

**IMPLEMENTASI SISTEM DETEKSI RASA KANTUK PADA
PENGEMUDI KENDARAAN RODA EMPAT
MENGUNAKAN OPENCV**

SKRIPSI

DISUSUN OLEH

FARHAN RIZQI MA'AJID

NPM. 2109020152



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025**

**IMPLEMENTASI SISTEM DETEKSI RASA KANTUK PADA
PENGEMUDI KENDARAAN RODA EMPAT
MENGUNAKAN OPENCV**

SKRIPSI

**Ditujukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada
Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas
Muhammadiyah Sumatra Utara**

**FARHAN RIZQI MA'AJID
NPM : 2109020152**

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATRA UTARA
MEDAN
2025**

LEMBAR PENGESAHAN

Judul Skripsi : IMPLEMENTASI SISTEM DETEKSI RASA KANTUK
PADA PENGEMUDI KENDARAAN RODA EMPAT
MENGUNAKAN OPENCV
Nama Mahasiswa : Farhan Rizqi Ma'ajid
NPM : 2109020152
Program Studi : Teknologi Informasi

Menyetujui
Komis Pembimbing


(Dr. Al-Khowarizmi, S.Kom., M.Kom)
NIDN. 0127099201

Ketua Program Studi


(Fatma Sari Hutagalung, S.Kom., M.Kom.)
NIDN. 0117019301

Dekan


(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

IMPLEMENTASI SISTEM DETEKSI RASA KANTUK PADA PENGEMUDI KENDARAAN RODA EMPAT MENGGUNAKAN OPENCV

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Juni 2025

Yang membuat pernyataan



Farhan Rizqi Ma'ajid

NPM. 2109020152

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Farhan Rizqi Ma'ajid
NPM : 2109020152
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**IMPLEMENTASI SISTEM DETEKSI RASA KANTUK PADA
PENGEMUDI KENDARAAN RODA EMPAT MENGGUNAKAN OPENCV**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Juni 2025

Yang membuat pernyataan



Farhan Rizqi Ma'ajid

NPM. 2109020152

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Farhan Rizqi Ma'ajid
Tempat dan Tanggal Lahir : Medan, 22 Januari 2004
Alamat Rumah : Jl. Cempaka No.56, Tj. Gusta, Kota Medan
Telepon/HP : 0895 6119 15258
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD	: SDS IKAL MEDAN	TAMAT : 2015
SMP	: SMPIT AL FITYAN MEDAN	TAMAT : 2018
SMA	: MAN 2 MODEL MEDAN	TAMAT : 2021

KATA PENGANTAR



Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “IMPLEMENTASI SISTEM DETEKSI RASA KANTUK PADA PENGEMUDI KENDARAAN RODA EMPAT MENGGUNAKAN OPENCV” ini dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknologi Informasi, Fakultas Ilmu Komputer & Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara.

Penulisan skripsi ini tidak lepas dari bantuan, dukungan, serta bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom. Ketua Program Studi Teknologi Informasi .
4. Bapak Mhd. Basri, S.Si, M.Kom Sekretaris Program Studi Teknologi Informasi.

5. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom selaku pembimbing skripsi yang telah memberikan arahan, saran, dan bimbingan dengan penuh kesabaran dan ketulusan.
6. Ibunda Enni Rahma Hasibuan yang telah memberikan dukungan dari segi moral maupun materiil serta menjadi motivasi penulis untuk dapat menyelesaikan tugas akhir ini
7. Seluruh Bapak dan Ibu Dosen di Jurusan Teknologi Informasi Fakultas Ilmu Komputer & Teknologi Informasi Universitas Muhammdiyah Sumatera Utara atas ilmu dan didikannya selama perkuliahan.
8. Sahabat dan teman-teman seperjuangan jurusan teknologi informasi UMSU
9. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa yang akan datang. Akhir kata, penulis berharap skripsi ini dapat bermanfaat bagi semua pihak yang berkepentingan, khususnya bagi perkembangan ilmu pengetahuan di bidang Nama Bidang Ilmu.

IMPLEMENTASI SISTEM DETEKSI RASA KANTUK PADA PENGEMUDI KENDARAAN RODA EMPAT MENGGUNAKAN OPENCV

ABSTRAK

Kantuk saat mengemudi merupakan penyebab signifikan kecelakaan lalu lintas, terutama pada perjalanan jarak jauh. Penelitian ini bertujuan mengembangkan sistem deteksi kantuk pengemudi secara real-time yang praktis dan ekonomis tanpa menggunakan sensor tambahan. Sistem ini menggunakan metode Eye Aspect Ratio (EAR) untuk menghitung rasio keterbukaan mata, model Convolutional Neural Network (CNN) untuk mengklasifikasikan kondisi mata, serta MediaPipe FaceMesh sebagai pengganti Haar Cascade untuk mendeteksi landmark wajah secara lebih stabil. Dataset yang digunakan berasal dari Kaggle dan terdiri dari gambar mata terbuka dan tertutup yang telah diproses melalui normalisasi dan augmentasi. Sistem awalnya dirancang sebagai aplikasi desktop berbasis Tkinter, namun dalam proses pengembangan dialihkan menjadi aplikasi web menggunakan Flask untuk mendukung fleksibilitas dan kemudahan akses lintas perangkat. Deteksi kantuk dilakukan ketika nilai EAR di bawah ambang batas dan CNN mengklasifikasikan mata sebagai tertutup dalam beberapa frame berturut-turut, yang kemudian memicu peringatan suara secara otomatis. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi kondisi kantuk dengan akurasi tinggi dan respons cepat. Penelitian ini memberikan solusi non-invasif untuk meningkatkan keselamatan berkendara dengan pendekatan visual yang efisien dan mudah diintegrasikan.

Kata Kunci: Deteksi kantuk, EAR, MediaPipe FaceMesh, CNN, Flask, OpenCV.

IMPLEMENTASI SISTEM DETEKSI RASA KANTUK PADA PENGEMUDI KENDARAAN RODA EMPAT MENGGUNAKAN OPENCV

ABSTRACT

Drowsiness while driving is a major cause of traffic accidents, especially during long-distance or nighttime travel. This study aims to develop a real-time driver drowsiness detection system that is practical and cost-effective without relying on additional hardware sensors. The system employs the Eye Aspect Ratio (EAR) method to calculate eye openness, a Convolutional Neural Network (CNN) to classify eye states, and MediaPipe FaceMesh as a replacement for Haar Cascade to improve the stability of facial landmark detection. The dataset used was sourced from Kaggle, consisting of open and closed eye images processed through normalization and augmentation. Initially developed as a desktop application using Tkinter, the system was later implemented as a web application using the Flask framework to enhance flexibility and cross-device accessibility. Drowsiness is detected when the EAR falls below a certain threshold and the CNN classifies both eyes as closed for several consecutive frames, triggering an automatic sound alert. Testing results show that the system accurately detects drowsy conditions with fast response times. This research presents a non-invasive solution for improving driving safety through an efficient and easily integrable visual-based detection approach.

Keywords: Drowsiness detection, EAR, MediaPipe FaceMesh, CNN, Flask, OpenCV.

DAFTAR ISI

HALAMAN PENGESAHAN	iii
PERNYATAAN ORISINALITAS.....	iv
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
RIWAYAT HIDUP	vi
KATA PENGANTAR.....	vii
ABSTRAK	ix
ABSTRACT	x
DAFTAR ISI.....	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	6
1.3 Batasan Masalah	7
1.4 Tujuan Penelitian	8
1.5 Manfaat Penelitian	8
BAB II LANDASAN TEORI	10
2.1 Mata dan Kantuk.....	10
2.2 Pengolahan Citra Digital	10
2.3 Computer Vision.....	12
2.4 OpenCV	13
2.5 Eye Aspect Ratio (EAR).....	13
2.6 Haar Cascade	14
2.7 Deep Learning	15
2.8 Convolutional Neural Networks (CNN).....	16
2.9 TensorFlow dan Keras.....	17
2.10 Kantuk Pada Pengemudi.....	18
2.11 Sistem Peringatan Kantuk.....	20
2.12 Penelitian Terkait.....	20

BAB III METODOLOGI PENELITIAN	24
3.1 Metode Penelitian	24
3.1.1 Karakteristik Metode Observasi	25
3.2 Alur Penelitian	26
3.2.1 Identifikasi Masalah	26
3.2.2 Studi Literatur	27
3.2.3 Pengumpulan Data	28
3.2.4 Perancangan Sistem	30
3.2.5 Implementasi Sistem	31
3.2.6 Pengujian dan Evaluasi	32
3.3 Perancangan Sistem	32
3.3.1 Pengambilan gambar	33
3.3.2 Deteksi Wajah dan Mata	34
3.3.3 Perhitungan Tingkat Kantuk	34
3.3.4 Klasifikasi Mata dan Pembangunan Model CNN	35
3.3.5 Peringatan Dini	38
3.3.6 Evaluasi dan Pengujian Model serta Kinerja Sistem	39
3.4 Desain Implementasi Sistem pada Kendaraan Roda Empat	41
3.5 Kebutuhan Hardware dan Software	43
3.5.1 Kebutuhan Hardware	43
3.5.2 Kebutuhan Software	43
BAB IV HASIL DAN PEMBAHASAN	45
4.1 Implementasi Sistem	45
4.1.1 Alur Sistem Deteksi	45
4.1.2 Struktur Implementasi	46
4.1.3 Kode Inti yang Menunjukkan Integrasi EAR + CNN	46
4.1.4 Arsitektur Endpoint Aplikasi	47
4.2 Pembangunan Model CNN	47
4.2.1 Prapemrosesan Data	47
4.2.2 Perancangan Arsitektur CNN	52
4.2.3 Pelatihan Model CNN	56
4.2.4 Evaluasi Model CNN	61

4.3 Implementasi dan Uji Coba Sistem	66
4.3.1 Integrasi Sistem	66
4.3.2 Antarmuka Web.....	66
4.3.3 Hasil Uji Coba	67
DAFTAR PUSTAKA	74

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	20
Tabel 3.1 Kebutuhan Hardware	43
Table 4.1 Kode Logika Utama Deteksi Kantuk	46
Tabel 4.2 Arsitektur Endpoint.....	47
Table 4.3 Pembagian Dataset	48
Tabel 4.4 Pembagian Dataset	49
Table 4.5 Augmentasi dan Normalisasi Data.....	50
Table 4.6 Pemanggilan Data	50
Tabel 4.7 Pemanggilan Data	51
Table 4.8 Struktur Model CNN.....	53
Tabel 4.9 Kode Implementasi Arsitektur	53
Table 4.10 Kode Implementasi Arsitektur	53
Table 4.11 Kode Kompilasi Model	54
Tabel 4.12 Kode Program	55
Table 4.13 Kode Kompilasi Model.....	57
Tabel 4.14 Kode Konfigurasi Callback.....	57
Table 4.15 Kode Pelatihan	58
Table 4.16 Kode Visualisasi Hasil Pelatihan	60
Table 4.17 Kode Evaluasi Akurasi Terhadap Data Uji.....	62
Table 4.18 Kode Prediksi Terhadap Data Uji	63
Table 4.19 Kode Confusion Matrix	63
Table 4.20 Kode Classification Report	65

DAFTAR GAMBAR

Gambar 2.1 Konsep Pengolahan Citra Digital	11
Gambar 2.2 Konsep Computer Vision	12
Gambar 2.3 Konsep OpenCV	13
Gambar 2.5 Eye Aspect Ratio untuk Mata Terbuka dan Mata Tertutup	14
Gambar 2.6 Konsep Haar Cascade.....	15
Gambar 2.7 Konsep Deep Learning.....	16
Gambar 2.8 Konsep Convolutional Neural Network (CNN).....	17
Gambar 3.1 Diagram Alur Penelitian.....	26
Gambar 3.2 Mata Terbuka	29
Gambar 3.3 Mata Tertutup.....	30
Gambar 3.4 Diagram Perancangan Sistem.....	33
Gambar 3.5 Ilustrasi Implementasi Sistem Deteksi Kantuk pada Kendaraan	42
Gambar 4.1 Visualisasi Sample Gambar	52
Gambar 4.2 Ringkasan Arsitektur Model	56
Gambar 4.3 Hasil Pelatihan.....	59
Gambar 4.4 Visualisasi Hasil Gambar	61
Gambar 4.5 Hasil Evaluasi Akurasi Terhadap Data Uji	62
Gambar 4.6 Hasil Confusion Matrix	64
Gambar 4.7 Hasil Classification Report.....	65
Gambar 4.8 Halaman Utama Web	67
Gambar 4.9 Kondisi Terjaga (Normal) pada Pengujian Langsung di Kendaraan	68
Gambar 4.10 Kondisi Terjaga (Normal) pada Tampilan Web Mobile	69
Gambar 4.11 Kondisi Terjaga (Normal) pada Tampilan Web Desktop	69
Gambar 4.12 Deteksi Kantuk Aktif pada Pengujian Langsung di Kendaraan.....	70
Gambar 4.13 Deteksi Kantuk Aktif pada Tampilan Web Mobile	71
Gambar 4.14 Deteksi Kantuk Aktif pada Tampilan Web Desktop.....	71

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Teknologi pengolahan citra digital telah mengalami perkembangan yang pesat dan diterapkan dalam berbagai bidang, termasuk keselamatan berkendara. Salah satu metode yang banyak digunakan adalah *Eye Aspect Ratio* (EAR) dan *Haar Cascade* untuk mendeteksi kondisi mata pengemudi. EAR digunakan untuk menghitung rasio antara tinggi dan lebar mata guna mendeteksi apakah mata pengemudi terbuka atau tertutup (Suradi et al., 2023). Di sisi lain, *Haar Cascade* yang diimplementasikan dalam *OpenCV* merupakan algoritma yang digunakan secara luas untuk pendeteksian fitur wajah dan mata secara *real-time* (Yulina et al., 2021). Kedua teknologi ini menawarkan solusi praktis dan ekonomis karena tidak memerlukan perangkat keras tambahan seperti sensor fisiologis yang mahal (Firdaus et al., 2023). Dengan demikian, penggunaan teknologi ini memberikan alternatif yang efisien dan hemat biaya dalam deteksi kantuk pengemudi.

Selain pengolahan citra digital, machine learning berbasis *Convolutional Neural Networks* (CNN) juga telah diterapkan dalam penelitian ini untuk melakukan klasifikasi kondisi mata pengemudi. CNN, yang dikenal sebagai algoritma *deep learning* yang efektif dalam analisis gambar, dilatih menggunakan *dataset* dari Kaggle yang berisi gambar mata terbuka dan tertutup (Iswahyudi et al., 2024). Kombinasi antara pengolahan citra dan *machine learning* ini diharapkan mampu mendeteksi kantuk pengemudi secara lebih akurat dan efisien. Teknologi ini memberikan alternatif yang lebih canggih dan responsif dalam upaya

meningkatkan keselamatan berkendara (Suyuti, 2023). Penerapan CNN dalam sistem ini berpotensi untuk memberikan hasil yang lebih presisi dan cepat dalam mendeteksi kondisi pengemudi yang mengantuk.

Selain pada sistem pendeteksian kantuk, teknologi deteksi kantuk berbasis citra juga telah berhasil diimplementasikan secara nyata dalam kendaraan produksi massal. Salah satu contoh paling awal dan menonjol adalah sistem *Driver Monitoring System* (DMS) yang diperkenalkan oleh Toyota pada model Lexus GS 450h tahun 2006 di Jepang. Sistem ini menggunakan kamera CCD dan sensor LED inframerah yang dipasang di kolom kemudi untuk memantau posisi kepala dan kondisi mata pengemudi secara *real-time*, bahkan dalam kondisi pencahayaan rendah (Toyota Global, 2006). Jika pengemudi tidak memperhatikan jalan saat risiko tabrakan terdeteksi, sistem akan memberikan peringatan visual dan suara. Apabila tidak ada respons, sistem secara otomatis akan mengaktifkan rem untuk mengurangi dampak kecelakaan (Lexus UK, 2016).

Kantuk pengemudi telah menjadi salah satu penyebab utama kecelakaan lalu lintas di seluruh dunia. Menurut laporan National Sleep Foundation, sekitar 32% pengemudi melaporkan pernah mengalami kantuk saat mengemudi setidaknya sekali dalam sebulan, dan 25% kecelakaan lalu lintas tahunan diakibatkan oleh kantuk (Suradi et al., 2023). Kantuk dapat menyebabkan *microsleep*, yaitu kondisi tidur singkat yang tidak disadari, yang menyebabkan pengemudi kehilangan kendali kendaraan dalam hitungan detik. Oleh karena itu, deteksi dini kantuk menjadi sangat penting untuk meminimalisir risiko kecelakaan (Firdaus et al., 2023). Hal ini menunjukkan urgensi dari penerapan sistem deteksi kantuk untuk meningkatkan keselamatan berkendara di jalan raya.

Kecelakaan lalu lintas menjadi salah satu penyebab kematian terbesar di dunia, dengan kantuk sebagai faktor utama yang mengurangi kewaspadaan pengemudi (Thoriq et al., 2024). Berbagai metode telah dikembangkan untuk mendeteksi kantuk, termasuk penggunaan *Eye Aspect Ratio* (EAR), yang menghitung rasio jarak antara kelopak mata atas dan bawah sebagai indikator kantuk pengemudi (Thoriq et al., 2024). Teknologi ini dapat diimplementasikan secara *real-time* menggunakan pustaka *OpenCV* tanpa perlu perangkat keras tambahan, sehingga lebih efisien dan praktis untuk berbagai jenis kendaraan (Fauzi et al., 2024). Penerapan EAR dalam sistem ini memberikan solusi praktis untuk deteksi kantuk pengemudi dengan biaya rendah.

Meskipun beberapa sistem deteksi menggunakan sensor fisiologis seperti EEG (*Electroencephalography*) memberikan hasil yang akurat, pendekatan ini dinilai mahal dan tidak nyaman bagi pengemudi, terutama untuk perjalanan jarak jauh (Zebua et al., 2024). Sebagai alternatif yang lebih murah dan nyaman, teknologi pengolahan citra berbasis algoritma *Haar Cascade* dalam *OpenCV* telah dikembangkan untuk mendeteksi kantuk melalui pengenalan pola mata. Algoritma ini memungkinkan sistem untuk mengidentifikasi kondisi mata pengemudi, apakah terbuka atau tertutup, dan memberikan peringatan dini ketika tanda-tanda kantuk terdeteksi (Fauzi et al., 2024). Dengan teknologi ini, sistem deteksi kantuk dapat diterapkan secara lebih luas tanpa membutuhkan perangkat keras tambahan yang mahal.

Sebagai pelengkap keabsahan pendekatan berbasis visual, beberapa penelitian juga menunjukkan bahwa indikator kantuk yang bersumber dari karakteristik mata memiliki dasar biologis yang kuat. Studi oleh Ftouni et al. (2013) menunjukkan

bahwa perubahan kondisi fisiologis seperti frekuensi kedipan, durasi penutupan mata (*blink duration*), serta rasio amplitudo terhadap kecepatan (AVR) sangat berkorelasi dengan tingkat kantuk seseorang. Indikator-indikator ini terbukti mengikuti ritme homeostatik dan sirkadian, yang artinya meningkat seiring lamanya waktu terjaga dan mengikuti pola jam biologis tubuh. Selain itu, hasil penelitian ini juga menunjukkan korelasi yang tinggi antara indikator visual mata tersebut dengan performa kognitif, aktivitas EEG, serta hasil uji kewaspadaan seperti *Karolinska Sleepiness Scale* (KSS) dan *Psychomotor Vigilance Task* (PVT).

Dengan temuan tersebut, dapat disimpulkan bahwa penggunaan citra mata sebagai dasar dalam sistem deteksi kantuk bukan hanya praktis dan non-invasif, tetapi juga memiliki validitas ilmiah yang kuat secara fisiologis. Kombinasi metode EAR, Haar Cascade, dan klasifikasi citra oleh CNN dapat memanfaatkan indikator biologis tersebut secara efektif untuk memberikan peringatan dini kepada pengemudi. Pendekatan ini diharapkan mampu meningkatkan akurasi deteksi kantuk secara real-time dengan kenyamanan yang lebih baik dibandingkan metode konvensional berbasis sensor fisiologis (Ftouni et al., 2013).

Sistem ini juga memanfaatkan *Eye Aspect Ratio* (EAR) sebagai parameter utama untuk mendeteksi penutupan mata dalam durasi tertentu. Ketika nilai EAR turun di bawah ambang batas 0,25 selama beberapa detik, sistem akan memberikan peringatan kepada pengemudi bahwa ia mungkin mengalami kantuk atau *microsleep*, sehingga risiko kecelakaan dapat diminimalisir (Suradi et al., 2023). Penerapan ambang batas EAR dalam sistem ini memastikan respons yang cepat dan tepat waktu untuk mencegah kecelakaan akibat kantuk.

Selain pada sistem pendeteksian kantuk, *Eye Aspect Ratio* (EAR) juga telah diterapkan dalam berbagai bidang lain. Sebagai contoh, dalam penelitian yang menganalisis kedipan mata pada pasien dengan kelumpuhan wajah, EAR digunakan untuk mendeteksi tingkat keterbukaan mata pasien dan dibandingkan dengan subjek yang sehat. Nilai EAR dihitung berdasarkan rasio jarak vertikal dan horizontal mata, di mana nilai EAR akan turun mendekati nol ketika mata tertutup sepenuhnya (Schuhmann et al., 2024). Pada pasien dengan kelumpuhan wajah, nilai EAR minimum lebih tinggi di sisi yang lumpuh dibandingkan dengan kelompok kontrol, menunjukkan adanya kesulitan dalam menutup mata sepenuhnya (Schuhmann et al., 2024). Penelitian ini menunjukkan fleksibilitas penggunaan EAR dalam berbagai aplikasi medis dan non-medis, termasuk pemantauan kesehatan mata pada pasien dengan kelumpuhan wajah.

Eye Aspect Ratio (EAR) juga telah diterapkan dalam sistem deteksi kelelahan pengemudi. Sistem ini memonitor tingkat pembukaan mata secara otomatis dengan menganalisis video pengemudi dan mendeteksi tanda-tanda kantuk, seperti mata yang mulai menutup secara perlahan (Schuhmann et al., 2024). EAR menjadi parameter yang efisien karena dapat diimplementasikan menggunakan kamera standar dan pengolahan citra berbasis video tanpa memerlukan perangkat tambahan yang mahal (Schuhmann et al., 2024). Penerapan EAR dalam sistem deteksi kantuk mengindikasikan bahwa teknologi ini sangat efisien dan praktis, serta dapat diterapkan pada kendaraan apa pun.

Penerapan teknologi pengolahan citra digital dengan metode *Eye Aspect Ratio* (EAR), *Haar Cascade*, dan *Convolutional Neural Networks* (CNN) menawarkan solusi yang efektif dalam mendeteksi kantuk pada pengemudi. Metode EAR

digunakan untuk mengukur rasio antara tinggi dan lebar mata guna menentukan apakah mata dalam keadaan terbuka atau tertutup. *Haar Cascade* berperan dalam mendeteksi wajah dan mata secara *real-time* dari citra video, sedangkan CNN digunakan untuk melakukan klasifikasi kondisi mata berdasarkan pola visual yang telah dipelajari. Kombinasi dari ketiga metode ini memungkinkan terciptanya sistem yang efisien, akurat, dan praktis dalam memberikan peringatan dini kepada pengemudi, serta lebih ekonomis dibandingkan dengan sistem deteksi berbasis sensor fisiologis yang mahal.

Untuk mendukung proses klasifikasi yang optimal, sistem ini dilatih menggunakan *dataset* dari *Kaggle* yang dikembangkan oleh *Dheeraj Perumandla*. *Dataset* ini terdiri dari total 1.452 gambar, yang terbagi secara seimbang antara 726 gambar mata terbuka dan 726 gambar mata tertutup. *Dataset* tersebut telah digunakan secara luas dalam berbagai penelitian karena mencerminkan kondisi nyata dengan variasi pencahayaan dan sudut pandang yang beragam. Pemilihan *dataset* ini bertujuan untuk meningkatkan akurasi dan responsivitas model CNN dalam mendeteksi kondisi mata pengemudi, sehingga sistem dapat bekerja secara andal dalam berbagai situasi visual yang mungkin terjadi saat berkendara.

1.2. Rumusan Masalah

Berdasarkan permasalahan pada latar belakang di atas, penelitian ini akan difokuskan pada beberapa isu utama berikut :

1. Deteksi kantuk pengemudi secara *real-time* merupakan tantangan penting dalam upaya meningkatkan keselamatan berkendara, terutama dalam menyediakan solusi yang akurat dan ekonomis tanpa memerlukan perangkat keras tambahan.

2. Penggunaan metode pengolahan citra digital seperti *EAR*, *Haar Cascade* dan algoritma *machine learning* seperti *CNN* telah banyak diteliti untuk deteksi kondisi mata pengemudi, namun kombinasi kedua metode ini masih memerlukan evaluasi lebih lanjut dalam penerapannya pada sistem deteksi kantuk.
3. Sistem deteksi kantuk yang ada di pasaran masih menghadapi tantangan dalam hal akurasi dan keandalan, terutama dalam mendeteksi kondisi mata pengemudi (terbuka atau tertutup) secara tepat, yang berpengaruh terhadap efektivitas peringatan dini kepada pengemudi.

1.3. Batasan Masalah

Penelitian ini dilakukan dengan batasan-batasan yang ditetapkan secara hati-hati untuk memastikan penelitian tetap fokus dan terarah. Batasan-batasan ini diterapkan agar ruang lingkup penelitian dapat dikelola secara efisien, sehingga menghasilkan keluaran yang sesuai dengan tujuan penelitian. Batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Penelitian ini hanya berfokus pada deteksi kantuk berdasarkan kondisi mata pengemudi (mata terbuka atau tertutup), tanpa mempertimbangkan ekspresi wajah atau gerakan tubuh lainnya.
2. *Dataset* yang digunakan untuk pelatihan model *CNN* hanya mencakup gambar mata terbuka dan tertutup yang diambil dari *Kaggle*.
3. Penelitian ini menggunakan metode *EAR*, *Haar Cascade* dari *OpenCV* untuk deteksi fitur wajah dan mata, serta *CNN* untuk klasifikasi gambar mata.
4. Sistem yang dikembangkan tidak memperhitungkan kondisi pencahayaan ekstrem dan hanya diuji dalam lingkungan pencahayaan normal.

5. Penelitian ini tidak mencakup pengujian dalam kondisi nyata saat mengemudi di jalan raya, namun diuji dalam skenario simulasi dengan video pengemudi.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini diformulasikan untuk memberikan arah yang jelas terhadap pengembangan dan penerapan sistem deteksi kantuk pada pengemudi kendaraan roda empat. Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut :

1. Mengembangkan sistem deteksi kantuk pengemudi menggunakan metode *Haar Cascade* dan EAR pada *OpenCV* untuk mendeteksi kondisi mata pengemudi secara *real-time*.
2. Mengimplementasikan dan melatih model CNN menggunakan *TensorFlow* dan *Keras* untuk mengklasifikasikan kondisi mata pengemudi berdasarkan *dataset* dari *Kaggle*.
3. Mengevaluasi akurasi dan performa sistem deteksi kantuk pengemudi yang dihasilkan dari kombinasi metode EAR, *Haar Cascade* dan CNN.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan berbagai manfaat, baik dalam konteks keselamatan berkendara maupun dalam pengembangan teknologi yang lebih praktis dan ekonomis. Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut :

1. Memberikan kontribusi dalam pengembangan teknologi keselamatan berkendara yang dapat mendeteksi kantuk pengemudi secara *real-time*, sehingga dapat mengurangi risiko kecelakaan akibat kantuk.

2. Menyediakan solusi praktis dan ekonomis untuk deteksi kantuk pengemudi tanpa memerlukan perangkat keras tambahan yang mahal, seperti sensor fisiologis.
3. Menjadi referensi dalam penggunaan teknologi pengolahan citra dan *machine learning* untuk aplikasi keselamatan berkendara, khususnya dalam pendeteksian kantuk.
4. Meningkatkan pemahaman tentang penggunaan kombinasi metode EAR, *Haar Cascade*, dan CNN dalam aplikasi deteksi kondisi mata pengemudi.

BAB II

LANDASAN TEORI

2.1. Mata dan Kantuk

Kantuk merupakan kondisi fisiologis yang memengaruhi sistem saraf pusat dan berdampak langsung pada perilaku ocular seperti kedipan mata dan penutupan kelopak mata. Beberapa indikator visual yang sering diamati selama kondisi kantuk meliputi *blink duration*, *percentage of eye closure* (%TEC), dan *amplitude velocity ratio* (AVR), yang secara signifikan meningkat seiring bertambahnya waktu terjaga. Fenomena ini berkaitan dengan dua mekanisme biologis utama, yaitu proses homeostatik dan sirkadian, yang mengatur kebutuhan tidur berdasarkan waktu dan ritme biologis tubuh. Indikator-indikator tersebut terbukti mampu mencerminkan penurunan tingkat kewaspadaan, bahkan sebelum individu secara sadar merasakannya (Ftouni et al., 2013).

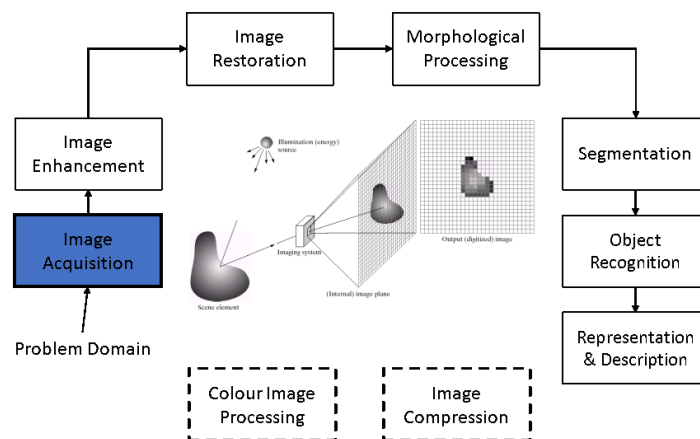
Selain itu, indikator mata seperti %TEC dan *blink duration* juga menunjukkan korelasi yang erat dengan berbagai pengukuran objektif dan subjektif, termasuk *Electroencephalography* (EEG), *Karolinska Sleepiness Scale* (KSS), dan *Psychomotor Vigilance Task* (PVT). Korelasi ini memperkuat validitas penggunaan visual mata sebagai parameter fisiologis dalam sistem deteksi kantuk. Oleh karena itu, pendekatan berbasis pengolahan citra seperti *Eye Aspect Ratio* (EAR) dan *Haar Cascade*, yang memanfaatkan citra mata untuk mendeteksi kondisi mengantuk, menjadi alternatif yang tidak hanya praktis, tetapi juga didukung oleh landasan ilmiah yang kuat (Ftouni et al., 2013).

2.2. Pengolahan Citra Digital

Pengolahan citra digital adalah proses manipulasi gambar menggunakan algoritma komputer dengan tujuan untuk mengekstrak informasi penting dari citra yang diambil. Dalam konteks deteksi kantuk, pengolahan citra digital digunakan untuk mengidentifikasi wajah dan mata pengemudi secara otomatis menggunakan

kamera. Teknik pengolahan citra seperti segmentasi, deteksi tepi, dan ekstraksi fitur dapat membantu sistem mendeteksi perubahan pada wajah pengemudi yang berhubungan dengan kantuk (Suradi et al., 2023).

Pengolahan citra digital melibatkan beberapa tahap, termasuk pra-pemrosesan seperti konversi gambar menjadi skala keabuan (*grayscale*), penajaman gambar, dan normalisasi. Setelah itu, dilakukan proses yang lebih lanjut seperti deteksi objek dan pengenalan pola untuk mengidentifikasi fitur wajah yang relevan (Samsinar et al., 2024). Sistem deteksi kantuk yang menggunakan pengolahan citra digital bertujuan untuk memantau pola mata dan mendeteksi tanda-tanda kantuk berdasarkan perubahan posisi mata atau durasi penutupan mata (Suradi et al., 2023).



Gambar 2.1 sdfghjgf

Gambar 2.1 Konsep Pengolahan Citra Digital

(Sumber : Wikipedia. (2025). Fig 1 Kata kunci pengolahan citra digital.

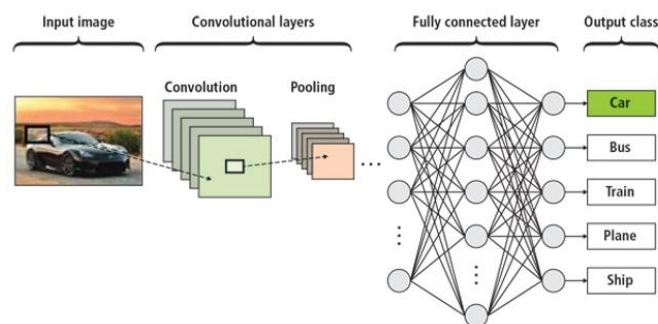
Wikipedia.

https://commons.wikimedia.org/wiki/File:Fig_1_Kata_kunci_pengolahan_citra_digital.png

2.3. Computer Vision

Computer vision adalah bidang ilmu komputer yang bertujuan untuk memberikan kemampuan kepada komputer untuk memahami dan menafsirkan informasi visual dari gambar atau video (Suradi et al., 2023). Pada sistem deteksi kantuk, *computer vision* digunakan untuk mengidentifikasi dan melacak fitur wajah pengemudi secara *real-time*. Algoritma *computer vision* memungkinkan sistem untuk mengenali mata, menghitung rasio aspek mata (EAR), dan memberikan peringatan ketika tanda-tanda kantuk terdeteksi (Samsinar et al., 2024).

Proses ini melibatkan penggunaan algoritma pengenalan pola, di mana sistem mempelajari karakteristik visual yang menunjukkan kantuk, seperti mata yang tertutup lebih lama dari ambang batas yang telah ditentukan (Suradi et al., 2023). Teknik *computer vision* juga mampu menganalisis citra wajah dalam berbagai kondisi pencahayaan dan sudut pandang, sehingga cocok untuk digunakan dalam kendaraan yang bergerak (Samsinar et al., 2024).



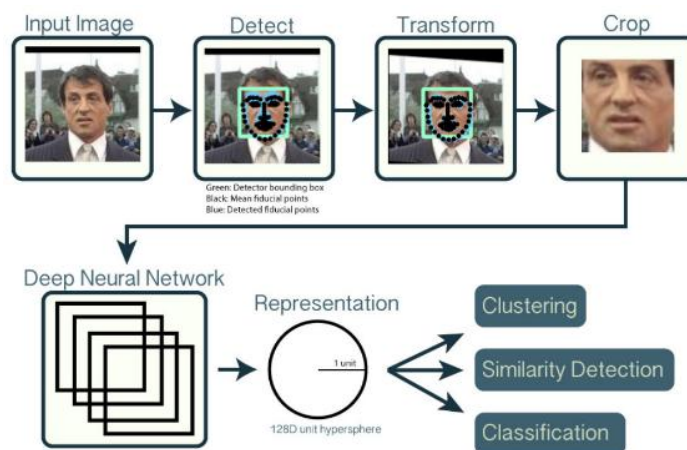
Gambar 2.2 Konsep Computer Vision

(Sumber : NVIDIA. (2024). Ilustrasi konsep Computer Vision. NVIDIA.

<https://www.nvidia.com/content/dam/en-zz/Solutions/glossary/data-science/computer-vision/img-1.png>)

2.4. OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah pustaka *open-source* yang digunakan untuk memproses citra digital dalam aplikasi *real-time*. *OpenCV* mendukung berbagai fungsi visi komputer, seperti deteksi wajah, pelacakan objek, dan pengenalan pola, yang sangat berguna dalam mendeteksi fitur wajah, termasuk mata (Zein, 2018). Pada sistem deteksi kantuk, *OpenCV* berperan dalam menangkap gambar pengemudi secara *real-time* melalui kamera, kemudian memproses citra tersebut untuk mendeteksi fitur wajah, seperti mata dan mulut, yang menjadi indikator kantuk (Thoriq et al., 2024).



Gambar 2.3 Konsep OpenCV

(Sumber : Akademi AI. (2020). Ilustrasi Face Recognition menggunakan

OpenCV. Medium.

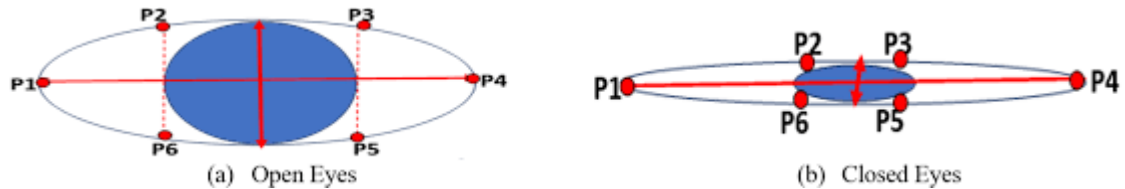
https://miro.medium.com/v2/resize:fit:720/format:webp/1*XWks8Z2KQ4TjCmf7VST7UA.png

2.5. Eye Aspect Ratio (EAR)

Eye Aspect Ratio (EAR) adalah metode yang digunakan untuk mendeteksi kondisi mata berdasarkan rasio jarak vertikal kelopak mata atas dan bawah terhadap jarak horizontal mata (Thoriq et al., 2024). EAR dihitung menggunakan enam titik

referensi pada mata, yaitu P1, P2, P3, P4, P5, dan P6 (Thoriq et al., 2024). Rumus EAR adalah sebagai berikut :

$$EAR = \frac{||P_2 - P_6|| + ||P_3 - P_5||}{2 \times ||P_1 - P_4||}$$



Gambar 2.5 Eye Aspect Ratio untuk Mata Terbuka dan Mata Tertutup

(Sumber : Dewi, C., Chen, R.-C., Jiang, X., & Yu, H. (2022). Eye detection process using facial landmarks. Dalam Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks (hal. 6). PeerJ Computer Science.

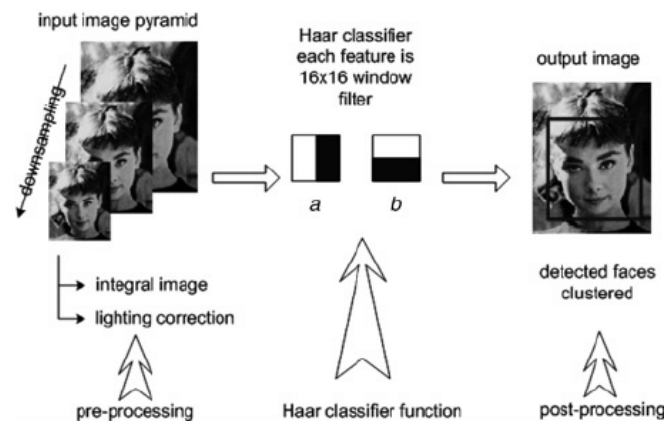
<https://doi.org/10.7717/peerj-cs.943/fig-1>

Jika nilai EAR turun di bawah ambang batas tertentu (biasanya 0.25) dan bertahan dalam beberapa frame, ini mengindikasikan bahwa mata tertutup dan pengemudi mulai mengantuk (Thoriq et al., 2024; Zein, 2018). EAR adalah metode yang efektif dalam mendeteksi kantuk pengemudi secara real-time, karena mengandalkan perubahan kecil pada bentuk mata yang berhubungan dengan kelelahan (Thoriq et al., 2024).

2.6. Haar Cascade

Haar Cascade adalah algoritma deteksi objek yang dikembangkan oleh Paul Viola dan Michael Jones. Algoritma ini memanfaatkan fitur *Haar* untuk mendeteksi pola visual dalam gambar, seperti perbedaan intensitas antara area terang dan gelap (Zein, 2018). Dalam konteks deteksi kantuk, *Haar Cascade* digunakan untuk mendeteksi posisi mata pengemudi secara otomatis (Thoriq et al., 2024).

Haar Cascade bekerja dengan serangkaian tahap klasifikasi untuk mengenali fitur-fitur spesifik wajah, termasuk mata. Proses ini dimulai dengan mendeteksi wajah, kemudian mengekstraksi fitur mata, yang kemudian digunakan dalam perhitungan EAR (Thoriq et al., 2024; Zein, 2018).



Gambar 2.6 Konsep Haar Cascade

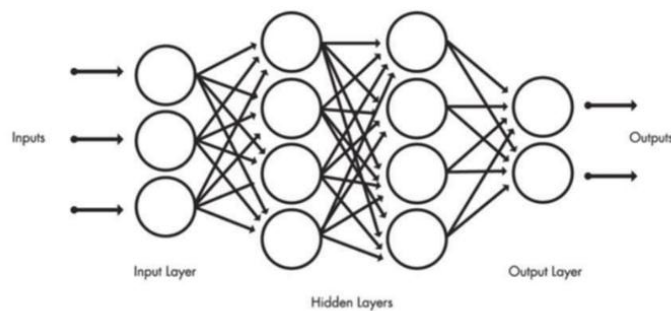
(Sumber : Gao, C., Lu, S.-L., & Lim, H. (2010). Field Programmable Gate Array-Based Haar Classifier for Accelerating Face Detection Algorithm. ResearchGate. https://www.researchgate.net/figure/Face-detection-flow-based-on-the-Haar-classifier_fig1_224141453)

2.7. Deep Learning

Deep Learning merupakan cabang dari pembelajaran mesin yang menggunakan arsitektur jaringan saraf tiruan dengan banyak lapisan untuk mengekstraksi dan memproses fitur dari data yang kompleks, seperti gambar. Salah satu model *Deep Learning* yang sering digunakan adalah *Convolutional Neural Networks* (CNN), yang sangat efektif untuk pengenalan pola, termasuk dalam pengenalan wajah dan objek visual lainnya (Faiq et al., 2024). CNN terdiri dari beberapa lapisan utama seperti lapisan konvolusi yang mengekstraksi fitur-fitur penting dari citra, lapisan pooling untuk mengurangi dimensi data tanpa kehilangan

informasi, serta lapisan *fully connected* untuk melakukan klasifikasi berdasarkan fitur yang telah diekstraksi (Faiq et al., 2024).

Pada penelitian ini, CNN digunakan untuk mengklasifikasikan gambar mata pengemudi sebagai mata terbuka atau tertutup, yang menjadi indikator penting dalam deteksi kantuk. CNN dilatih menggunakan *dataset* yang sesuai dan diuji dengan gambar yang belum pernah dilihat sebelumnya untuk memastikan akurasi klasifikasi. Selain itu, teknik augmentasi data dilakukan untuk meningkatkan jumlah sampel pelatihan guna mencegah *overfitting*, yang dapat berdampak pada generalisasi model (Faiq et al., 2024).



Gambar 2.7 Konsep Deep Learning

(Sumber : Friebe, M. (2017). Concept of Deep Learning. ResearchGate.

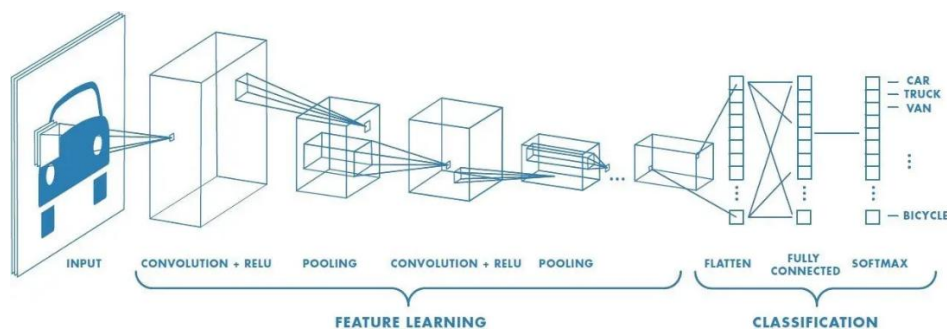
https://www.researchgate.net/figure/Concept-of-Deep-Learning_fig19_319999320)

2.8. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) adalah salah satu algoritma *deep learning* yang dirancang khusus untuk analisis data berbentuk *grid* seperti gambar. CNN terdiri dari beberapa lapisan konvolusi yang berfungsi untuk mengekstrak fitur penting dari gambar, seperti pola, bentuk, dan tepi, yang kemudian digunakan untuk melakukan klasifikasi gambar. CNN banyak digunakan dalam aplikasi

seperti pengenalan wajah, deteksi objek, dan klasifikasi citra medis karena kemampuan model ini dalam mengotomatisasi ekstraksi fitur yang relevan dari data gambar tanpa perlu campur tangan manusia secara langsung (Iswahyudi et al., 2024).

Dalam penelitian ini, CNN digunakan untuk mendeteksi kondisi mata pengemudi, apakah mata pengemudi terbuka atau tertutup, yang merupakan indikator dari kondisi kantuk. Model CNN dilatih menggunakan *dataset* gambar mata terbuka dan tertutup, sehingga sistem dapat mengidentifikasi kondisi kantuk berdasarkan pola yang telah dipelajari (Iswahyudi et al., 2024). Penggunaan CNN memungkinkan pendeteksian yang lebih akurat dan respons yang cepat terhadap perubahan kondisi mata pengemudi, dengan akurasi rata-rata mencapai 97,2% (Suyuti, n.d.).



Gambar 2.8 Konsep Convolutional Neural Network (CNN)

(Sumber : Saha, S. (2018). Ilustrasi arsitektur Convolutional Neural Networks (CNN).

Saturn Cloud. <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>)

2.9. TensorFlow dan Keras

TensorFlow adalah *framework open-source* yang dikembangkan oleh *Google* untuk membangun dan mengimplementasikan model *deep learning*, termasuk *Convolutional Neural Networks (CNN)*. *Framework* ini menyediakan *platform*

yang kuat dan fleksibel untuk komputasi numerik yang kompleks, memungkinkan pengembangan model dengan performa tinggi yang dapat dijalankan pada berbagai perangkat, termasuk *desktop* dan *mobile* (Anwar, 2024). Dalam penelitian ini, *TensorFlow* digunakan sebagai *backend* yang menangani proses komputasi selama pelatihan model CNN untuk mendeteksi kondisi mata pengemudi (Anwar, 2024).

Keras adalah API tingkat tinggi yang berjalan di atas *TensorFlow*, dirancang untuk mempermudah pengembangan model *deep learning*. *Keras* memungkinkan para pengembang membangun model *deep learning* dengan lebih cepat dan efisien, tanpa perlu berurusan dengan detail teknis yang rumit (Anwar, 2024). Dalam penelitian ini, *Keras* digunakan untuk membangun dan melatih model CNN guna mengklasifikasikan gambar mata pengemudi sebagai "mata terbuka" atau "mata tertutup". Model CNN dilatih menggunakan *optimizer* seperti *Adam* dan *loss function binary cross-entropy*, memastikan model mencapai akurasi yang optimal. Selain itu, *TensorFlow Lite* digunakan untuk mengonversi model agar dapat diterapkan pada perangkat *mobile*, memungkinkan deteksi kondisi kantuk secara *real-time* melalui kamera standar yang terpasang di kendaraan (Anwar, 2024).

2.10. Kantuk pada Pengemudi

Kantuk menjadi salah satu penyebab signifikan kecelakaan lalu lintas di seluruh dunia. Pengemudi yang mengantuk sering kehilangan konsentrasi dan dapat mengalami *microsleep*, yaitu kondisi tertidur sejenak yang berlangsung antara 3 hingga 10 detik. Kondisi ini sangat berbahaya saat berkendara karena pengemudi dapat kehilangan kendali dalam hitungan detik (Saputra et al., 2021). Hal ini terutama terjadi pada perjalanan panjang atau shift malam, ketika kelelahan fisik dan kantuk semakin meningkat. Pengemudi yang mengalami *microsleep* tidak sadar

bahwa dirinya tertidur, yang membuat fenomena ini sulit diantisipasi secara mandiri (Saputra et al., 2021).

Sekitar 25% kecelakaan kendaraan disebabkan oleh kantuk, menunjukkan betapa berbahayanya kondisi ini (Asvin Mahersatillah Suradi et al., n.d.). Oleh karena itu, sistem deteksi kantuk yang efisien sangat penting untuk mencegah kecelakaan. Pengembangan sistem yang mampu mendeteksi kantuk secara akurat menggunakan *Eye Aspect Ratio* (EAR) telah terbukti efektif dalam membantu pengemudi tetap waspada dan mencegah kecelakaan akibat kantuk. Sistem ini memungkinkan pemantauan kondisi pengemudi secara *real-time* melalui pengukuran pergerakan kelopak mata (Rusli, 2022).

2.11. Sistem Peringatan Kantuk

Sistem deteksi kantuk yang berbasis EAR bekerja dengan menghitung rasio jarak antara kelopak mata atas dan bawah. Ketika nilai EAR turun di bawah ambang batas tertentu, seperti 0.25, sistem akan memberikan peringatan berupa bunyi alarm atau getaran untuk meningkatkan kewaspadaan pengemudi (Zebua et al., 2024). Teknologi ini umumnya diimplementasikan secara *real-time* menggunakan kombinasi algoritma *Haar Cascade* dan *Convolutional Neural Network* (CNN) yang memungkinkan deteksi kondisi mata terbuka atau tertutup dengan tingkat akurasi yang tinggi (Saputra et al., 2021).

Sistem deteksi kantuk pada pengemudi telah menjadi topik yang penting dalam penelitian teknologi keamanan kendaraan. Sebuah sistem deteksi kantuk berbasis *Convolutional Neural Network* (CNN) dikembangkan untuk menganalisis kondisi mata pengemudi. Sistem ini menggunakan *webcam* untuk mendeteksi pergerakan mata dan menganalisis apakah mata pengemudi terbuka atau tertutup dalam

hitungan detik. Dengan akurasi mencapai 94,51%, sistem ini dapat mendeteksi kondisi pengemudi secara andal pada jarak antara 40 hingga 60 cm dari kamera, memberikan peringatan tepat waktu untuk mencegah kecelakaan akibat kantuk (Rusli, 2022).

2.12. Penelitian Terkait

Penelitian ini membandingkan beberapa penelitian terdahulu untuk mengidentifikasi kelebihan dan kekurangan metode yang telah digunakan guna mengembangkan sistem deteksi kantuk yang lebih efektif.

Tabel 2.1 Penelitian Terkait

No	Nama Tahun dan Judul Penelitian	Metode Penelitian	Hasil Penelitian	Perbedaan Penelitian
1.	Cahya Aji Saputra, 2021, "Deteksi Kantuk Pengendara Roda Empat Menggunakan Haar"	Menggunakan metode Haar Cascade Classifier dan Convolutional Neural Network (CNN) untuk mendeteksi wajah dan mata pengemudi dalam menentukan tingkat kantuk.	Sistem berhasil mendeteksi kantuk dengan akurasi 93,9% berdasarkan perubahan posisi mata.	Penelitian ini mengandalkan Haar Cascade sebagai metode utama, sedangkan penelitian saya mengintegrasikan OpenCV dengan parameter tambahan untuk meningkatkan akurasi deteksi secara real-time.
2.	Ridwan Iswahyudi, 2024, "Deteksi Kantuk Menggunakan Convolutional Neural Network (CNN) Berdasarkan	Menggunakan model Convolutional Neural Network (CNN) untuk mendeteksi tingkat kantuk berdasarkan pola kedipan mata.	CNN mampu mengenali kantuk dengan akurasi 94,4%, lebih baik dibandingkan metode tradisional.	Penelitian ini hanya menggunakan CNN untuk klasifikasi kantuk, sementara penelitian saya menggabungkan Haar Cascade dan teknik tambahan guna meningkatkan kecepatan deteksi dan keandalan sistem.

	Kedipan Mata"			
3.	Auliya Firdaus, 2023, "Sistem Pendeteksi Kantuk Pengemudi berbasis Eye Aspect Ratio dan Mouth"	Menggunakan Eye Aspect Ratio (EAR) dan Mouth Aspect Ratio (MAR) untuk mendeteksi kantuk berdasarkan fitur wajah.	Sistem mendeteksi kantuk dengan akurasi 97%. EAR dan MAR efektif dalam menentukan kantuk pengemudi.	Penelitian ini mengandalkan perhitungan rasio aspek mata dan mulut, sementara penelitian saya menggunakan pendekatan deteksi kedipan mata berbasis OpenCV untuk analisis yang lebih cepat dan efisien.
4.	Aurelia Vitania Rusli, 2022, "Sistem Deteksi Driver Drowsiness Menggunakan Algoritma Convolutional Neural Network (CNN)"	Menggunakan CNN untuk mendeteksi tingkat kantuk berdasarkan pola mata dengan dataset gambar wajah pengemudi.	Sistem mencapai akurasi 94,51% dalam mendeteksi kantuk pada berbagai kondisi pencahayaan.	Penelitian ini hanya mengandalkan CNN, sementara penelitian saya mengimplementasikan OpenCV dengan metode tambahan untuk mendukung deteksi kantuk secara lebih adaptif.
5.	Syefrida Yulina, 2021, "Penerapan Haar Cascade Classifier dalam Mendeteksi Wajah dan Transformasi Citra Grayscale Menggunakan OpenCV"	Menggunakan Haar Cascade Classifier untuk mendeteksi wajah dan mengubah citra menjadi grayscale.	Akurasi deteksi wajah mencapai 100% pada citra frontal, tetapi menurun pada citra dengan sudut kemiringan.	Penelitian ini hanya fokus pada deteksi wajah, sementara penelitian saya mengembangkan sistem lengkap dengan mekanisme peringatan kantuk berbasis analisis kedipan mata.

6.	Mhd. Riski Fauzi, 2024, "Penerapan Algoritma Haar Cascade Eye Detection Untuk Sistem Peringatan Keamanan Pada Pengemudi Mobil"	Menggunakan Haar Cascade untuk mendeteksi mata dan mengaktifkan alarm serta getaran saat kantuk terdeteksi.	Sistem dapat mendeteksi mata dengan baik dalam kondisi pencahayaan optimal, tetapi kurang akurat dalam pencahayaan rendah.	Penelitian ini berfokus pada sistem peringatan berbasis alarm dan getaran, sedangkan penelitian saya mengoptimalkan algoritma deteksi citra agar lebih responsif terhadap perubahan kondisi pencahayaan dan posisi wajah pengemudi.
7.	Sri Nur Marshella, 2024, "Prototype Alat Pendeteksi Kantuk Menggunakan Sensor MAX30102 dan Kamera dengan Metode Eye Aspect Ratio"	Menggunakan sensor MAX30102 untuk mendeteksi detak jantung dan metode Eye Aspect Ratio (EAR) untuk mengukur kedipan mata pengemudi..	Sistem dapat bekerja dalam kondisi pencahayaan berbeda dengan akurasi yang baik, serta mendeteksi kantuk berdasarkan kombinasi nilai detak jantung dan EAR.	Penelitian ini menggabungkan deteksi detak jantung dan kedipan mata, sedangkan penelitian saya lebih berfokus pada analisis citra berbasis OpenCV tanpa sensor tambahan.
8.	Satria Aditama Yadavia, 2023, "Rancang Bangun Alat Pendeteksi Kantuk pada Kendaraan Roda Empat dengan Metode Haar Cascade"	Menggunakan metode Haar Cascade untuk mendeteksi mata dengan Raspberry Pi sebagai prosesor utama.HSV.	Sistem dapat mendeteksi kantuk berdasarkan perubahan kedipan mata dengan tingkat keakuratan tinggi.	Penelitian ini menggunakan Raspberry Pi untuk pemrosesan, sedangkan penelitian saya menggunakan OpenCV dengan optimalisasi pada parameter deteksi.

9.	Maghfirah Suyuti, 2023, "Pengembangan Model Klasifikasi Mata Tertutup dan Terbuka Dalam Identifikasi Kelelahan Menggunakan Arsitektur Mobile CNN"	Menggunakan MobileNet CNN untuk mengklasifikasikan kondisi mata terbuka atau tertutup.	Model CNN yang digunakan mencapai akurasi 98% dengan inference time 0,05 detik.	Penelitian ini berfokus pada model CNN yang ringan untuk klasifikasi mata, sedangkan penelitian saya mengoptimalkan metode deteksi kantuk menggunakan OpenCV dan algoritma tambahan.
----	---	--	---	--

Tabel ini memberikan gambaran mengenai penelitian terdahulu dan perbedaannya dengan penelitian yang sedang dilakukan, sehingga dapat menunjukkan kontribusi penelitian ini dalam pengembangan sistem deteksi kantuk pengemudi.

Dari penelitian-penelitian sebelumnya, dapat disimpulkan bahwa berbagai pendekatan telah digunakan untuk mendeteksi kantuk, baik dengan metode berbasis citra seperti Haar Cascade dan CNN, maupun dengan sensor tambahan seperti MAX30102. Penelitian saya berbeda dengan mengintegrasikan OpenCV dengan optimalisasi parameter deteksi, sehingga sistem dapat bekerja secara real-time dengan tingkat akurasi yang lebih baik tanpa memerlukan sensor tambahan. Selain itu, penelitian ini berfokus pada peningkatan keandalan deteksi dalam berbagai kondisi pencahayaan dan sudut pandang, yang menjadi tantangan utama dalam implementasi sistem pendeteksi kantuk.

BAB III

METODOLOGI PENELITIAN

3.1. Metode Penelitian

Metode penelitian merupakan pendekatan yang digunakan untuk memperoleh data dan informasi secara sistematis guna menjawab permasalahan penelitian serta mencapai tujuan yang telah ditetapkan. Dalam penelitian ini, metode yang digunakan adalah observasi, yaitu metode pengumpulan data dengan cara mengamati langsung fenomena yang menjadi objek penelitian. Observasi dilakukan untuk memahami dan menganalisis pola keterbukaan dan penutupan mata yang menjadi indikator kantuk pengemudi. Dengan metode ini, penelitian dapat memperoleh data yang bersifat empiris dan berdasarkan kondisi nyata yang diamati, sehingga memungkinkan pengembangan sistem deteksi kantuk yang lebih akurat dan aplikatif.

Observasi dalam penelitian ini difokuskan pada pengumpulan dan analisis data citra digital mata dalam kondisi terbuka dan tertutup. Data yang digunakan berasal dari *dataset* yang diperoleh melalui platform *Kaggle*, yang berisi kumpulan gambar mata dengan berbagai variasi kondisi pencahayaan, sudut pandang, serta karakteristik individu yang berbeda. Penggunaan metode observasi dalam penelitian ini bertujuan untuk mengidentifikasi pola visual dari mata yang mengalami perubahan kondisi, sehingga dapat digunakan dalam implementasi sistem deteksi kantuk berbasis pengolahan citra digital. Dengan pendekatan ini, penelitian dapat memastikan bahwa sistem yang dikembangkan mampu bekerja secara optimal tanpa memerlukan perangkat keras tambahan seperti sensor

fisiologis, sehingga lebih efisien dan mudah diterapkan dalam berbagai jenis kendaraan.

3.1.1 Karakteristik Metode Observasi

Metode observasi dalam penelitian ini memungkinkan pengumpulan data secara langsung tanpa intervensi, memastikan hasil yang empiris dan dapat diverifikasi. Prosesnya dilakukan secara sistematis dan terstruktur, dari pengumpulan data hingga analisis, guna menjaga validitas penelitian. Berikut karakteristik utamanya :

1. Langsung dan Objektif

Observasi dalam penelitian ini dilakukan secara langsung terhadap dataset gambar mata yang telah dikumpulkan. Proses pengamatan dan analisis data dilakukan secara objektif berdasarkan parameter yang telah ditentukan.

2. Empiris dan Berdasarkan Data Nyata

Data yang digunakan dalam penelitian ini berasal dari kumpulan gambar mata terbuka dan tertutup yang tersedia dalam *dataset*. Data ini dianalisis menggunakan metode pengolahan citra digital untuk memperoleh pola yang dapat digunakan dalam sistem deteksi kantuk.

3. Sistematis dan Terstruktur

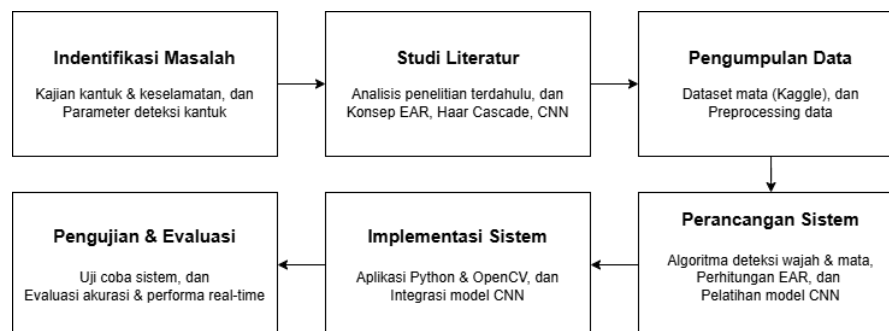
Proses observasi dilakukan secara sistematis dengan tahapan yang jelas, mulai dari pengumpulan data, analisis gambar, hingga penerapan metode pengolahan citra. Dengan demikian, hasil penelitian dapat dipertanggungjawabkan secara ilmiah.

Metode observasi dalam penelitian ini berperan penting dalam memastikan bahwa sistem deteksi kantuk yang dikembangkan dapat bekerja secara optimal

berdasarkan pola visual yang dianalisis. Dengan menggunakan pendekatan ini, penelitian dapat menghasilkan temuan yang lebih akurat, terutama dalam memahami karakteristik mata pengemudi saat mengalami kantuk. Selain itu, metode ini juga memberikan fleksibilitas dalam eksplorasi data, sehingga memungkinkan pengembangan model yang lebih adaptif terhadap berbagai kondisi pencahayaan, sudut pandang kamera, dan variasi individu. Dengan demikian, hasil penelitian ini diharapkan tidak hanya memberikan kontribusi akademik, tetapi juga memiliki dampak nyata dalam meningkatkan keselamatan berkendara melalui sistem deteksi kantuk yang lebih efisien dan aplikatif.

3.2. Alur Penelitian

Alur penelitian ini mencakup beberapa tahapan penting yang disusun secara sistematis untuk memastikan setiap proses berjalan dengan baik dan menghasilkan data yang valid. Tahapan-tahapan ini akan dijelaskan dalam bentuk diagram alur untuk memberikan gambaran yang lebih jelas mengenai proses penelitian. Berikut adalah tahapan dalam penelitian ini :



Gambar 3.1 Diagram Alur Penelitian

3.2.1 Identifikasi Masalah

Identifikasi masalah merupakan tahap awal yang krusial dalam penelitian, di mana perumusan masalah harus memenuhi prinsip keilmuan, seperti rasionalitas,

bukti empiris, dan keteraturan dalam prosedur. Penelitian yang dilakukan harus dapat diterima oleh akal sehat, sehingga setiap langkah yang diambil memiliki dasar logis dan dapat dipahami melalui penalaran yang jelas.

Dalam konteks penelitian ini, masalah yang dikaji adalah kantuk pada pengemudi dan dampaknya terhadap keselamatan berkendara. Kantuk dapat menyebabkan *microsleep*, kondisi tidur singkat yang tidak disadari, yang berpotensi menyebabkan kecelakaan fatal. Untuk memastikan penelitian ini memiliki dasar empiris, digunakan data nyata dalam bentuk citra mata terbuka dan tertutup yang dapat dianalisis secara sistematis. Sistem deteksi kantuk ini memanfaatkan metode *Eye Aspect Ratio* (EAR) yang mengukur perubahan bentuk mata sebagai indikator kantuk, *Haar Cascade* untuk mendeteksi wajah dan mata secara *real-time*, serta *Convolutional Neural Networks* (CNN) untuk meningkatkan akurasi klasifikasi kondisi mata pengemudi. Dengan pendekatan yang sistematis dan berbasis bukti, penelitian ini diharapkan dapat menghasilkan solusi yang efektif dan aplikatif dalam meningkatkan keselamatan berkendara.

3.2.2 Studi Literatur

Pada tahapan awal penelitian, saya mengumpulkan, meninjau, dan menganalisis berbagai literatur yang relevan dengan topik deteksi kantuk pada pengemudi. Studi literatur ini mencakup pengumpulan informasi dari berbagai sumber seperti jurnal ilmiah, buku, dan penelitian terdahulu yang membahas sistem deteksi kantuk berbasis pengolahan citra digital. Selain itu, proses membaca dan mencatat informasi dilakukan secara sistematis untuk mengidentifikasi metode yang telah digunakan dalam penelitian sebelumnya serta memahami perkembangan teknologi dalam bidang ini.

Sebagai bagian dari proses penelitian, studi literatur digunakan untuk memahami konsep-konsep yang telah dikembangkan terkait deteksi kantuk, seperti penggunaan parameter *Eye Aspect Ratio* (EAR) dalam mengidentifikasi perubahan bentuk mata yang menjadi indikator kantuk. Selain itu, teknologi seperti *Haar Cascade* dan *Convolutional Neural Networks* (CNN) telah banyak diterapkan dalam pengolahan citra digital untuk mendeteksi kondisi kantuk secara *real-time*. Dengan menelaah penelitian terdahulu, saya dapat mengidentifikasi metode yang paling efektif serta menentukan pendekatan yang sesuai untuk penelitian ini.

Disamping itu, studi literatur juga berperan dalam mengidentifikasi celah pengetahuan yang masih dapat dikembangkan dalam penelitian ini. Beberapa penelitian sebelumnya telah mengusulkan berbagai metode deteksi kantuk, namun masih terdapat tantangan dalam hal akurasi dan kecepatan pemrosesan sistem. Oleh karena itu, penelitian ini berupaya untuk menyempurnakan metode deteksi kantuk dengan pendekatan yang lebih optimal dan efisien, sehingga dapat memberikan kontribusi dalam meningkatkan keselamatan berkendara.

3.2.3 Pengumpulan Data

Dalam penelitian ini, data yang digunakan diperoleh dari dataset *Drowsiness_dataset* yang dikembangkan oleh Dheeraj Perumandla dan diperbarui empat tahun yang lalu di platform *Kaggle*. *Dataset* ini terdiri dari empat folder, yaitu *Closed*, *Open*, *Yawn*, dan *No Yawn*. Namun, dalam penelitian ini, hanya digunakan data dari folder *Closed* dan *Open*, yang masing-masing berisi 726 gambar mata dalam kondisi tertutup dan 726 gambar mata dalam kondisi terbuka.

Pengumpulan data dilakukan dengan mengunduh dataset secara langsung dari *Kaggle* dan melakukan proses validasi terhadap kualitas gambar untuk memastikan

kejelasan serta konsistensi data. Setiap gambar dalam dataset dianalisis untuk mengidentifikasi parameter utama yang diperlukan dalam penelitian ini, seperti *Eye Aspect Ratio* (EAR), yang digunakan untuk membedakan kondisi mata terbuka dan tertutup. Selain itu, dataset ini diproses menggunakan metode augmentasi data guna meningkatkan keberagaman sampel dan memperkuat kemampuan model dalam mendeteksi kantuk pada berbagai kondisi pencahayaan dan sudut pandang.

Penggunaan dataset *Drowsiness_dataset* dipilih karena memenuhi kriteria yang diperlukan dalam penelitian ini, yaitu ketersediaan gambar dengan resolusi yang cukup untuk analisis pengolahan citra serta keseimbangan antara jumlah gambar mata terbuka dan tertutup. Dengan data yang telah dikumpulkan, proses pelatihan model dilakukan menggunakan metode *Haar Cascade* untuk pendeteksian mata serta *Convolutional Neural Networks* (CNN) untuk klasifikasi kondisi mata. Pengolahan citra dilakukan secara *real-time* menggunakan pustaka *OpenCV*, sehingga sistem dapat mendeteksi kantuk pengemudi dengan lebih akurat dan efisien.

Untuk memberikan gambaran lebih jelas mengenai *dataset* yang digunakan, berikut adalah contoh sampel gambar dari data mata terbuka dan tertutup yang diambil dari *dataset* :



Gambar 3.2 Mata Terbuka



Gambar 3.3 Mata Tertutup

3.2.4 Perancangan Sistem

Perancangan sistem deteksi kantuk pada pengemudi ini menerapkan metode pengolahan citra digital dengan tujuan untuk menganalisis kondisi mata secara *real-time*. Sistem ini memanfaatkan kamera untuk menangkap citra wajah pengemudi, yang selanjutnya diproses menggunakan metode *Haar Cascade* guna mendeteksi area mata. Setelah itu, nilai *Eye Aspect Ratio* (EAR) dihitung untuk menentukan apakah mata pengemudi dalam keadaan terbuka atau tertutup. Apabila nilai EAR berada di bawah ambang batas tertentu dalam durasi waktu yang telah ditentukan, sistem akan memberikan peringatan dalam bentuk suara atau tampilan visual guna meningkatkan kewaspadaan pengemudi.

Selain menggunakan pendekatan *rule-based*, sistem ini akan diintegrasikan dengan metode *machine learning*, khususnya menggunakan model *Convolutional Neural Network* (CNN), untuk meningkatkan akurasi deteksi kantuk. CNN digunakan untuk menganalisis karakteristik wajah pengemudi secara lebih mendalam, sehingga mampu mengenali pola kantuk berdasarkan dataset pelatihan yang telah dikembangkan. Dengan kombinasi kedua pendekatan ini, sistem diharapkan dapat memberikan peringatan secara *real-time* ketika pengemudi menunjukkan tanda-tanda kantuk, sehingga meningkatkan keselamatan berkendara.

3.2.5 Implementasi Sistem

Sistem deteksi kantuk ini diimplementasikan dalam bentuk aplikasi desktop berbasis *Python*. Aplikasi ini dirancang untuk mendeteksi kondisi kantuk pengemudi secara *real-time* dengan memanfaatkan kamera (*webcam*) dan dijalankan secara lokal pada perangkat laptop atau komputer.

Proses deteksi dimulai dengan menangkap video wajah pengemudi menggunakan *OpenCV*, kemudian dilakukan deteksi wajah dan mata menggunakan metode *Haar Cascade*. Setelah area mata teridentifikasi, sistem menghitung nilai *Eye Aspect Ratio* (EAR) untuk menentukan apakah mata terbuka atau tertutup. Sebagai validasi tambahan, sistem juga melakukan klasifikasi menggunakan model *Convolutional Neural Network* (CNN) yang telah dilatih sebelumnya dengan dataset gambar mata.

Jika kedua metode mendeteksi kondisi mata tertutup selama lebih dari ambang waktu tertentu (misalnya 2 detik), aplikasi akan memberikan peringatan dalam bentuk alarm suara dan notifikasi visual. Antarmuka aplikasi menggunakan *Tkinter* untuk menampilkan tampilan video, status deteksi, serta peringatan kepada pengguna.

Pemilihan aplikasi desktop dipilih karena lebih stabil untuk proses pengolahan citra dan integrasi model deep learning. Implementasi berbasis website dinilai kurang ideal karena keterbatasan akses kamera di *browser*, *latency* tinggi, dan kompleksitas integrasi model CNN ke format web. Sementara implementasi *mobile* memerlukan konversi model ke *TensorFlow Lite* serta pengembangan aplikasi *Android* secara terpisah, yang membutuhkan sumber daya lebih besar.

Dengan demikian, bentuk desktop memberikan solusi yang praktis, efisien, dan responsif, serta sangat sesuai untuk keperluan penelitian ini yang membutuhkan deteksi *real-time* dan akurasi tinggi.

3.2.6 Pengujian dan Evaluasi

Pengujian dan evaluasi dilakukan terhadap sistem deteksi kantuk yang telah diimplementasikan dalam aplikasi untuk memastikan kinerjanya dalam kondisi nyata. Pengujian mencakup integrasi model CNN dengan aplikasi, di mana sistem diuji dalam berbagai skenario berkendara untuk menilai akurasi deteksi mata terbuka dan tertutup secara *real-time*. Selain itu, metode *Eye Aspect Ratio* (EAR) dan *Haar Cascade* dievaluasi dalam aplikasi untuk mengukur kecepatan deteksi serta kestabilan dalam mengenali fitur wajah dan mata. Hasil deteksi dibandingkan dengan kondisi aktual pengemudi untuk memastikan bahwa sistem dapat memberikan peringatan dini secara akurat ketika ambang batas EAR tercapai.

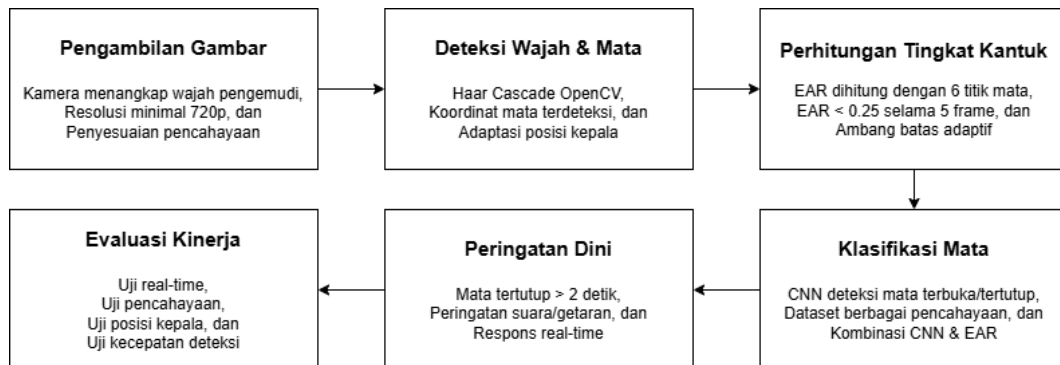
Evaluasi dilakukan dengan menganalisis performa aplikasi berdasarkan akurasi deteksi, responsivitas peringatan, serta pengaruh faktor eksternal seperti pencahayaan, sudut wajah, dan kualitas kamera perangkat. Pengujian juga mencakup uji coba pengguna untuk menilai pengalaman dan efektivitas aplikasi dalam mendukung keselamatan berkendara. Hasil evaluasi digunakan untuk mengidentifikasi kekurangan sistem dan memberikan rekomendasi perbaikan guna meningkatkan akurasi, efisiensi, serta kenyamanan pengguna dalam mengoperasikan aplikasi deteksi kantuk ini.

3.3. Perancangan Sistem

Sistem deteksi kantuk ini dirancang untuk mengidentifikasi tanda-tanda kantuk pengemudi secara *real-time*. Proses dimulai dari pengambilan citra wajah,

dilanjutkan dengan deteksi area mata, analisis kondisi mata, hingga pemberian peringatan jika kantuk terdeteksi. Setiap tahapan dirancang agar bekerja secara efisien dan akurat.

Diagram berikut menggambarkan alur sistem deteksi kantuk, mulai dari akuisisi gambar hingga aktivasi peringatan :



Gambar 3.4 Diagram Perancangan Sistem

3.3.1 Pengambilan gambar

Pengambilan gambar merupakan tahap awal dalam sistem deteksi kantuk yang dirancang dalam penelitian ini. Proses ini bertujuan untuk memperoleh citra wajah pengemudi secara *real-time* sebagai data utama dalam analisis kondisi mata. Kamera yang digunakan harus memiliki resolusi minimal 720p untuk memastikan kejernihan citra yang diperoleh, sehingga fitur wajah, terutama area mata, dapat terdeteksi dengan akurat.

Dalam proses ini, beberapa faktor penting harus diperhatikan, seperti sudut pengambilan gambar, pencahayaan, dan kestabilan citra. Kamera perlu diposisikan sedemikian rupa agar wajah pengemudi terekam dengan jelas tanpa terhalang objek lain. Pencahayaan yang memadai juga diperlukan agar fitur wajah tetap terlihat dengan baik dalam berbagai kondisi lingkungan. Gambar yang telah diperoleh akan

melalui tahap *preprocessing* guna meningkatkan kualitas citra sebelum dianalisis lebih lanjut dengan metode deteksi yang telah dirancang.

3.3.2 Deteksi Wajah dan Mata

Pada tahap ini, sistem dirancang untuk melakukan identifikasi wajah dan mata pengemudi menggunakan metode *Haar Cascade* dari *OpenCV*. Algoritma *Haar Cascade* dipilih karena sifatnya yang efisien dalam mendeteksi wajah secara *real-time*. Dalam perancangan ini, proses deteksi wajah bertujuan sebagai langkah awal sebelum menentukan koordinat area mata yang akan dianalisis lebih lanjut.

Setelah wajah terdeteksi, sistem akan diarahkan untuk mengekstrak fitur mata dengan menentukan koordinatnya secara relatif terhadap posisi wajah. Perancangan ini mempertimbangkan berbagai kemungkinan pergeseran posisi kepala, seperti sedikit miring atau menunduk, agar sistem tetap dapat mengidentifikasi area mata dengan baik. Selain itu, untuk mengatasi potensi gangguan dari pencahayaan atau noise, dipertimbangkan pula penggunaan teknik pra-pemrosesan citra, seperti peningkatan kontras atau penghalusan tepi, guna memperjelas area mata yang akan dianalisis.

Hasil dari tahap ini akan menjadi dasar dalam perhitungan *Eye Aspect Ratio* (EAR) pada tahap berikutnya. Oleh karena itu, pemilihan metode dan parameter deteksi perlu disesuaikan agar sistem dapat mengenali mata pengemudi secara optimal dalam berbagai kondisi.

3.3.3 Perhitungan Tingkat Kantuk

Setelah wajah dan mata berhasil dideteksi, langkah selanjutnya adalah menghitung tingkat kantuk berdasarkan perubahan bentuk mata. Untuk mengukur kondisi mata terbuka atau tertutup, digunakan metode *Eye Aspect Ratio* (EAR),

yang dihitung berdasarkan enam titik kunci pada mata. Formula EAR didefinisikan sebagai berikut :

$$EAR = \frac{||P_2 - P_6|| + ||P_3 - P_5||}{2 \times ||P_1 - P_4||}$$

Di mana P1, P2, P3, P4, P5, P6 merupakan koordinat titik fitur mata yang diperoleh dari hasil deteksi wajah sebelumnya. Nilai EAR akan mengalami penurunan ketika mata mulai tertutup dan kembali meningkat ketika mata terbuka. Oleh karena itu, nilai EAR dapat digunakan sebagai indikator untuk mendeteksi kondisi kantuk.

Agar sistem dapat mengidentifikasi kantuk dengan lebih akurat, diperlukan ambang batas tertentu. Berdasarkan studi sebelumnya, nilai EAR di bawah 0.25 sering digunakan sebagai indikasi bahwa mata dalam kondisi tertutup. Namun, untuk mengurangi kemungkinan kesalahan deteksi akibat kedipan mata biasa, sistem hanya akan menganggap pengemudi dalam kondisi mengantuk jika nilai EAR berada di bawah ambang batas selama 5 frame berturut-turut.

Selain itu, untuk meningkatkan akurasi deteksi, perlu dilakukan penyesuaian terhadap faktor-faktor eksternal seperti kondisi pencahayaan dan sudut wajah. Jika sistem tidak mampu menangkap fitur mata dengan jelas, dapat terjadi kesalahan dalam perhitungan EAR yang berpotensi mengurangi keandalan deteksi kantuk. Oleh karena itu, optimasi parameter dan pengujian dalam berbagai skenario menjadi aspek penting dalam tahap ini.

3.3.4 Klasifikasi Mata dan Pembangunan Model CNN

Dalam penelitian ini, klasifikasi mata dilakukan untuk menentukan apakah mata pengemudi dalam keadaan terbuka atau tertutup, sebagai parameter utama dalam deteksi kantuk. Proses klasifikasi ini dilakukan menggunakan kombinasi

metode *Eye Aspect Ratio* (EAR) dan model *Convolutional Neural Network* (CNN) untuk meningkatkan akurasi deteksi serta mengatasi berbagai kondisi pencahayaan dan sudut wajah.

Penelitian ini menggunakan *Drowsiness_dataset*, yang dikembangkan oleh Dheeraj Perumandla dan tersedia pada platform Kaggle. *Dataset* ini terdiri dari empat kategori utama, yaitu *Closed* (mata tertutup), *Open* (mata terbuka), *Yawn* (menguap), dan *No Yawn* (tidak menguap). Namun, dalam penelitian ini hanya digunakan dua kategori, yaitu *Closed* dan *Open*, dengan jumlah masing-masing 726 gambar, sehingga total dataset yang digunakan adalah 1.452 gambar.

Sebelum digunakan dalam proses pelatihan, dataset melalui tahap *preprocessing* untuk meningkatkan kualitas data serta mengoptimalkan kinerja model. Proses ini mencakup :

1. Konversi ke Grayscale

Mengurangi kompleksitas fitur dengan hanya mempertahankan informasi bentuk dan tekstur.

2. Normalisasi Intensitas Piksel

Nilai piksel dinormalisasi ke rentang 0 hingga 1 guna mempercepat konvergensi model selama proses pelatihan.

3. Augmentasi Data

Teknik augmentasi seperti rotasi, *flipping horizontal*, dan perubahan pencahayaan diterapkan untuk meningkatkan generalisasi model dan mengurangi risiko *overfitting*.

4. Resizing Citra

Semua citra disesuaikan ke ukuran 64×64 piksel untuk menyeragamkan dimensi input ke model.

Selain CNN, metode *Eye Aspect Ratio* (EAR) digunakan untuk mendeteksi perubahan kondisi mata berdasarkan pergerakan kelopak mata. EAR dihitung menggunakan enam titik referensi pada mata dengan rumus sebagai berikut :

$$EAR = \frac{||P_2 - P_6|| + ||P_3 - P_5||}{2 \times ||P_1 - P_4||}$$

Di mana :

- a. P1, P2, P3, P4, P5, dan P6 merupakan koordinat landmark mata berdasarkan deteksi wajah menggunakan *Haar Cascade* dari *OpenCV*.
- b. EAR akan bernilai tinggi jika mata terbuka dan akan menurun jika mata tertutup.
- c. Jika $EAR < 0.25$ selama 5 frame berturut-turut, mata dianggap tertutup.

Dengan mengombinasikan metode EAR dengan CNN, sistem dapat meningkatkan akurasi klasifikasi serta mendeteksi kantuk dengan lebih andal.

Model *Convolutional Neural Network* (CNN) yang digunakan dalam penelitian ini terdiri dari beberapa lapisan untuk mengekstraksi fitur visual dari citra mata. Lapisan pertama adalah *Convolutional Layer* dengan 32 *filter*, ukuran kernel 3×3, fungsi aktivasi *ReLU*, dan *padding same* untuk mempertahankan dimensi input. Setelah itu, diterapkan *MaxPooling Layer* berukuran 2×2 untuk mengurangi dimensi fitur. Model dilanjutkan dengan lapisan konvolusi kedua yang memiliki 64 *filter*, ukuran kernel 3×3, fungsi aktivasi *ReLU*, dan *padding same*. Untuk lebih mereduksi ukuran fitur, digunakan kembali *MaxPooling Layer* berukuran 2×2. Selain itu, *Dropout Layer* dengan rasio 0.25 diterapkan untuk mengurangi risiko *overfitting*.

Setelah proses ekstraksi fitur, model mengubah output konvolusi menjadi satu dimensi melalui *Flatten Layer*. Kemudian, fitur ini diproses pada *Dense Layer* dengan 128 *neuron* dan fungsi aktivasi *ReLU*. Untuk meningkatkan generalisasi model, diterapkan kembali *Dropout Layer* dengan *rasio* 0.5 sebelum masuk ke *Output Layer*. Lapisan ini terdiri dari 2 *neuron*, masing-masing untuk kelas mata terbuka dan tertutup, dengan fungsi aktivasi *Softmax*. Model dilatih menggunakan *optimizer Adam* dengan *learning rate* 0.001, fungsi *loss Categorical Crossentropy*, *batch size* 32, dan dijalankan selama 25 *epoch*. Pembagian *dataset* dilakukan dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian guna memastikan performa model yang optimal.

3.3.5 Peringatan Dini

Sistem peringatan dini dalam penelitian ini dikembangkan untuk memberikan notifikasi secara *real-time* kepada pengemudi saat kondisi kantuk terdeteksi. Implementasi dilakukan dalam bentuk aplikasi *desktop* berbasis *Python*, yang akan memberikan peringatan jika mata pengemudi terdeteksi tertutup selama 2 detik berturut-turut. Deteksi ini dilakukan dengan mengombinasikan hasil klasifikasi model CNN dan perhitungan *Eye Aspect Ratio* (EAR) untuk meningkatkan akurasi dalam berbagai kondisi pencahayaan dan sudut wajah.

Setelah kondisi kantuk terdeteksi, aplikasi akan mengaktifkan mekanisme peringatan berupa notifikasi visual dan alarm suara. Notifikasi visual ditampilkan melalui antarmuka aplikasi, yang menampilkan pesan peringatan kepada pengemudi. Selain itu, aplikasi akan memutar suara peringatan (alarm) dengan intensitas yang cukup untuk menarik perhatian. Dengan sistem ini, peringatan dapat

disampaikan secara efektif kepada pengemudi, sehingga membantu mengurangi risiko kecelakaan akibat kantuk saat berkendara.

3.3.6 Evaluasi dan Pengujian Model serta Kinerja Sistem

Evaluasi kinerja model dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* guna memastikan performa sistem dalam mendeteksi kondisi kantuk dengan akurasi yang tinggi. Selain itu evaluasi kinerja sistem dilakukan untuk menguji efektivitas dan keandalan model dalam mendeteksi kantuk berdasarkan indikator utama, yaitu kondisi mata. Pengujian ini mencakup beberapa aspek penting, yaitu pengujian *real-time*, pengaruh pencahayaan, variasi posisi kepala, serta analisis kecepatan deteksi dan waktu respons sistem

1. Evaluasi dan Pengujian Model

Evaluasi dan pengujian model dilakukan untuk menilai performa algoritma dalam mendeteksi kondisi mata (terbuka atau tertutup) sebagai indikator utama kantuk. Pengujian ini dilakukan menggunakan metrik evaluasi yang umum digunakan dalam klasifikasi biner, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-score*. Metrik ini dihitung berdasarkan perbandingan antara prediksi model dan data sebenarnya.

Untuk menghitung performa model, digunakan konsep *Confusion Matrix*, yang terdiri dari :

- a. True Positive (TP): Mata tertutup terdeteksi dengan benar.
- b. True Negative (TN): Mata terbuka terdeteksi dengan benar.
- c. False Positive (FP): Mata terbuka tetapi salah terdeteksi sebagai tertutup.
- d. False Negative (FN): Mata tertutup tetapi salah terdeteksi sebagai terbuka.

Metrik evaluasi dihitung menggunakan rumus berikut:

a. Accuracy :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Mengukur persentase prediksi yang benar terhadap seluruh data.

b. Precision :

$$Precision = \frac{TP}{TP + FP}$$

Mengukur seberapa akurat model dalam mendeteksi mata tertutup tanpa menghasilkan banyak kesalahan positif.

c. Recall :

$$Recall = \frac{TP}{TP + FN}$$

Mengukur seberapa baik model mendeteksi semua kondisi mata tertutup yang sebenarnya.

d. F1-Score :

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Menggabungkan precision dan recall untuk memberikan evaluasi keseimbangan antara keduanya.

Pengujian model dilakukan menggunakan dataset yang berisi gambar mata dalam berbagai kondisi pencahayaan dan sudut kepala. Model diuji dengan data uji yang belum pernah digunakan dalam proses pelatihan untuk mendapatkan metrik evaluasi yang objektif. Model yang memiliki *accuracy* tinggi dengan *F1-score* yang seimbang menunjukkan bahwa sistem mampu mengklasifikasikan kondisi mata dengan baik dan siap untuk diimplementasikan dalam sistem deteksi kantuk secara *real-time*.

2. Evaluasi Kinerja Sistem

Evaluasi kinerja sistem bertujuan untuk mengukur efektivitas dan *keandalan* sistem dalam mendeteksi kantuk berdasarkan kondisi mata. Pengujian ini mencakup uji *real-time*, pengaruh pencahayaan, variasi posisi kepala, serta analisis kecepatan deteksi dan waktu respons sistem guna memastikan bahwa sistem bekerja optimal dalam berbagai kondisi.

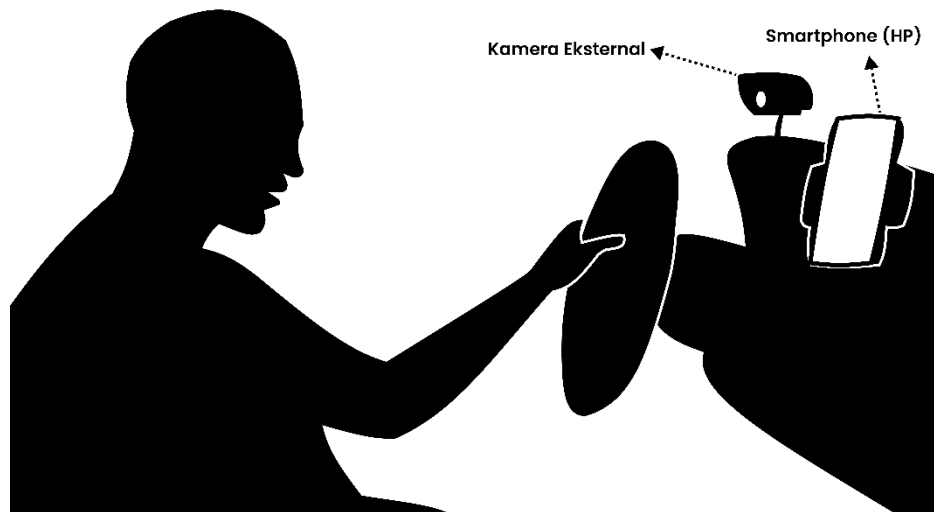
Pada uji *real-time*, sistem dijalankan dengan menangkap video dari *webcam* untuk mengamati klasifikasi kondisi mata secara langsung. Uji pencahayaan dilakukan dalam kondisi terang dan redup untuk memastikan model tetap akurat. Pengujian posisi kepala mencakup variasi seperti menunduk atau miring guna mengevaluasi ketepatan deteksi mata. Terakhir, analisis kecepatan deteksi dan waktu respons dilakukan dengan mengukur durasi dari pengambilan citra hingga keluarnya peringatan, memastikan sistem dapat memberikan respons yang cepat dan akurat.

3.4. Desain Implementasi Sistem pada Kendaraan Roda Empat

Untuk mendukung kemudahan penggunaan dan pemantauan oleh pengemudi, sistem deteksi kantuk ini dirancang dengan arsitektur sebagai berikut :

1. Proses utama deteksi dilakukan oleh laptop atau mini-PC yang terhubung ke kamera.
2. *Smartphone* (HP) digunakan sebagai perangkat *display* monitoring secara *real-time* melalui antarmuka web (*web-based monitoring*).
3. Sistem ini memanfaatkan koneksi Wi-Fi lokal (misalnya hotspot dari HP) agar HP dapat mengakses antarmuka aplikasi *Flask* yang berjalan di laptop.

Berikut ini ditampilkan ilustrasi konfigurasi perangkat keras dan alur penggunaan sistem di dalam kabin kendaraan :



Gambar 3.5 Ilustrasi Implementasi Sistem Deteksi Kantuk pada Kendaraan

Alur kerja:

1. Kamera terhubung dengan laptop dan merekam wajah pengemudi secara *real-time*.
2. Laptop memproses *stream* video menggunakan metode EAR + CNN.
3. Hasil deteksi ditampilkan pada halaman web *Flask*.
4. HP pengemudi atau *co-driver* membuka halaman monitoring (misalnya <http://192.168.x.x:5000>) melalui browser.
5. HP akan menampilkan status :
 - *AWAKE* (warna hijau, EAR normal)
 - *DROWSY* (warna merah, $EAR < 0.25 + CNN$ mata tertutup)
6. Sistem juga dapat memicu alarm audio melalui speaker laptop atau output tambahan.

Manfaat penggunaan HP sebagai display monitoring :

1. Pengemudi dapat melihat status secara sekilas tanpa mengganggu fokus berkendara.
2. Posisi HP fleksibel, bisa diletakkan di *holder dashboard*.

3. Komunikasi *wireless* memudahkan integrasi dan pengembangan lebih lanjut.

3.5. Kebutuhan Hardware dan Software

Dalam penelitian ini, sistem deteksi kantuk pada pengemudi kendaraan roda empat menggunakan kombinasi metode *Haar Cascade*, *Eye Aspect Ratio* (EAR), dan *Convolutional Neural Network* (CNN). Untuk mendukung implementasi sistem ini, diperlukan spesifikasi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang memadai agar proses deteksi dapat berjalan secara optimal dan *real-time*.

3.5.1 Kebutuhan Hardware

Tabel 3.1 Kebutuhan Hardware

No	Komponen	Spesifikasi
1.	Prosesor	Intel Core i5/i7 atau AMD Ryzen 5/7
2.	RAM	8 GB
3.	GPU	NVIDIA GTX 1050 Ti
4.	Penyimpanan	SSD 475 GB
5.	Sistem Operasi	Windows 10/11
6.	Kamera Webcam	Resolusi minimal 720p, 30 FPS
7.	Speaker/Alarm	Untuk peringatan suara jika kantuk terdeteksi

Dengan spesifikasi perangkat keras yang telah ditetapkan, sistem deteksi kantuk ini diharapkan dapat berjalan secara optimal dan memberikan hasil deteksi yang akurat serta respons yang cepat dalam kondisi *real-time*.

3.5.2 Kebutuhan Software

Penelitian ini menggunakan *Python 3.8+* sebagai bahasa pemrograman utama dengan *OpenCV* untuk deteksi wajah dan mata secara *real-time* serta metode

Eye Aspect Ratio (EAR) untuk menganalisis kondisi mata pengemudi. Model deteksi kantuk dikembangkan menggunakan TensorFlow dan Keras dengan arsitektur Convolutional Neural Network (CNN). Pelatihan model dilakukan pada Google Colaboratory, yang mendukung komputasi berbasis GPU, sedangkan implementasi sistem dikembangkan menggunakan Visual Studio Code (VS Code). *Dataset* yang digunakan berasal dari Kaggle, yaitu *Drowsiness_dataset*, yang berisi gambar mata terbuka dan tertutup untuk melatih model deteksi kantuk.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Implementasi Sistem

Sistem deteksi kantuk yang dikembangkan pada penelitian ini diimplementasikan menggunakan bahasa pemrograman *Python* dengan *framework Flask* sebagai *backend* aplikasi desktop/web sederhana. Teknologi utama yang digunakan meliputi *OpenCV*, *MediaPipe FaceMesh*, dan model Convolutional Neural Network (CNN) yang telah dilatih sebelumnya dengan file model *best_drowsiness_model.h5*.

4.1.1 Alur Sistem Deteksi

Sistem berjalan dalam bentuk aplikasi *real-time* yang menangkap video wajah pengemudi melalui *webcam*, kemudian melakukan langkah-langkah berikut :

1. Deteksi Titik Wajah dan Mata

Deteksi dilakukan menggunakan *MediaPipe FaceMesh* untuk mendeteksi 468 titik wajah, khususnya area mata kiri dan kanan. Titik-titik ini digunakan untuk menghitung *Eye Aspect Ratio* (EAR) dari kedua mata.

2. Perhitungan EAR (*Eye Aspect Ratio*)

EAR dihitung menggunakan rumus geometri *Euclidean* berdasarkan enam titik mata kiri dan kanan. Nilai EAR yang rendah (< 0.25) menandakan bahwa mata dalam kondisi tertutup.

3. Klasifikasi Mata dengan CNN

Area mata kiri dan kanan di-crop, diubah ukurannya menjadi 64×64 piksel, dan dinormalisasi sebelum diklasifikasikan oleh model CNN. Model ini mengembalikan probabilitas mata terbuka. Jika probabilitas < 0.5 , maka mata dianggap tertutup.

4. Peringatan Kantuk

Sistem memberikan status “*DROWSY*” jika :

- a. Nilai EAR < 0.25 dan

- b. Kedua mata diklasifikasikan sebagai *closed* oleh CNN

Ketika kondisi ini terjadi, status ditampilkan dengan teks dan warna merah di layar, dan nilai *boolean alert_triggered* disiapkan untuk memicu alarm atau API peringatan.

4.1.2 Struktur Implementasi

1. *Framework* : *Flask* digunakan untuk *routing* dan menyajikan halaman HTML.
2. Model CNN : Dimuat menggunakan *load_model("best_drowsiness_model.h5")*.
3. *Streaming Video* : Kamera diakses melalui *cv2.VideoCapture(0)* dan *frame* disiarkan ke halaman web melalui *endpoint /video_feed* dengan format MJPEG.
4. Visualisasi :
 - a. EAR dan status CNN ditampilkan langsung pada layar.
 - b. Kotak warna hijau/merah menunjukkan apakah mata terbuka/tertutup.
 - c. Status keseluruhan (*AWAKE / DROWSY*) ditampilkan dengan font besar.

4.1.3 Kode Inti yang Menunjukkan Integrasi EAR + CNN

Table 4.1 Kode Logika Utama Deteksi Kantuk

Soure Code
<pre> if avg_EAR < EAR_THRESHOLD and cnn_status == "CNN: CLOSED": overall_status = "STATUS: DROWSY" overall_color = (0, 0, 255) alert_triggered = True else: overall_status = "STATUS: AWAKE" overall_color = (0, 255, 0) alert_triggered = False </pre>

Kode ini merupakan logika utama deteksi kantuk. Sistem akan menyatakan pengemudi mengantuk (*DROWSY*) hanya jika dua kondisi terpenuhi secara bersamaan :

1. Nilai EAR (*Eye Aspect Ratio*) rata-rata di bawah ambang batas (< 0.25)
2. Model CNN mengklasifikasikan kedua mata sebagai tertutup (CNN: *CLOSED*)

Jika kedua syarat ini terpenuhi, maka sistem menandai status sebagai *DROWSY*, memberi warna merah, dan mengaktifkan peringatan (*alert_triggered = True*). Jika tidak, sistem tetap dalam status *AWAKE* (mata terbuka), memberi warna hijau, dan alarm tidak dipicu.

Kombinasi dua metode ini (EAR + CNN) membantu meningkatkan akurasi deteksi dan mengurangi kesalahan karena kedipan singkat atau *noise*.

4.1.4 Arsitektur Endpoint Aplikasi

Tabel 4.2 Arsitektur Endpoint

Endpoint	Fungsi
/	Menyajikan halaman utama (<i>index.html</i>)
/video_vid	Menyediakan <i>streaming</i> kamera secara <i>real-time</i>
/status	Menyediakan status <i>alert_triggered</i> dalam format JSON
/logout	Endpoint simulasi keluar aplikasi

Endpoint-endpoint tersebut memungkinkan sistem berjalan secara interaktif dan responsif. *Streaming* video dari kamera diproses dan ditampilkan melalui */video_feed*, sedangkan status deteksi kantuk dapat diakses secara *real-time* melalui */status*, yang dapat pula digunakan untuk integrasi dengan sistem lain seperti alarm atau *dashboard* monitoring.

4.2. Pembangunan Model CNN

Pembangunan model CNN bertujuan untuk mengklasifikasikan kondisi mata (terbuka atau tertutup) berdasarkan citra hasil pemrosesan kamera, sebagai bagian utama dari sistem deteksi kantuk pengemudi. Proses pembangunan model ini meliputi prapemrosesan data, perancangan arsitektur, pelatihan model, dan evaluasi performa model.

4.2.1 Prapemrosesan Data

Dataset yang digunakan adalah *Drowsiness Dataset* dari *Kaggle*, terdiri dari dua kelas :

1. 726 gambar mata terbuka (*Open*)

2. 726 gambar mata tertutup (*Closed*)

Langkah-langkah prapemrosesan data :

1. Pembagian Dataset

Dataset dibagi menjadi :

- a. 508 gambar per kelas untuk pelatihan
- b. 109 gambar per kelas untuk validasi
- c. 109 gambar per kelas untuk pengujian

Pembagian dilakukan secara acak menggunakan *random.seed(42)* untuk konsistensi eksperimen.

Table 4.3 Pembagian Dataset

Source Code
<pre># Target jumlah gambar per subset split_config = { 'train': 508, 'val': 109, 'test': 109 } # Dua kelas classes = ['Open', 'Closed'] # Buat folder kosong untuk train/val/test per kelas for split in split_config: for cls in classes: Path(f'{target_base}/{split}/{cls}').mkdir(parents=True, exist_ok=True)</pre>

Tabel 4.4 Pembagian Dataset

Source Code
<pre># Fungsi pembagian dan penyalinan def split_and_copy(class_name): src_dir = os.path.join(base_dir, class_name) images = os.listdir(src_dir) random.shuffle(images) start = 0 for split in ['train', 'val', 'test']: n = split_config[split] split_imgs = images[start:start + n] for img_name in split_imgs: src_path = os.path.join(src_dir, img_name) dst_path = os.path.join(target_base, split, class_name, img_name) shutil.copy2(src_path, dst_path) start += n # Jalankan untuk open dan closed for cls in classes: split_and_copy(cls)</pre>

2. Augmentasi dan Normalisasi

Untuk meningkatkan generalisasi model, dilakukan augmentasi data khusus pada data latih dengan parameter :

- Rotasi acak hingga 10 derajat
- Zoom 10%
- Translasi horizontal dan vertical
- Flipping horizontal
- Semua gambar dinormalisasi ($rescale=1./255$) dan diubah ke ukuran 64x64 piksel

Tabel 4.5 Augmentasi dan Normalisasi Data

Source Code
<pre># Untuk data training: gunakan augmentasi agar model lebih robust train_datagen = ImageDataGenerator(rescale=1./255, rotation_range=10, zoom_range=0.1, width_shift_range=0.1, height_shift_range=0.1, horizontal_flip=True) # Untuk validation dan test: hanya normalisasi val_test_datagen = ImageDataGenerator(rescale=1./255)</pre>

3. Pemanggilan Data

Data yang telah diproses dan disimpan dalam folder *train/val/test* dimuat menggunakan fungsi *flow_from_directory()* dengan parameter :

- Ukuran gambar : 64×64 piksel
- Warna : RGB (3 channel)
- Mode klasifikasi : Biner (*binary*)

Tabel 4.6 Pemanggilan Data

Source Code
<pre># Ukuran gambar yang akan digunakan oleh CNN IMG_SIZE = (64, 64) BATCH_SIZE = 32</pre>

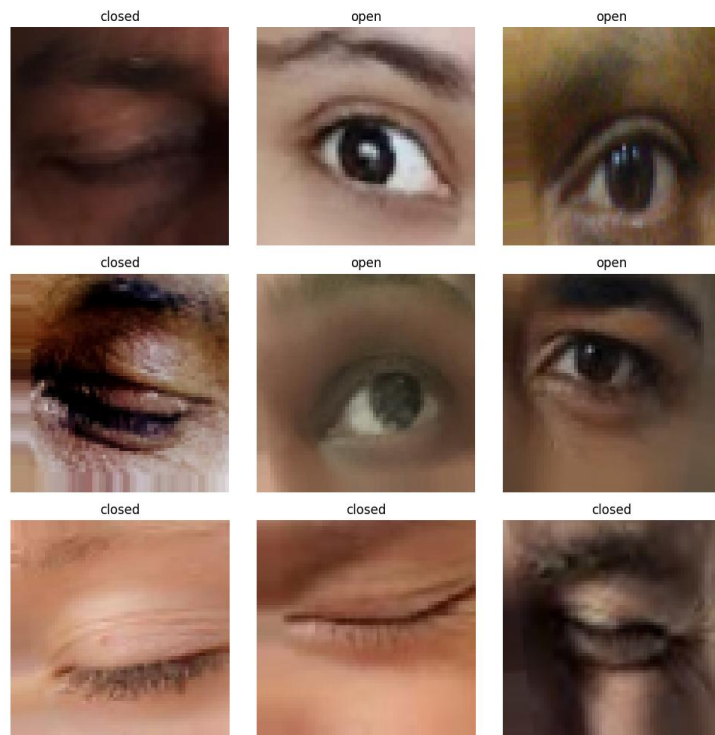
Tabel 4.7 Pemanggilan Data

Source Code
<pre># Train data train_data = train_datagen.flow_from_directory(train_dir, target_size=IMG_SIZE, batch_size=BATCH_SIZE, class_mode='binary') # Validation data val_data = val_test_datagen.flow_from_directory(val_dir, target_size=IMG_SIZE, batch_size=BATCH_SIZE, class_mode='binary') # Test data test_data = val_test_datagen.flow_from_directory(test_dir, target_size=IMG_SIZE, batch_size=BATCH_SIZE, class_mode='binary', shuffle=False # Jangan shuffle agar metrik evaluasi tepat)</pre>

4. Visualisasi Sample Gambar

Setelah proses augmentasi dan pemuatan dataset, dilakukan pengecekan visual terhadap sampel gambar dari data pelatihan. Tujuannya adalah untuk memastikan

bahwa proses augmentasi, normalisasi, serta pemetaan label telah berjalan dengan benar.



Gambar 4.1 Visualisasi Sample Gambar

Langkah ini juga membantu untuk memahami bagaimana data dilihat oleh model sebelum dilatih, serta memberikan gambaran awal terhadap variasi bentuk dan kondisi citra mata yang akan diproses oleh CNN.

4.2.2 Perancangan Arsitektur CNN

Model Convolutional Neural Network (CNN) dirancang untuk melakukan klasifikasi citra mata ke dalam dua kelas: mata terbuka (open) dan mata tertutup (closed). Arsitektur CNN pada penelitian ini disusun secara bertahap mulai dari lapisan konvolusi hingga ke lapisan output, dengan tujuan agar model mampu mengenali pola visual mata secara efektif.

Model dibangun menggunakan Sequential API dari Keras, dengan kombinasi lapisan konvolusi, batch normalization, pooling, dan dropout untuk mengurangi risiko overfitting. Arsitektur ini dirancang dengan mengacu pada praktik umum CNN untuk klasifikasi citra sederhana berukuran kecil (64x64 piksel).

1. Struktur Model CNN

Tabel 4.8 Struktur Model CNN

Blok	Jenis Layer	Konfigurasi
1	<i>Conv2D + BN</i>	<i>32 filter (3×3), ReLU, input_shape=(64,64,3) + BatchNormalization</i>
	<i>MaxPooling2D</i>	<i>Pool size (2×2)</i>
2	<i>Conv2D + BN</i>	<i>64 filter (3×3), ReLU + BatchNormalization</i>
	<i>MaxPooling2D</i>	<i>Pool size (2×2)</i>
3	<i>Conv2D + BN</i>	<i>128 filter (3×3), ReLU + BatchNormalization</i>
	<i>MaxPooling2D</i>	<i>Pool size (2×2), Dropout (0.3)</i>
FC	<i>Flatten + Dense</i>	<i>Dense(128), ReLU, Dropout (0.5)</i>
Out	<i>Output</i>	<i>Dense(1), activation sigmoid (klasifikasi biner)</i>

2. Kode Program Implementasi Arsitektur

Tabel 4.9 Kode Implementasi Arsitektur

Source Code
<pre>from tensorflow.keras.models import Sequential from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization from tensorflow.keras.optimizers import Adam</pre>

Tabel 4.10 Kode Implementasi Arsitektur

Source Code
<pre>model = Sequential([# Blok 1 Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)), BatchNormalization(), MaxPooling2D(pool_size=(2, 2)),</pre>

```

# Blok 2
Conv2D(64, (3, 3), activation='relu'),
BatchNormalization(),
MaxPooling2D(pool_size=(2, 2)),

# Blok 3
Conv2D(128, (3, 3), activation='relu'),
BatchNormalization(),
MaxPooling2D(pool_size=(2, 2)),
Dropout(0.3), # untuk cegah overfitting

# FC Layer
Flatten(),
Dense(128, activation='relu'),
Dropout(0.5),
Dense(1, activation='sigmoid') # binary classification
])

```

Model ini menggunakan aktivasi ReLU pada semua lapisan kecuali output, serta sigmoid pada output untuk menghasilkan nilai probabilitas klasifikasi biner. Dropout digunakan pada dua tahap akhir untuk mencegah overfitting.

3. Kompilasi Model

Sebelum dilakukan pelatihan, model perlu dikompilasi dengan pengaturan fungsi loss, optimizer, dan metrik evaluasi :

Tabel 4.11 Kode Kompilasi Model

Source Code
<pre> from tensorflow.keras.optimizers import Adam model.compile(</pre>

```
optimizer=Adam(learning_rate=0.0001),
loss='binary_crossentropy',
metrics=['accuracy']
)
```

Pada proses kompilasi model, beberapa parameter penting digunakan untuk mengatur proses pembelajaran, antara lain :

- a. *Loss function* : *binary_crossentropy*, karena tugas klasifikasi biner.
- b. *Optimizer* : *Adam* digunakan karena kemampuannya beradaptasi dengan learning rate secara dinamis.
- c. *Metrics* : Akurasi digunakan sebagai metrik utama

4. Ringkasan Model (Model Summary)

Model CNN yang dibangun terdiri dari total 13 lapisan, dengan total parameter sebesar 684.225. Dari jumlah tersebut, sebanyak 683.777 parameter dapat dilatih (*trainable*), sedangkan 448 merupakan parameter tidak dapat dilatih (*non-trainable*) yang berasal dari lapisan *BatchNormalization*. Model ini tergolong ringan dan efisien, sehingga dapat dilatih menggunakan GPU pada platform *Google Colaboratory* dalam waktu yang relatif singkat.

Tabel 4.12 Kode Program

Source Code
model.summary()

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
batch_normalization (BatchNormalization)	(None, 62, 62, 32)	128
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 29, 29, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 12, 12, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout (Dropout)	(None, 6, 6, 128)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 128)	589,952
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129
Total params: 684,225 (2.61 MB)		
Trainable params: 683,777 (2.61 MB)		
Non-trainable params: 448 (1.75 KB)		

Gambar 4.2 Ringkasan Arsitektur Model

4.2.3 Pelatihan Model CNN

Setelah perancangan arsitektur model selesai, tahap selanjutnya adalah proses pelatihan (*training*) model menggunakan dataset yang telah diproses. Tujuan dari pelatihan ini adalah agar model dapat belajar mengenali pola visual dari citra mata terbuka dan tertutup secara akurat.

1. Kompilasi Model

Sebelum model dilatih, dilakukan proses kompilasi dengan pengaturan sebagai berikut:

- a. *Optimizer* : *Adam*, dengan *learning rate* sebesar 0.0001

- b. *Loss Function : Binary Crossentropy*, karena klasifikasi bersifat biner (dua kelas)
- c. *Metrics : Accuracy*, untuk mengevaluasi kinerja model selama dan setelah pelatihan

Tabel 4.13 Kode Kompilasi Model

Source Code
<pre>model.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=['accuracy'])</pre>

2. Konfigurasi *Callbacks*

Agar proses pelatihan berjalan optimal dan efisien, digunakan dua *callback* utama :

a. *EarlyStopping*

Berfungsi untuk menghentikan pelatihan secara otomatis jika nilai *val_loss* tidak mengalami perbaikan selama 5 *epoch* berturut-turut. Ini mencegah *overfitting* dan menghemat waktu komputasi.

b. *ModelCheckpoint*

Berfungsi untuk menyimpan bobot model terbaik secara otomatis berdasarkan nilai *val_loss* terendah selama pelatihan.

Tabel 4.14 Kode Konfigurasi Callback

Source Code
<pre>from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint checkpoint_path = "/content/drive/MyDrive/best_drowsiness_model.h5" callbacks = [EarlyStopping(</pre>

```

        monitor='val_loss',
        patience=5,          # stop training jika val_loss tidak membaik 5 epoch
berturut-turut
        restore_best_weights=True
    ),
    ModelCheckpoint(
        filepath=checkpoint_path,
        monitor='val_loss',
        save_best_only=True
    )
]

```

3. Proses Pelatihan

Model dilatih menggunakan data latih dan divalidasi menggunakan data validasi selama maksimal 30 *epoch*, atau sampai terjadi *early stopping*. *Batch size* yang digunakan adalah 32, yang berarti model memproses 32 gambar dalam satu iterasi.

Tabel 4.15 Kode Pelatihan

Source Code
<pre> history = model.fit(train_data, validation_data=val_data, epochs=30, callbacks=callbacks) </pre>

```

Epoch 1/30
32/32 ----- 0s 433ms/step - accuracy: 0.8725 - loss: 0.6125WARNING:absl:You are saving your model as an
32/32 ----- 21s 556ms/step - accuracy: 0.8732 - loss: 0.6065 - val_accuracy: 0.9464 - val_loss: 0.2824
Epoch 2/30
32/32 ----- 0s 450ms/step - accuracy: 0.9350 - loss: 0.2627WARNING:absl:You are saving your model as an
32/32 ----- 16s 507ms/step - accuracy: 0.9351 - loss: 0.2616 - val_accuracy: 0.9633 - val_loss: 0.1936
Epoch 3/30
32/32 ----- 16s 491ms/step - accuracy: 0.9215 - loss: 0.3059 - val_accuracy: 0.9495 - val_loss: 0.3415
Epoch 4/30
32/32 ----- 0s 449ms/step - accuracy: 0.9567 - loss: 0.1522WARNING:absl:You are saving your model as an
32/32 ----- 17s 544ms/step - accuracy: 0.9567 - loss: 0.1524 - val_accuracy: 0.9358 - val_loss: 0.1923
Epoch 5/30
32/32 ----- 0s 449ms/step - accuracy: 0.9587 - loss: 0.1119WARNING:absl:You are saving your model as an
32/32 ----- 16s 503ms/step - accuracy: 0.9587 - loss: 0.1125 - val_accuracy: 0.9725 - val_loss: 0.1098
Epoch 6/30
32/32 ----- 16s 493ms/step - accuracy: 0.9440 - loss: 0.2267 - val_accuracy: 0.9725 - val_loss: 0.1454
Epoch 7/30
32/32 ----- 0s 449ms/step - accuracy: 0.9735 - loss: 0.1033WARNING:absl:You are saving your model as an
32/32 ----- 18s 547ms/step - accuracy: 0.9734 - loss: 0.1035 - val_accuracy: 0.9908 - val_loss: 0.0162
Epoch 8/30
32/32 ----- 19s 490ms/step - accuracy: 0.9715 - loss: 0.2134 - val_accuracy: 0.9771 - val_loss: 0.0456
Epoch 9/30
32/32 ----- 17s 530ms/step - accuracy: 0.9749 - loss: 0.0790 - val_accuracy: 0.9954 - val_loss: 0.0200
Epoch 10/30
32/32 ----- 17s 533ms/step - accuracy: 0.9795 - loss: 0.0842 - val_accuracy: 0.9817 - val_loss: 0.0341
Epoch 11/30
32/32 ----- 16s 489ms/step - accuracy: 0.9787 - loss: 0.0656 - val_accuracy: 0.9908 - val_loss: 0.0379
Epoch 12/30
32/32 ----- 17s 527ms/step - accuracy: 0.9700 - loss: 0.0863 - val_accuracy: 0.9725 - val_loss: 0.0627

```

Gambar 4.3 Hasil Pelatihan

Untuk melihat perkembangan performa model selama pelatihan, dilakukan analisis terhadap tren akurasi dan *loss* pada data pelatihan dan validasi. Hasil pengamatannya dijelaskan sebagai berikut :

- a. Akurasi Pelatihan meningkat secara konsisten dari 87% hingga mencapai 97–98% pada *epoch* akhir.
 - b. Akurasi Validasi juga meningkat secara signifikan, mencapai 99.54% pada *epoch* ke-9.
 - c. *Val_loss* mencapai nilai minimum sebesar 0.0162 pada *epoch* ke-7, yang menunjukkan performa terbaik model terhadap data validasi.
 - d. Tidak terjadi *overfitting*, karena selisih antara akurasi pelatihan dan validasi tetap rendah.
 - e. Penggunaan *EarlyStopping* berhasil mencegah pelatihan berlebihan.
4. Hasil Pelatihan

Selama proses pelatihan, sistem mencatat nilai akurasi dan *loss* untuk data pelatihan maupun validasi pada setiap *epoch*. Setelah pelatihan selesai, dilakukan visualisasi terhadap hasil pelatihan.

Tabel 4.16 Kode Visualisasi Hasil Pelatihan

Source Code

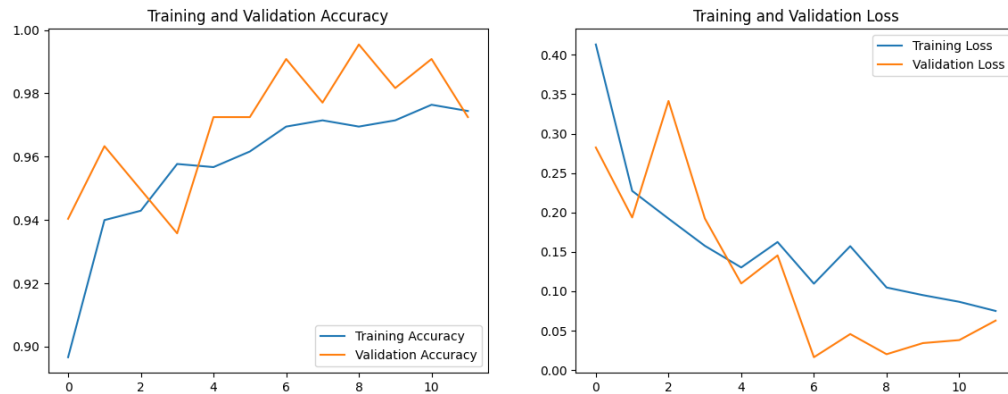
```
def plot_training_history(history):
    acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']
    loss = history.history['loss']
    val_loss = history.history['val_loss']
    epochs_range = range(len(acc))

    plt.figure(figsize=(14, 5))

    # Accuracy
    plt.subplot(1, 2, 1)
    plt.plot(epochs_range, acc, label='Training Accuracy')
    plt.plot(epochs_range, val_acc, label='Validation Accuracy')
    plt.legend(loc='lower right')
    plt.title('Training and Validation Accuracy')

    # Loss
    plt.subplot(1, 2, 2)
    plt.plot(epochs_range, loss, label='Training Loss')
    plt.plot(epochs_range, val_loss, label='Validation Loss')
    plt.legend(loc='upper right')
    plt.title('Training and Validation Loss')
    plt.show()

plot_training_history(history)
```



Gambar 4.4 Visualisasi Hasil Gambar

Berikut pembahasan hasil grafik pelatihan dan validasi model berdasarkan akurasi, *loss*, dan mekanisme *early stopping* :

- Pada grafik kiri, terlihat bahwa akurasi pelatihan dan validasi meningkat stabil dari awal hingga akhir pelatihan. Akurasi validasi bahkan mencapai nilai puncak sebesar 99,54% pada *epoch* ke-9.
- Grafik kanan menunjukkan bahwa nilai *loss* pada data validasi menurun drastis, terutama antara *epoch* ke-5 hingga ke-7. Ini menunjukkan bahwa model mampu mempelajari pola penting tanpa mengalami *overfitting*.
- Pola kurva yang konsisten dan tidak berlawanan arah menandakan bahwa model memiliki generalitas yang baik. Tidak terdapat perbedaan besar antara akurasi/*loss* pelatihan dan validasi, yang biasanya menjadi indikator awal *overfitting*.
- Early stopping* secara otomatis menghentikan pelatihan ketika performa tidak lagi meningkat, sehingga model yang disimpan adalah model terbaik berdasarkan nilai *val_loss* terendah.

4.2.4 Evaluasi Model CNN

Setelah proses pelatihan selesai dan model terbaik disimpan, dilakukan evaluasi terhadap performa model menggunakan data uji (test data) yang sebelumnya tidak pernah digunakan dalam proses pelatihan ataupun validasi. Evaluasi dilakukan dalam beberapa tahap, yaitu perhitungan akurasi, prediksi label, *confusion matrix*, dan *classification report*.

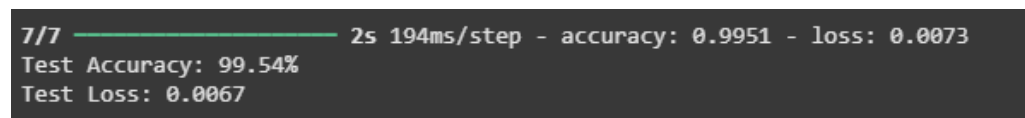
1. Evaluasi Akurasi Terhadap Data Uji

Model dievaluasi menggunakan `model.evaluate()` pada dataset uji. Metrik yang diukur adalah :

- Accuracy* : persentase prediksi benar dari seluruh data uji.
- Loss* : nilai *loss binary crossentropy* pada data uji.

Tabel 4.17 Kode Evaluasi Akurasi Terhadap Data Uji

Source Code
<pre>test_loss, test_acc = model.evaluate(test_data) print(f'Test Accuracy: {test_acc*100:.2f}%") print(f'Test Loss: {test_loss:.4f}")</pre>



```
7/7 ————— 2s 194ms/step - accuracy: 0.9951 - loss: 0.0073
Test Accuracy: 99.54%
Test Loss: 0.0067
```

Gambar 4.5 Hasil Evaluasi Akurasi Terhadap Data Uji

Berikut pembahasan hasil evaluasi model berdasarkan nilai akurasi, *loss*, dan kualitas arsitektur serta pelatihan :

- Nilai akurasi yang sangat tinggi (mendekati 100%) menunjukkan bahwa model berhasil mengklasifikasikan citra mata pada data uji dengan sangat baik.
 - Nilai *loss* yang sangat rendah (0,0067) menandakan bahwa prediksi model sangat dekat dengan label sebenarnya, dan tidak terjadi *overfitting*.
 - Hal ini menunjukkan bahwa arsitektur model, teknik augmentasi, serta proses pelatihan yang diterapkan telah memberikan hasil yang optimal.
- ## 2. Prediksi Terhadap Data Uji

Model digunakan untuk memprediksi seluruh citra pada data uji menggunakan `model.predict()`. Karena output model adalah probabilitas, maka dikonversi menjadi label 0 atau 1 menggunakan ambang batas (*threshold*) 0.5 :

Tabel 4.18 Kode Prediksi Terhadap Data Uji

Source Code
<pre>pred_prob = model.predict(test_data) pred_labels = (pred_prob > 0.5).astype(int).reshape(-1)</pre>

Model menghasilkan angka probabilitas, lalu angka itu dikonversi menjadi label kelas berdasarkan ambang batas 0.5 :

- a. Label 0: Mata tertutup (*Closed*)
- b. Label 1: Mata terbuka (*Open*)

3. *Confusion Matrix*

Confusion matrix digunakan untuk mengevaluasi performa klasifikasi model secara rinci, dengan menunjukkan jumlah prediksi benar dan salah terhadap masing-masing kelas. Matriks ini sangat penting untuk melihat jenis kesalahan apa yang paling sering terjadi dan bagaimana distribusi prediksi model.

Tabel 4.19 Kode Confusion Matrix

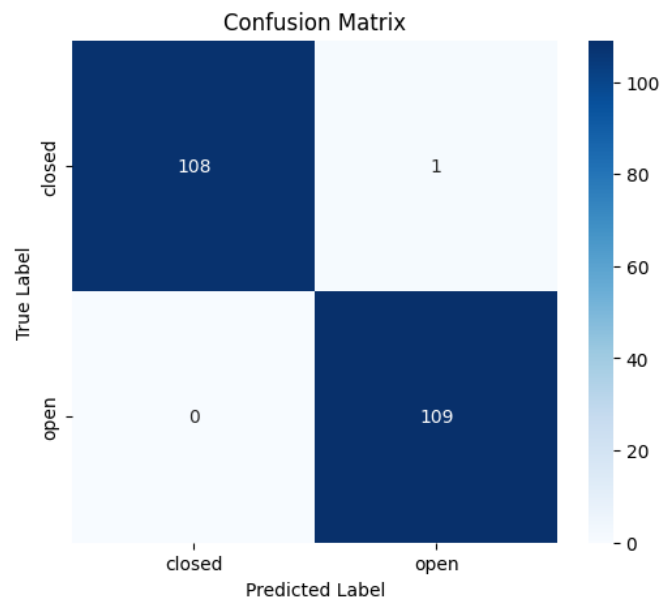
Source Code
<pre>import numpy as np from sklearn.metrics import confusion_matrix, classification_report import seaborn as sns import matplotlib.pyplot as plt # Prediksi label test data test_data.reset() # reset generator agar prediksi mulai dari awal pred_prob = model.predict(test_data) pred_labels = (pred_prob > 0.5).astype(int).reshape(-1) # Label asli true_labels = test_data.classes</pre>

```

cm = confusion_matrix(true_labels, pred_labels)
class_names = ['closed', 'open']

plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=class_names, yticklabels=class_names)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()

```



Gambar 4.6 Hasil Confusion Matriks

Berdasarkan *confusion matrix* yang dihasilkan, berikut ini beberapa poin penting yang menggambarkan kinerja model dalam mengklasifikasikan kondisi mata pada data uji :

- Tingkat akurasi sangat tinggi dengan hanya 1 kesalahan klasifikasi dari 218 data uji.
- Model tidak pernah mengeluarkan *false* alarm (tidak ada mata terbuka yang salah terdeteksi sebagai tertutup).

- c. Hal ini sangat penting dalam sistem deteksi kantuk karena kesalahan mendeteksi “mata terbuka” sebagai “tertutup” dapat menyebabkan alarm palsu.

4. *Classification Report*

Evaluasi lebih lanjut dilakukan dengan menggunakan *classification report* dari pustaka *Scikit-learn*, yang memberikan metrik evaluasi berupa *precision*, *recall*, dan *F1-score* untuk masing-masing kelas. Evaluasi ini menggunakan label asli (*true_labels*) dan hasil prediksi (*pred_labels*) terhadap data uji.

Tabel 4.20 Kode Classification Report

Source Code
<pre>print(classification_report(true_labels, pred_labels, target_names=class_names))</pre>

	precision	recall	f1-score	support
closed	1.00	0.99	1.00	109
open	0.99	1.00	1.00	109
accuracy			1.00	218
macro avg	1.00	1.00	1.00	218
weighted avg	1.00	1.00	1.00	218

Gambar 4.7 Hasil Classification Report

Evaluasi lebih lanjut dilakukan dengan *classification report* untuk melihat performa model pada setiap kelas, khususnya dalam hal *precision*, *recall*, dan *F1-score*. Hasilnya dijelaskan sebagai berikut :

- Kelas "*closed*" (mata tertutup) memiliki *recall* sebesar 0.99, artinya 99% dari seluruh gambar mata tertutup berhasil diklasifikasikan dengan benar.
- Kelas "*open*" (mata terbuka) memiliki *precision* 0.99 dan *recall* 1.00, menunjukkan bahwa prediksi "*open*" hampir seluruhnya benar.
- Nilai *F1-score* untuk kedua kelas adalah 1.00, yang mencerminkan keseimbangan sempurna antara *precision* dan *recall*.

- d. Nilai *accuracy* keseluruhan mencapai 100%, menandakan bahwa model mampu mengklasifikasikan data uji dengan sangat baik dan stabil.

4.3. Implementasi dan Uji Coba Sistem

4.3.1 Integrasi Sistem

Sistem deteksi kantuk berbasis web yang telah dibangun mengombinasikan tiga komponen utama, yaitu :

- Eye Aspect Ratio* (EAR) untuk menghitung rasio aspek mata yang menjadi indikator utama kondisi mata terbuka atau tertutup,
- MediaPipe FaceMesh* sebagai pengganti *Haar Cascade* untuk mendeteksi 468 titik landmark wajah, khususnya area mata, dengan akurasi tinggi secara real-time, dan
- Convolutional Neural Network* (CNN) yang berfungsi sebagai klasifikasi akhir untuk mendeteksi apakah mata pengguna dalam keadaan terbuka (OPEN) atau tertutup (CLOSED) berdasarkan citra mata hasil pemotongan (cropping) dari kamera.

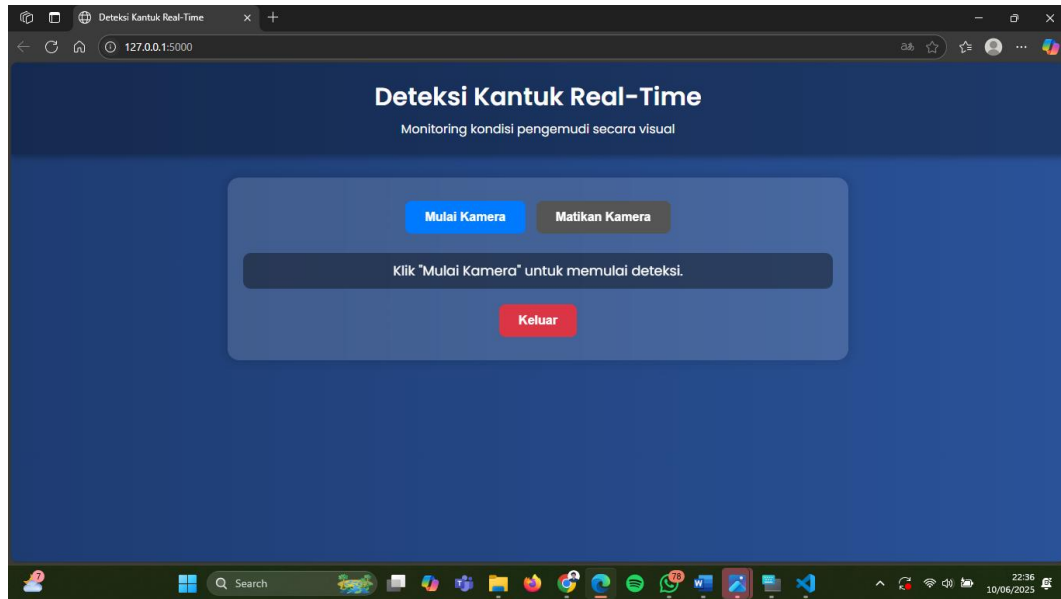
Ketiga komponen ini berjalan secara simultan dalam aplikasi web yang dirancang menggunakan *framework Flask*. Hasil deteksi ditampilkan secara langsung pada halaman web, lengkap dengan status EAR, hasil klasifikasi CNN, dan status akhir pengemudi (AWAKE/DROWSY).

4.3.2 Antarmuka Web

Web dibangun menggunakan Flask dengan endpoint sebagai berikut :

Tabel 4.21 Endpoin Web

Endpoint	Fungsi
/	Menyajikan halaman utama (<i>index.html</i>)
/video_vid	Menyediakan <i>stream</i> MJPEG dari webcam
/status	Menyediakan status <i>alert_triggered</i> dalam format JSON
/logout	Endpoint simulasi keluar aplikasi



Gambar 4.8 Halaman Utama Web

Halaman utama aplikasi web menampilkan pemantauan kondisi pengemudi secara *real-time* melalui video dari *webcam*. Sistem secara otomatis menghitung nilai EAR, mengklasifikasikan kondisi mata dengan CNN, dan menampilkan status “*AWAKE*” atau “*DROWSY*” dengan indikator warna. Antarmuka ini responsif dan dapat diakses melalui desktop maupun perangkat *mobile*, memudahkan pemantauan selama berkendara.

4.3.3 Hasil Uji Coba

Uji coba dilakukan untuk mengevaluasi performa sistem deteksi kantuk secara *real-time* berdasarkan kombinasi nilai *Eye Aspect Ratio* (EAR) dan hasil klasifikasi dari model CNN, dengan bantuan pelacakan fitur wajah menggunakan *MediaPipe FaceMesh*. Sistem diuji menggunakan webcam dan ditampilkan melalui antarmuka web yang dibangun menggunakan *Flask*.

Untuk mendeteksi kantuk, sistem menetapkan dua syarat utama agar status *DROWSY* aktif:

1. Nilai EAR < 0.25 secara berurutan dalam beberapa *frame*.
2. CNN mengklasifikasikan kedua mata sebagai *CLOSED* (tertutup).

Jika kedua syarat ini terpenuhi, maka sistem mengubah status menjadi *DROWSY*, menampilkan notifikasi warna merah, dan memicu variabel *alert_triggered = True*.

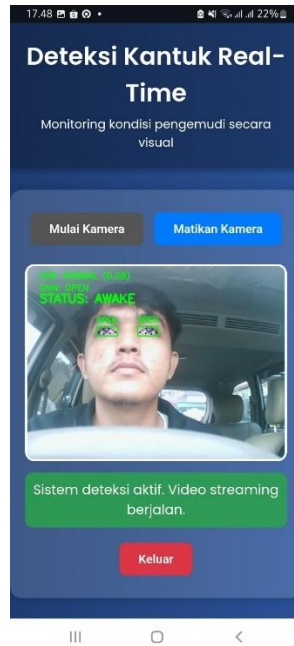
Berikut adalah hasil dokumentasi visual dari dua skenario yang diuji :

1. Skenario Pertama (Kondisi Terjaga)



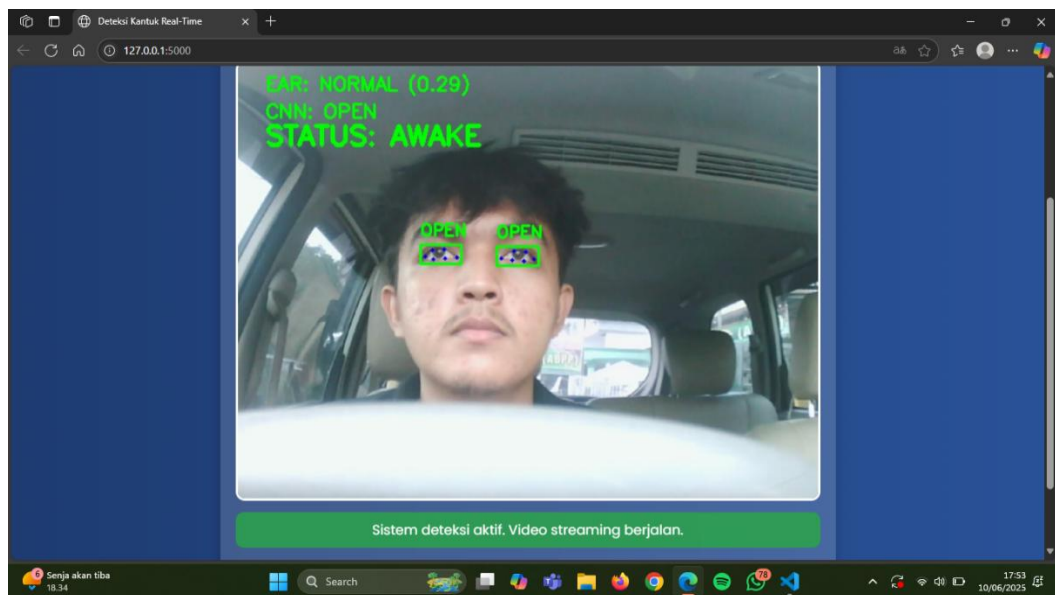
Gambar 4.9 Kondisi Terjaga (Normal) pada Pengujian Langsung di Kendaraan

Uji coba di dalam kendaraan menunjukkan hasil serupa. Kamera mendeteksi wajah dengan baik, dan status tetap *AWAKE*, dengan EAR tinggi dan klasifikasi CNN : *OPEN*. Tampilan di layar *smartphone* tetap stabil meski berada di lingkungan nyata.



Gambar 4.10 Kondisi Terjaga (Normal) pada Tampilan Web *Mobile*

Saat diuji melalui *smartphone*, sistem juga menunjukkan status *AWAKE* dengan indikator hijau. Nilai *EAR* tetap tinggi dan *CNN* mendeteksi mata terbuka, membuktikan sistem dapat diakses fleksibel tanpa mengurangi akurasi.



Gambar 4.11 Kondisi Terjaga (Normal) pada Tampilan Web *Desktop*

Pada tampilan web *desktop*, sistem mendeteksi pengemudi dalam kondisi sadar. Nilai $EAR = 0.29$ dan *CNN* mengklasifikasikan mata sebagai *OPEN*, sehingga status

ditampilkan sebagai “*AWAKE*” dengan warna hijau. Visualisasi area mata ditampilkan secara *real-time*, menunjukkan akurasi deteksi sistem.

2. Skenario Kedua (Kondisi Mengantuk)



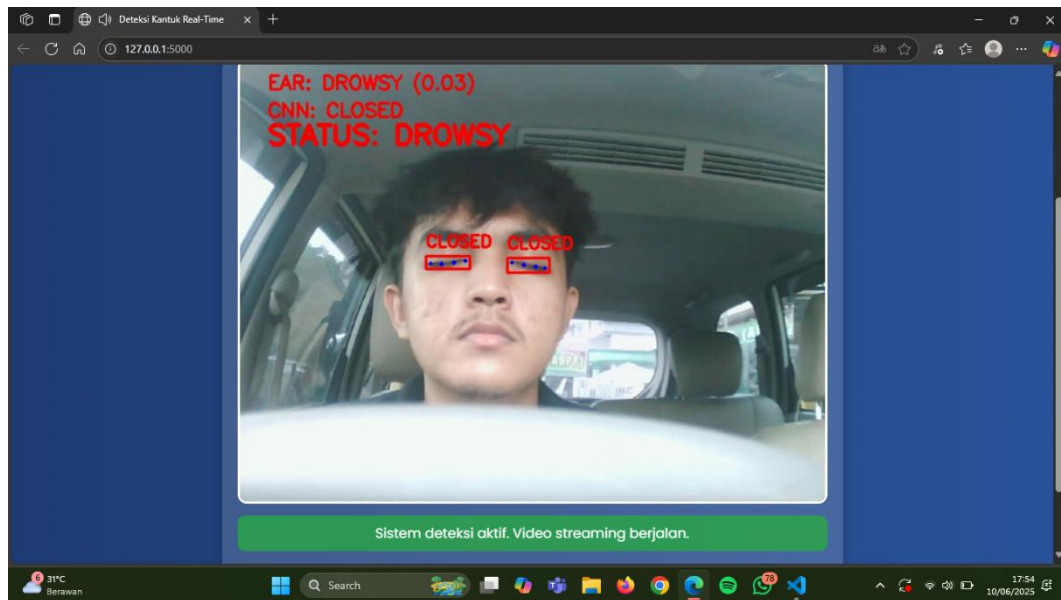
Gambar 4.12 Deteksi Kantuk Aktif pada Pengujian Langsung di Kendaraan

Pada pengujian di dalam mobil dengan mata tertutup, sistem tetap mendeteksi status *DROWSY*. EAR sangat rendah dan CNN sesuai, sehingga peringatan visual merah muncul dengan respons cepat dan akurat.



Gambar 4.13 Deteksi Kantuk Aktif pada Tampilan Web *Mobile*

Melalui tampilan smartphone, status *DROWSY* juga muncul jelas saat mata tertutup. EAR rendah dan CNN mendeteksi kondisi tertutup, menandakan bahwa sistem siap memberi alarm meski dipantau dari perangkat *mobile*.



Gambar 4.14 Deteksi Kantuk Aktif pada Tampilan Web *Desktop*

Ketika mata sengaja ditutup, sistem mendeteksi $EAR = 0.03$ dan CNN mengklasifikasikan mata sebagai *CLOSED*. Status langsung berubah menjadi “*DROWSY*” dengan warna merah, menandakan kondisi kantuk dan memicu sistem peringatan.

Secara keseluruhan, hasil uji coba membuktikan bahwa sistem mampu mendeteksi perbedaan antara kondisi mata terbuka dan tertutup secara konsisten dalam berbagai kondisi. Sistem juga terbukti dapat memberikan peringatan secara cepat dan responsif baik pada tampilan *desktop* maupun *mobile*. Integrasi metode EAR dan CNN terbukti memperkuat keakuratan deteksi, sekaligus mengurangi kesalahan yang mungkin muncul karena kedipan biasa atau gangguan pencahayaan. Sistem ini dinilai siap untuk diterapkan dalam kendaraan sebagai salah satu upaya pencegahan kecelakaan akibat kantuk.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai implementasi sistem deteksi rasa kantuk pada pengemudi kendaraan roda empat menggunakan *OpenCV*, *MediaPipe FaceMesh*, dan *Convolutional Neural Network* (CNN), maka dapat disimpulkan hal-hal sebagai berikut:

1. Sistem berhasil mendeteksi kondisi mata pengemudi (terbuka atau tertutup) secara *real-time* dengan menggunakan metode *Eye Aspect Ratio* (EAR) yang diukur dari titik-titik *landmark* wajah yang diperoleh melalui *MediaPipe FaceMesh*, tanpa memerlukan perangkat keras tambahan.
2. Integrasi model CNN yang dilatih menggunakan *dataset* dari *Kaggle* mampu mengklasifikasikan kondisi mata (terbuka atau tertutup) dengan akurasi yang baik, sehingga memperkuat validitas sistem dalam mengidentifikasi tanda-tanda kantuk.
3. Sistem memberikan peringatan dini dalam bentuk alarm visual dan audio secara otomatis apabila kondisi mata terdeteksi tertutup selama 2 detik berturut-turut, dengan nilai $EAR < 0.25$ dan hasil klasifikasi CNN menunjukkan mata tertutup.
4. Penggunaan *MediaPipe FaceMesh* dipilih sebagai alternatif dari *Haar Cascade* karena mampu memberikan deteksi titik wajah (*facial landmarks*) yang lebih stabil dan presisi, serta mendukung proses perhitungan EAR dengan lebih akurat.
5. Aplikasi ini awalnya dirancang untuk diimplementasikan sebagai program desktop berbasis *Python* menggunakan antarmuka *Tkinter*. Namun, dalam proses pengembangan, sistem akhirnya diimplementasikan menggunakan *Flask* sebagai antarmuka web untuk alasan fleksibilitas dan kemudahan akses lintas perangkat. Sistem bekerja secara responsif dan stabil dalam simulasi, sehingga tetap memiliki

potensi besar untuk dikembangkan lebih lanjut sebagai fitur keselamatan pada kendaraan roda empat.

5.2. Saran

Beberapa saran yang dapat diberikan untuk pengembangan sistem ini ke depan adalah sebagai berikut :

1. Sistem perlu diuji lebih lanjut pada kondisi nyata, seperti di dalam kendaraan yang sedang berjalan, dengan berbagai variasi pencahayaan, sudut wajah, dan gangguan lingkungan lainnya untuk mengukur ketahanannya dalam situasi nyata.
2. Disarankan untuk mengembangkan sistem dalam bentuk aplikasi *mobile* atau IoT berbasis *embedded system*, misalnya dengan menggunakan *Raspberry Pi* atau *TensorFlow Lite*, agar lebih mudah diintegrasikan dengan kendaraan secara langsung.
3. Sistem dapat ditingkatkan dengan menambahkan indikator lain seperti deteksi menguap (*yawning*) atau ekspresi wajah lelah untuk menghasilkan deteksi kantuk yang lebih komprehensif.
4. Perlu dilakukan optimalisasi terhadap sensitivitas sistem terhadap posisi wajah yang miring atau berpaling, dengan memanfaatkan fitur pelacakan adaptif lanjutan dari *MediaPipe* atau algoritma lain seperti *head pose estimation*.
5. Penggunaan *MediaPipe FaceMesh* sebagai pengganti *Haar Cascade* telah terbukti meningkatkan stabilitas dalam deteksi *landmark* wajah. Namun, pengujian lebih lanjut dalam kondisi ekstrem dan penerapan pada perangkat keras portabel tetap diperlukan.

DAFTAR PUSTAKA

- Asvin Mahersatillah Suradi, A., Alam, S., Furqan Rasyid, M., Djafar, I., Dipa Makassar, U., & Perintis Kemerdekaan, J. K. (n.d.). *Sistem Deteksi Kantuk Pengemudi Mobil Berdasarkan Analisis Rasio Mata Menggunakan Computer Vision*.
- Fadiil Thoriq, M., Fathi Ramdhana, M., Rahma, D. N., Putri, N. A., Zahir, R. H., Mindara, G. P., & Giri, P. (2024). Implementasi Sistem Deteksi Kantuk Secara Real-Time Bagi Pengemudi Menggunakan Metode Eye Aspect Ratio. *Jurnal Sistem Informasi Dan Ilmu Komputer*, 2, 70–85. <https://doi.org/10.59581/jusiik-widyakarya.v2i2.4226>
- Firdaus, A., Utaminingrum, F., & Widasari, E. R. (2023). *Sistem Pendeteksi Kantuk Pengemudi berbasis Eye Aspect Ratio dan Mouth Opening Ratio menggunakan Algoritme C-LSTM* (Vol. 7, Issue 2). <http://j-ptiik.ub.ac.id>
- Iswahyudi, R., Fauzi Ikhsan, A., Muhamad, I., & Matin, M. (n.d.). *Fuse-teknik Elektro DETEKSI KANTUK MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) BERDASARKAN KEDIPAN MATA*.
- Pada, M. L., Kendaraan, P., Berbasis, B., Citra, P., Samsinar, R., Isyanto, H., Almanda, D., & Amrullah, F. (n.d.). *Sistem Monitoring Mendeteksi*. 7(1).
- Riski Fauzi, M., Ibnutama, K., Komputer, S., & Triguna Dharma, S. (n.d.). *Penerapan Algoritma Haar Cascade Eye Detection Untuk Sistem Peringatan Keamanan Pada Pengemudi Mobil*. <https://ojs.trigunadharma.ac.id/index.php/jskom>
- Saputra, C. A., Erwanto, D., Rahayu, P. N., & Kadiri, I. (2021). DETEKSI KANTUK PENGENDARA RODA EMPAT MENGGUNAKAN HAAR CASCADE CLASSIFIER dan CONVOLUTIONAL NEURAL NETWORK. *JEECOM*, 3(1).
- Schuhmann, L., Büchner, T., Heinrich, M., Volk, G. F., Denzler, J., & Guntinas-Lichius, O. (2024). Automated analysis of spontaneous eye blinking in patients with acute facial palsy or facial synkinesis. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-68707-x>
- SISTEM DETEKSI DRIVER DROWSINESS MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)*. (n.d.).
- SISTEM DETEKSI PENYAKIT DAUN APEL MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK BERBASIS MOBILE*. (n.d.).
- Suyuti, M. (n.d.). *Lembar Pengesahan Pembimbing Pengembangan Model Klasifikasi Mata Tertutup dan Terbuka Dalam Identifikasi Kelelahan Menggunakan Arsitektur Mobile CNN*.
- Teknika, J., Faiq, H. A., & Sabita, H. (n.d.-b). Teknik 18 (1): 197-209 Pengembangan Model Deep Learning Untuk Pengenalan Wajah pada Sistem Keamanan. *IJCCS*, x, No.x, 1–5.

- Transparan Putra Zebua, E., & Rosyani, P. (2024). Perancangan Deteksi Objek Kendaraan Bermotor Berbasis OpenCV Python menggunakan Metode HOG-SVM untuk Analisis Lalu Lintas Cerdas. In *Jurnal Artificial Intelligent dan Sistem Penunjang Keputusan* (Vol. 2, Issue 1). <https://jurnalmahasiswa.com/index.php/aidanspk>
- Yulina, S. (2021). 100-109 Dokumen diterima pada 21 Januari. In *Jurnal Komputer Terapan* (Vol. 7, Issue 1). <https://jurnal.pcr.ac.id/index.php/jkt/>
- Zein, A., Raya, J., Serpong, P., 10 Tangerang, N., & Banten, S. (2018). *Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON Real Time Sleepiness Detection Using OPENCV Library and PYTHON DLIB* (Vol. 28, Issue 2).

ORIGINALITY REPORT

20%	17%	9%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	repository.umsu.ac.id Internet Source	2%
2	Submitted to Universitas Muhammadiyah Sumatera Utara Student Paper	2%
3	Submitted to Fakultas Teknik Student Paper	1%
4	Submitted to Munster Technological University (MTU) Student Paper	1%
5	Submitted to BB9.1 PROD Student Paper	<1%
6	ejurnal.seminar-id.com Internet Source	<1%
7	Submitted to Perpustakaan Student Paper	<1%
8	repository.its.ac.id Internet Source	<1%
9	etd.repository.ugm.ac.id Internet Source	<1%
10	123dok.com Internet Source	<1%
11	Submitted to University of Huddersfield Student Paper	<1%

12	digilib.uin-suka.ac.id Internet Source	<1 %
13	ejournal.unuja.ac.id Internet Source	<1 %
14	repository.usbypkp.ac.id Internet Source	<1 %
15	Submitted to Deakin University Student Paper	<1 %
16	repository.ub.ac.id Internet Source	<1 %
17	Submitted to Universitas Putera Batam Student Paper	<1 %
18	Adinne Islamiyati, Ratih Titi Komalasari. "IDENTIFIKASI MENGANTUK MENGGUNAKAN ALGORITMA CNN", Jurnal Ilmiah Multidisiplin, 2025 Publication	<1 %
19	Submitted to Dublin City University Student Paper	<1 %
20	jurnalunived.com Internet Source	<1 %
21	publikasi.dinus.ac.id Internet Source	<1 %
22	Submitted to Chester College of Higher Education Student Paper	<1 %
23	Submitted to University of Hertfordshire Student Paper	<1 %

24	Internet Source	<1 %
25	repository.usni.ac.id Internet Source	<1 %
26	github.com Internet Source	<1 %
27	gitlab.sliit.lk Internet Source	<1 %
28	www.researchgate.net Internet Source	<1 %
29	Submitted to Universitas Negeri Jakarta Student Paper	<1 %
30	Yudisman Ferdinan Bili, Tundo, Nandang Sutisna, Atsilah Daini Putri, Dita Tri Yuliantoro, Laily Nurmayanti. "Prediksi Motif Batik dengan Menggunakan Metode Gabor Filter Convolution Neural Network", Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 2025 Publication	<1 %
31	dspace.uui.ac.id Internet Source	<1 %
32	repository.upnvj.ac.id Internet Source	<1 %
33	researchportal.port.ac.uk Internet Source	<1 %
34	Submitted to Universitas Muhammadiyah Sukabumi Student Paper	<1 %

35	Internet Source	<1 %
36	repository.upnjatim.ac.id Internet Source	<1 %
37	stackoverflow.com Internet Source	<1 %
38	www.coursehero.com Internet Source	<1 %
39	core.ac.uk Internet Source	<1 %
40	origin.geeksforgeeks.org Internet Source	<1 %
41	Submitted to Multimedia University Student Paper	<1 %
42	eprints.amikompurwokerto.ac.id Internet Source	<1 %
43	Submitted to Yeditepe University Student Paper	<1 %
44	Submitted to unimal Student Paper	<1 %
45	Submitted to INTI Universal Holdings SDM BHD Student Paper	<1 %
46	Submitted to UNIVERSITAS BUDI LUHUR Student Paper	<1 %
47	Submitted to Universitas Merdeka Malang Student Paper	<1 %
48	ejournal.unkhair.ac.id Internet Source	<1 %

49	Submitted to Georgetown University Student Paper	<1 %
50	abecindonesia.org Internet Source	<1 %
51	media.neliti.com Internet Source	<1 %
52	repository.uinsaizu.ac.id Internet Source	<1 %
53	Submitted to Coventry University Student Paper	<1 %
54	id.123dok.com Internet Source	<1 %
55	Submitted to itera Student Paper	<1 %
56	pt.scribd.com Internet Source	<1 %
57	repo.undiksha.ac.id Internet Source	<1 %
58	repository.futo.edu.ng Internet Source	<1 %
59	Submitted to University of Sydney Student Paper	<1 %
60	widuri.raharja.info Internet Source	<1 %
61	Submitted to STT PLN Student Paper	<1 %
62	Submitted to Universitas Maritim Raja Ali Haji Student Paper	<1 %

63	Submitted to University of Portsmouth Student Paper	<1 %
64	eprints.uns.ac.id Internet Source	<1 %
65	lup.lub.lu.se Internet Source	<1 %
66	ojs.amikomsolo.ac.id Internet Source	<1 %
67	openlibrarypublications.telkomuniversity.ac.id Internet Source	<1 %
68	repository.ar-raniry.ac.id Internet Source	<1 %
69	repository.pnj.ac.id Internet Source	<1 %
70	www.slideshare.net Internet Source	<1 %
71	Akram Kemal Dewantara. "Optimasi Pengambilan Keputusan dengan Neural Network: Menuju Era Keputusan Pintar", The Indonesian Journal of Computer Science, 2024 Publication	<1 %
72	Submitted to Fakultas Ekonomi Universitas Indonesia Student Paper	<1 %
73	Ni Putu Irene Pasca Nurahdika, Joko Sutopo. "Analisa Performa Convolution Neural Network Dalam Pengenalan Tulisan Tangan Aksara Lontara", INTECOMS: Journal of Information Technology and Computer Science, 2025	<1 %

74	Submitted to Sekolah Teknik Elektro & Informatika Student Paper	<1 %
75	Submitted to Telkom University Student Paper	<1 %
76	Submitted to Universitas Dian Nuswantoro Student Paper	<1 %
77	Submitted to Universitas Pendidikan Ganesha Student Paper	<1 %
78	Vilvilia Salim, Asrul Abdullah, Putri Yuli Utami. "Klasifikasi Citra Penyakit Tanaman pada Daun Paprika dengan Metode Transfer Learning Menggunakan DenseNet-201", The Indonesian Journal of Computer Science, 2024 Publication	<1 %
79	digibuo.uniovi.es Internet Source	<1 %
80	jurnal.stmik-amik-riau.ac.id Internet Source	<1 %
81	www.eecg.toronto.edu Internet Source	<1 %
82	epochai.org Internet Source	<1 %
83	ir.nust.na Internet Source	<1 %
84	repository.unhas.ac.id Internet Source	<1 %
85	vdocuments.net Internet Source	<1 %

86	www.neliti.com Internet Source	<1 %
87	garuda.kemdikbud.go.id Internet Source	<1 %
88	jurnal.pcr.ac.id Internet Source	<1 %
89	mljourney.com Internet Source	<1 %
90	penerbit.gunadarma.ac.id Internet Source	<1 %
91	Berliana Wahyu Nurlita, Sri Winarno, Adhitya Nugraha, Almas Najiib Imam Muttaqin et al. "Comparison of ArcFace and Dlib Performance in Face Recognition with Detection Using YOLOv8", INOVTEK Polbeng - Seri Informatika, 2024 Publication	<1 %
92	Submitted to Canadian University of Dubai Student Paper	<1 %
93	Jenny Rahma Hidayat, Jemakmum. "Implementasi Klasifikasi Citra Berbasis Tensorflow Untuk Mendeteksi Penyakit Tanaman Pada Aplikasi Agrosan", JURNAL FASILKOM, 2025 Publication	<1 %
94	Juanto Simangunsong, Nurmala Dewi Simanjuntak, Aprima Anugerah Matondang. "Penerapan Transfer Learning untuk Klasifikasi Citra Bunga Berbasis Convolutional Neural Network", Jurnal Minfo Polgan, 2025 Publication	<1 %

95	docobook.com Internet Source	<1 %
96	dokumen.tips Internet Source	<1 %
97	eprints.perbanas.ac.id Internet Source	<1 %
98	eprints.umm.ac.id Internet Source	<1 %
99	mafiadoc.com Internet Source	<1 %
100	repository.ubaya.ac.id Internet Source	<1 %
101	vdocuments.mx Internet Source	<1 %
102	Aang Alim Murtopo, Maulana Aditdya, Pingky Septiana Ananda, Gunawan Gunawan. "PENERAPAN COMPUTER VISION UNTUK MENDETEKSI KELENGKAPAN ATRIBUT SISWA MENGGUNAKAN METODE CNN", PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer, 2024 Publication	<1 %
103	Intania Dharma Hartarti, Intan Amelia Septiyani, Daniel Armando Gultom, Yayan Hendrian, Shynde Limar Kinanti. "Prediksi Harga Rumah di Boston Dengan Model Regresi Linear Menggunakan Python", RIGGS: Journal of Artificial Intelligence and Digital Business, 2025 Publication	<1 %

104	Muhammad Irfani, Siti Khomsah. "Analisis Sentimen Berbasis Aspek pada EDOM Pembelajaran Menggunakan Metode CNN dan Word2vec", Jurnal Sistem dan Teknologi Informasi (JustIN), 2024 Publication	<1 %
-----	---	------

105	classicalenthusiast.com Internet Source	<1 %
-----	--	------

106	de.scribd.com Internet Source	<1 %
-----	----------------------------------	------

107	discuss.tensorflow.org Internet Source	<1 %
-----	---	------

108	eprints.politeknikpu.ac.id Internet Source	<1 %
-----	---	------

109	eprints.utdi.ac.id Internet Source	<1 %
-----	---------------------------------------	------

110	es.scribd.com Internet Source	<1 %
-----	----------------------------------	------

111	id.scribd.com Internet Source	<1 %
-----	----------------------------------	------

112	idoc.pub Internet Source	<1 %
-----	-----------------------------	------

113	ioinformatic.org Internet Source	<1 %
-----	-------------------------------------	------

114	journal.mediapublikasi.id Internet Source	<1 %
-----	--	------

115	repository.ittelkom-pwt.ac.id Internet Source	<1 %
-----	--	------

repository.unej.ac.id

116	Internet Source	<1 %
117	vdocuments.site Internet Source	<1 %
118	www.kompasiana.com Internet Source	<1 %
119	www.scribd.com Internet Source	<1 %
120	Ananda Rizki Dani, Irma Handayani. "Classification of Yogyakarta Batik Motifs Using GLCM and CNN Methods", Jurnal Teknologi Terpadu, 2024 Publication	<1 %
121	Davito Rasendriya Rizqullah Putra, Reyhan Adi Saputra. "IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK MENDETEKSI PENGGUNAAN MASKER PADA GAMBAR", Jurnal Informatika dan Teknik Elektro Terapan, 2023 Publication	<1 %
122	Lusia Salmawati, Shinta Widya Puspita. "Factors Relating To Safety Riding Behaviors In Students Of Senior High School 5 Palu", Preventif : Jurnal Kesehatan Masyarakat, 2020 Publication	<1 %
123	Shankar Babu, Mahesh Babu Kota. "Synergies in Smart and Virtual Systems using computational intelligence", CRC Press, 2025 Publication	<1 %
124	Thimira Amaratunga. "Deep Learning on Windows", Springer Science and Business	<1 %

Media LLC, 2021

Publication

125

repository.uin-suska.ac.id

Internet Source

<1 %

126

repository.poliupg.ac.id

Internet Source

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72

PAGE 73

PAGE 74

PAGE 75

PAGE 76

PAGE 77

PAGE 78

PAGE 79

PAGE 80

PAGE 81

PAGE 82

PAGE 83

PAGE 84

PAGE 85

PAGE 86

PAGE 87

PAGE 88

PAGE 89
