinan (*confidence score*) dari hasil prediksi model serta *bounding box* berwarna hijau yang menandai area wajah atau objek yang berhasil terdeteksi. Kondisi *OPEN* menunjukkan bahwa pengemudi berada dalam keadaan waspada atau sadar, sehingga sistem tidak mengaktifkan peringatan. Dengan demikian, status pemantauan yang ditampilkan pada antarmuka berada pada kondisi *OPEN*, yang mengindikasikan bahwa tidak terdapat tanda kantuk berdasarkan hasil deteksi mata.



Gambar 11. Deteksi Kondisi Mata Tertutup (CLOSE)

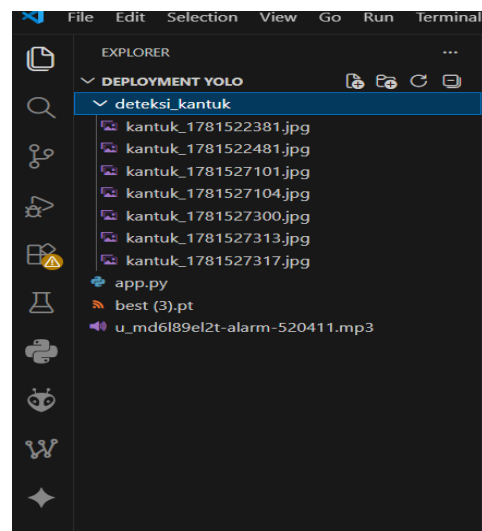
Gambar 11 memperlihatkan hasil deteksi ketika sistem mengidentifikasi kondisi mata pengemudi dalam keadaan tertutup (*CLOSE*). Pada kondisi tersebut, sistem menampilkan nilai tingkat keyakinan (*confidence score*) dari hasil prediksi model serta memberikan *bounding box* berwarna merah pada area wajah yang berhasil dikenali. Kondisi *CLOSE* digunakan sebagai indikator awal adanya penurunan kewaspadaan pengemudi. Oleh karena itu, sistem melakukan pemantauan secara berkelanjutan terhadap kondisi mata

untuk mengidentifikasi kemungkinan munculnya rasa kantuk. Deteksi mata tertutup secara berulang dapat menjadi dasar bagi sistem untuk memberikan peringatan dini guna meningkatkan kewaspadaan pengemudi selama berkendara.



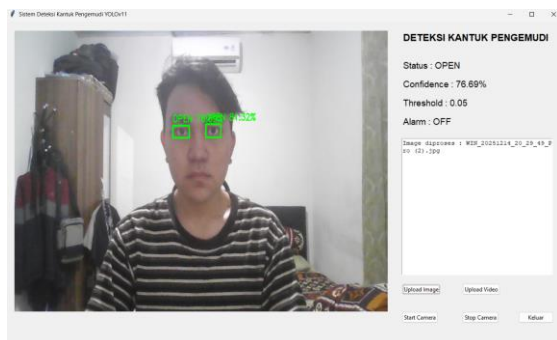
Gambar 12. Gambar Peringatan menguap

Pada sistem yang dikembangkan Gambar 12, selain mendeteksi kondisi mata terbuka (*open*) dan mata tertutup (*close*), sistem juga mendeteksi kondisi menguap (*yawning*). Deteksi menguap dilakukan menggunakan model YOLOv11 yang telah dilatih untuk mengenali objek berupa mulut yang terbuka lebar. Apabila model mendeteksi kelas yawning secara berturut-turut hingga mencapai ambang batas yang telah ditentukan, sistem akan mengaktifkan alarm sebagai peringatan kepada pengemudi. Hal ini dilakukan karena menguap merupakan salah satu indikator awal terjadinya kelelahan dan kantuk saat mengemudi.



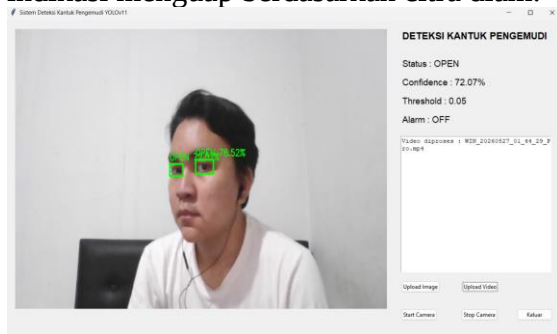
Gambar 13. Penyimpanan Hasil Deteksi
Ketika sistem berhasil mengidentifikasi kondisi kantuk, hasil

deteksi akan secara otomatis disimpan sebagaimana ditampilkan pada Gambar 13. Hasil tersebut tersimpan dalam folder “detect_drowsiness” dalam bentuk foto yang telah terdeteksi. Setiap foto diberi nama file berdasarkan waktu saat proses deteksi berlangsung. Fitur ini dirancang untuk mendokumentasikan setiap kejadian kantuk sehingga data hasil deteksi dapat digunakan sebagai bahan pemeriksaan dan analisis lebih lanjut.



Gambar 14. Fitur Upload Image

Proses deteksi melalui fitur Unggah Gambar ditampilkan pada Gambar 9. Melalui fitur ini, pengguna dapat memilih dan mengunggah file gambar yang tersimpan pada komputer untuk kemudian diproses oleh sistem menggunakan model YOLOv11 yang telah dilatih. Hasil deteksi ditampilkan dalam bentuk label kelas Terbuka (Open), Tertutup (Closed), atau Menguap (Yawning), disertai nilai tingkat kepercayaan (*confidence score*) dan kotak pembatas (*bounding box*) pada area mata atau mulut yang teridentifikasi. Fitur ini digunakan untuk menguji kemampuan model dalam mengenali kondisi mata dan indikasi menguap berdasarkan citra diam.



Gambar 15. Fitur Upload Video

Proses pendeteksian melalui fitur Unggah Video ditampilkan pada Gambar 10. Pada fitur ini, pengguna dapat memilih file video yang menampilkan wajah pengemudi untuk diproses oleh sistem. Selanjutnya, model YOLOv11 menganalisis setiap *frame* pada video secara berurutan dan menampilkan hasil deteksi pada masing-masing *frame*. Selama video diputar, pengguna dapat mengamati hasil identifikasi kondisi mata melalui tampilan kotak pembatas (*bounding box*), label kelas, serta nilai tingkat kepercayaan (*confidence score*) pada setiap *frame* yang dianalisis.

KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian terkait perbandingan kinerja model *deep learning* dalam mendeteksi kondisi kantuk pengemudi berbasis citra wajah, dapat disimpulkan bahwa ResNet50 dan MobileNetV2 telah berhasil diterapkan untuk melakukan klasifikasi citra, sedangkan YOLOv11, YOLOv12, dan YOLOv26 mampu digunakan untuk melakukan deteksi objek secara *real-time* pada tiga kelas, yaitu Open, Close, dan Yawning. Seluruh data penelitian melalui tahapan anotasi dan pengolahan menggunakan Roboflow, kemudian proses pelatihan model dilakukan dengan memanfaatkan Google Colab yang didukung GPU Tesla T4.

Berdasarkan hasil pengujian dan evaluasi performa, YOLOv11 menunjukkan kinerja paling optimal dibandingkan model lainnya, terutama dari aspek tingkat presisi dan kestabilan dalam proses deteksi. Oleh karena itu, YOLOv11 dipilih untuk diintegrasikan ke dalam aplikasi deteksi kantuk pengemudi secara *real-time* berbasis webcam. Namun, kinerja sistem masih dapat mengalami perubahan akibat kondisi lingkungan dan karakteristik pengambilan citra, seperti intensitas pencahayaan, jarak antara kamera dan pengemudi, serta orientasi atau posisi wajah. Oleh sebab itu, penelitian selanjutnya disarankan untuk memperluas

dan meningkatkan keberagaman dataset, khususnya dengan menambahkan variasi kondisi pencahayaan, sudut pandang wajah, jarak pengambilan gambar, dan kondisi lingkungan lainnya. Pengembangan tersebut diharapkan dapat meningkatkan kemampuan generalisasi serta keandalan sistem dalam mendeteksi kantuk pada berbagai kondisi penggunaan.

DAFTAR PUSTAKA

- Achmad, S. A. Y. A., Suryadhi, & Subur, J. (2023). Rancang bangun alat pendeteksi kantuk pada kendaraan roda empat dengan metode Haar Cascade. *Journal of Science and Engineering*, 6(2), 69–78. <https://doi.org/10.33387/josae.v6i2.6898>
- Albadawi, Y., Takruri, M., & Awad, M. (2022). A review of recent developments in driver drowsiness detection systems. *Sensors*, 22(5), 2069. <https://doi.org/10.3390/s22052069>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). IEEE. <https://doi.org/10.1109/CVPR.2016.90>
- Indra, G., Saputri, H. H., & Aziza, N. A. (2025). Penerapan MobileNet untuk klasifikasi kondisi wajah mengantuk pada citra wajah. *Journal of Informatics and Communication Technology (JICT)*, 7(1), 189–197. <https://doi.org/10.52661/jict.v7i1.413>
- Iriyanti, A. S., Hidayat, M. A. N., Hakim, J., Ndala, S., & Sasono, D. S. (2026). Perbandingan efektivitas arsitektur CNN pada sistem klasifikasi kelelahan pada wajah pengemudi. *Journal of Artificial Intelligence and Digital Business (RIGGS)*, 4(4), 15319–15326. <https://doi.org/10.31004/riggs.v4i4.6937>
- Jocher, G., Qiu, J., Liu, M., Lyu, S., Akyon, F. C., & Kalfaoglu, M. E. (2026). *Ultralytics YOLO26: Unified real-time end-to-end vision models*. arXiv. <https://doi.org/10.48550/arXiv.2606.03748>
- Kurniawan, A. R., Burdiarso, Z., & Ardianto, E. (2025). Implementasi algoritma YOLO untuk deteksi jumlah pengunjung di dalam gedung. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(2), 2944–2948. <https://doi.org/10.36040/jati.v9i2.13271>
- Lubis, N., Siambaton, M. Z., & Aulia, R. (2024). Implementasi algoritma deep learning pada aplikasi speech to text online dengan metode Recurrent Neural Network (RNN). *SUDO Jurnal Teknik Informatika*, 3(3), 113–126. <https://doi.org/10.56211/sudo.v3i3.583>
- Nuraeni, E., Sunardi, & Fadlil, A. (2026). Deteksi kelelahan pengemudi berdasarkan ekspresi wajah EAR dan MAR menggunakan 1DCNN-LSTM pada dataset NITYMED. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 10(1), 1730–1737. <https://doi.org/10.36040/jati.v10i1.17138>
- Ruseno, N., Miswanto, & Shufiyah, K. N. (2026). Face landmark-based drowsiness warning system for drivers in intelligent transport system to reduce accidents with YOLOv11. *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, 14(1), 65–72. <https://doi.org/10.33558/piksel.v14i1.12053>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*

- Pattern Recognition (CVPR)* (pp. 4510–4520). IEEE.
<https://doi.org/10.1109/CVPR.2018.00474>
- Saputri, R. S., Apriliani, A., Amar, R. S., & Astri, L. Y. (2025). Deteksi dini microsleep pada pengemudi kendaraan roda empat melalui citra mata menggunakan algoritma YOLOv8. *Jurnal Processor*, 20(2), 165–179.
<https://doi.org/10.33998/processor.2025.20.2.2529>
- Suradi, A. A. M., Alam, S., Mushaf, M., Rasyid, M. F., & Djafar, I. (2023). Sistem deteksi kantuk pengemudi mobil berdasarkan analisis rasio mata menggunakan computer vision. *JUKI: Jurnal Komputer dan Informatika*, 5(2), 222–230.
<https://doi.org/10.53842/juki.v5i2.269>
- Tian, Y., Ye, Q., & Doermann, D. (2025). YOLOv12: Attention-centric real-time object detectors. arXiv.
<https://doi.org/10.48550/arXiv.2502.12524>
- Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7464–7475). IEEE.
<https://doi.org/10.1109/CVPR52729.2023.00721>