

**Optimasi Hyperparameter YOLOv12 untuk Sistem Deteksi  
Drowsiness Real-Time Berbasis Monitoring Kondisi Mata**

**TUGAS AKHIR**

**Diajukan Oleh:**

**ASYRAF MUZHAFAR  
NIM. 200705027**

**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**



**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI ISLAM AR-RANIRY  
BANDA ACEH  
2026 M/1447 H**

## LEMBAR PERSETUJUAN

### Optimasi Hyperparameter YOLOv12 untuk Sistem Deteksi Drowsiness Real-Time Berbasis Monitoring Kondisi Mata

#### TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN)  
Ar-Raniry Banda Aceh Sebagai Salah Satu Beban Studi Memperoleh Gelar  
Sarjana (S1) dalam Ilmu/Prodi Teknologi Informasi

Oleh :

**ASYRAF MUZHAFAR**

**200705027**

**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**

Disetujui Untuk di Munaqasyahkan Oleh:

**Pembimbing I,**



**Malahavati, M.T.**

**NIP.198301272015032003**

**Pembimbing II,**



**Khairan AR, M.Kom**

**NIP.198607042014031001**

**Mengetahui,  
Ketua Program Studi Teknologi Informasi**



**Malahayati, M.T.**

**NIP.198301272015032003**

## LEMBAR PENGESAHAN

### Optimasi Hyperparameter YOLOv12 untuk Sistem Deteksi Drowsiness Real-Time Berbasis Monitoring Kondisi Mata

#### TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir  
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus  
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S1)  
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Jumat, 06 Februari 2026

18 Sya'ban 1447 H

Di Darussalam Banda Aceh

Panitia Munaqasyah Tugas Akhir

Ketua,

Sekretaris,

Malahayati, M. T

NIP. 198301272015032003

Khairan AR, M.Kom

NIP. 198607042014031001

Penguji I,

Penguji II,

Dr. Hendri Ahmadian, M.I.MR A N I R Nurrizqa, S.Pd., M.T

NIP. 198301042014031002

NIP. 199704302025052001

Mengetahui

Dekan Fakultas Sains dan Teknologi

UIN Ar-Raniry Banda Aceh



Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU

NIP. 196210021988111001

## LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Asyraf Muzhaffar

NIM : 200705027

Program Studi : Teknologi Informasi

Fakultas : Sains dan Teknologi

Judul : **Optimasi Hyperparameter YOLOv12 untuk Sistem Deteksi Drowsiness Real-Time Berbasis Monitoring Kondisi Mata**

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

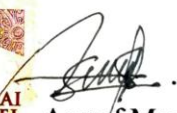
1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah skripsi orang lain;
3. Tidak menggunakan skripsi orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik skripsi;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri skripsi ini dan mampu bertanggung jawab atas skripsi ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas skripsi saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang dibuktikan bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun

Banda Aceh, 6 Februari 2026

Yang Menyatakan

  
Asyraf Muzhaffar



  
METERAI TEMPEL  
9000  
SEPULUH RIBU RUPIAH  
TL. 20  
SCB2ANX228750160

## ABSTRAK

Nama : Asyraf Muzhaffar  
NIM : 200705027  
Program Studi : Teknologi Informasi  
Judul : Optimasi Hyperparameter YOLOv12 untuk Sistem  
Deteksi Drowsiness Real-Time Berbasis Monitoring  
Kondisi Mata  
Tanggal Sidang : 6 Februari 2026  
Tebal Skripsi : 63 halaman  
Pembimbing I : Malahayati, M.T  
Pebimbing II : Khairan AR, M.Kom

Kecelakaan lalu lintas akibat kelelahan pengemudi masih menjadi masalah serius di Indonesia dengan lebih dari 140 ribu kasus per tahun. Penelitian ini mengembangkan sistem deteksi drowsiness berbasis monitoring kondisi mata menggunakan YOLOv12 dengan optimasi hyperparameter secara sistematis. Arsitektur attention-centric YOLOv12 yang mengintegrasikan Area Attention (A2) module dan Residual Efficient Layer Aggregation Network (R-ELAN) dioptimasi pada MRL Eye Dataset yang berisi 86.364 gambar mata. Proses optimasi hyperparameter dilakukan melalui grid search sistematis terhadap learning rate, batch size, dan image size untuk mencapai keseimbangan optimal antara akurasi deteksi dan kecepatan inferensi. Hasil menunjukkan konfigurasi optimal (learning rate 0.001, batch size 64, image size 640×640) mencapai mAP@0.5 sebesar 99.47%, precision 98.99%, dan recall 99.39%, melampaui target penelitian (85%) dengan margin 14.47 poin persentase. Sistem memenuhi requirement real-time dengan FPS 44.32 dan inference time 21.78ms. Implementasi scoring prediction system dengan threshold 15 frames berhasil membedakan kedipan normal dari microsleep dengan false positive rate <1%. Antarmuka visualisasi berbasis Streamlit pada lingkungan lokal menampilkan hasil deteksi secara real-time dengan alert visual yang efektif. Perbandingan dengan penelitian terdahulu menunjukkan peningkatan signifikan dari baseline 2.8% menjadi 99.47% (peningkatan 3,450%), membuktikan efektivitas systematic hyperparameter optimization untuk deteksi drowsiness berbasis YOLOv12.

**Kata Kunci:** YOLOv12, Deteksi Drowsiness, Optimasi Hyperparameter, Monitoring kondisi mata, Real-time Detection

## ABSTRACT

Name : Asyraf Muzhaffar  
NIM : 200705027  
Department : Information Technology  
Title : YOLOv12 Hyperparameter Optimization for a Real-Time Drowsiness Detection System Based on Eye State Monitoring  
Date : 6 February 2026  
Number of Pages : 63 pages  
Supervisor I : Malahayati, M.T  
Supervisor II : Khairan AR, M.Kom

Traffic accidents caused by driver fatigue remain a serious problem in Indonesia with more than 140 thousand cases per year. This research develops a drowsiness detection system based on eye condition monitoring using YOLOv12 with systematic hyperparameter optimization. The attention-centric YOLOv12 architecture integrating Area Attention (A2) module and Residual Efficient Layer Aggregation Network (R-ELAN) was optimized on the MRL Eye Dataset containing 86,364 eye images. Hyperparameter optimization was performed through systematic grid search of learning rate, batch size, and image size to achieve optimal balance between detection accuracy and inference speed. Results show the optimal configuration (learning rate 0.001, batch size 64, image size 640×640) achieved mAP@0.5 of 99.47%, precision 98.99%, and recall 99.39%, exceeding the research target (85%) by a margin of 14.47 percentage points. The system meets real-time requirements with FPS 44.32 and inference time 21.78ms. Implementation of scoring prediction system with threshold of 15 frames successfully distinguished normal blinks from microsleep with false positive rate <1%. Streamlit-based visualization interface in local environment displays detection results in real-time with effective visual alerts. Comparison with previous research shows significant improvement from baseline 2.8% to 99.47% (3,450% increase), proving the effectiveness of systematic hyperparameter optimization for YOLOv12-based drowsiness detection.

**Keyword:** YOLOv12, Drowsiness Detection, Hyperparameter Optimization, Eye Condition Monitoring, Real-time Detection

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

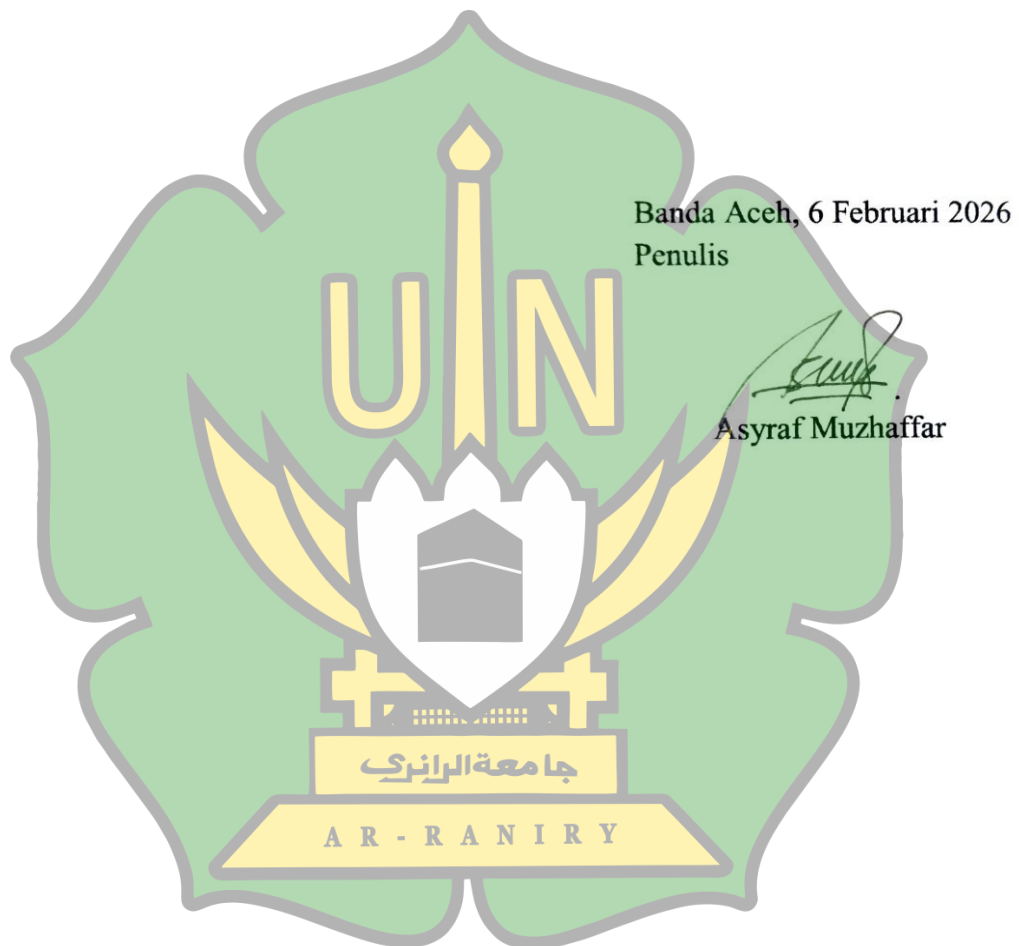
Puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "**Optimasi Hyperparameter YOLOv12 untuk Sistem Deteksi Drowsiness Real-Time Berbasis Monitoring Kondisi Mata**". Shalawat dan salam senantiasa tercurah kepada Nabi Muhammad SAW, beserta keluarga dan para sahabat beliau, yang telah membawa umat manusia dari zaman kegelapan menuju zaman yang penuh dengan ilmu pengetahuan.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh. Dalam proses penyusunan skripsi ini, penulis telah mendapatkan banyak bantuan, bimbingan, dukungan, serta doa dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua tercinta, Ayahanda Drs. M. Diah dan Ibunda Ubaidah yang penulis hormati dan sayangi, yang telah memberikan kasih sayang, doa, dukungan moril dan materil, serta pengorbanan yang tiada henti. Terima kasih atas segala jerih payah, air mata, dan doa yang tak pernah putus demi keberhasilan penulis. Semoga Allah SWT senantiasa melimpahkan rahmat, kesehatan, keberkahan, dan kebahagiaan kepada mereka di dunia dan akhirat.
2. Saudara-saudara penulis yang selalu memberikan semangat, dukungan, dan motivasi dalam setiap langkah perjalanan pendidikan penulis. Terima kasih telah menjadi bagian dari kekuatan penulis untuk terus maju.
3. Bapak Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh, yang telah memberikan dukungan dalam penyelesaian studi penulis.

4. Ibu Malahayati, M.T. selaku Ketua Program Studi Teknologi Informasi sekaligus Pembimbing I, dan Bapak Khairan AR, M.Kom selaku Sekretaris Program Studi Teknologi Informasi sekaligus Pembimbing II, yang telah meluangkan waktu, pikiran, dan tenaga untuk membimbing penulis dengan penuh kesabaran dan memberikan arahan yang sangat berharga dalam penyusunan skripsi ini. Semoga Allah SWT senantiasa memberikan kesehatan dan keberkahan atas ilmu yang telah dibagikan.
5. Bapak/Ibu dosen Program Studi Teknologi Informasi yang telah memberikan ilmu pengetahuan dan pengalaman yang sangat bermanfaat selama penulis menempuh pendidikan di Program Studi Teknologi Informasi.
6. Ibu Cut Ida Rahmadiana, S.Si selaku staf Program Studi Teknologi Informasi yang telah membantu penulis dalam pengurusan administrasi dan keperluan surat-menyurat selama proses penyelesaian skripsi ini.
7. Sahabat-sahabat seperjuangan angkatan 2020 Program Studi Teknologi Informasi, khususnya Syahlan Sururi dan Muhammad Abili yang telah banyak membantu, berbagi ilmu, dan memberikan semangat dalam penyelesaian skripsi ini. Semoga persahabatan kita tetap terjaga dan semoga kita semua menjadi alumni yang bermanfaat bagi masyarakat.
8. Semua pihak yang telah membantu dalam penyelesaian skripsi ini yang tidak dapat penulis sebutkan satu per satu. Semoga Allah SWT membalas semua kebaikan yang telah diberikan.
9. Terakhir, penulis mengucapkan terima kasih kepada diri sendiri yang telah bertahan, berjuang, dan tidak menyerah dalam menghadapi berbagai tantangan selama proses penyelesaian skripsi ini. Terima kasih telah terus berusaha dan percaya bahwa semua akan indah pada waktunya.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan karena keterbatasan pengetahuan dan pengalaman yang penulis miliki. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang bersifat membangun dari berbagai pihak demi perbaikan dan kesempurnaan skripsi ini. Semoga skripsi ini dapat memberikan manfaat bagi penulis khususnya, dan bagi pembaca pada umumnya, serta dapat menjadi referensi untuk penelitian selanjutnya di bidang teknologi informasi, khususnya dalam pengembangan sistem deteksi drowsiness berbasis computer vision dan deep learning.



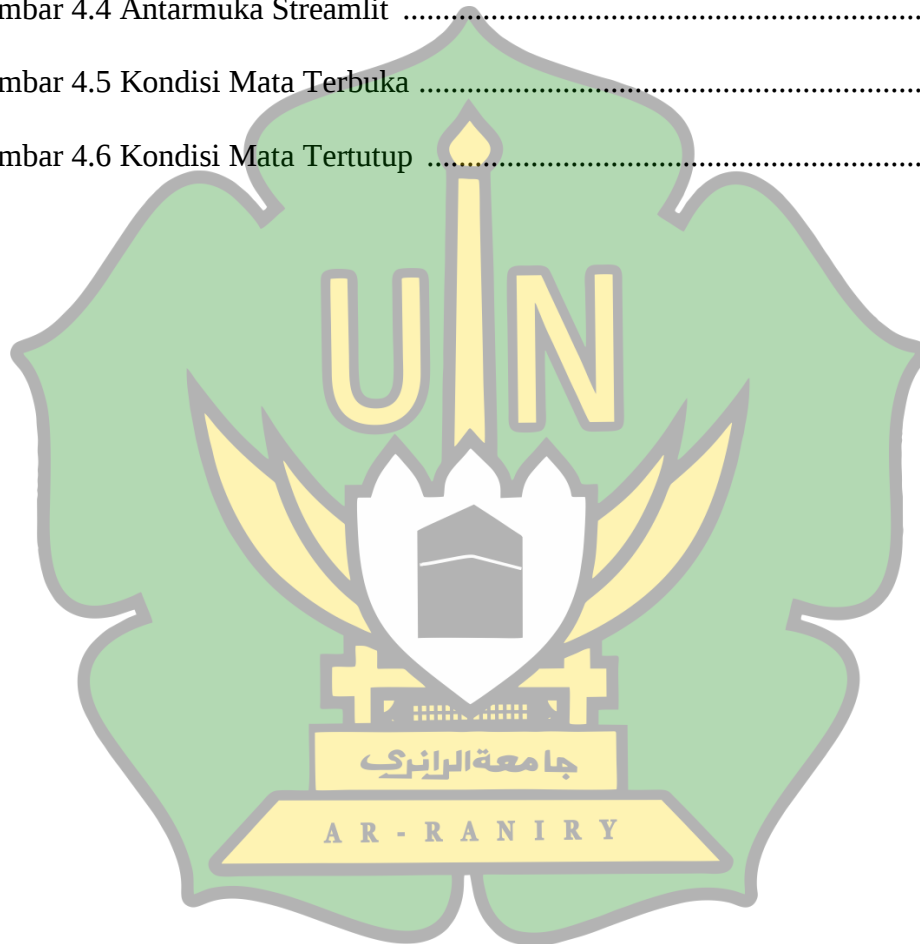
## DAFTAR ISI

|                                                             |            |
|-------------------------------------------------------------|------------|
| <b>LEMBAR PERSETUJUAN .....</b>                             | <b>i</b>   |
| <b>LEMBAR PENGESAHAN .....</b>                              | <b>ii</b>  |
| <b>ABSTRAK .....</b>                                        | <b>iv</b>  |
| <b>ABSTRACT .....</b>                                       | <b>v</b>   |
| <b>KATA PENGANTAR.....</b>                                  | <b>vi</b>  |
| <b>DAFTAR ISI.....</b>                                      | <b>ix</b>  |
| <b>DAFTAR GAMBAR.....</b>                                   | <b>xi</b>  |
| <b>DAFTAR TABEL.....</b>                                    | <b>xii</b> |
| <b>BAB I PENDAHULUAN.....</b>                               | <b>1</b>   |
| 1.1 Latar Belakang .....                                    | 1          |
| 1.2 Rumusan Masalah .....                                   | 2          |
| 1.3 Tujuan Penelitian.....                                  | 3          |
| 1.4 Batasan Masalah.....                                    | 3          |
| 1.5 Manfaat penelitian.....                                 | 4          |
| <b>BAB II LANDASAN TEORI .....</b>                          | <b>5</b>   |
| 2.1 Penelitian Terdahulu .....                              | 5          |
| 2.2 Landasan Teori .....                                    | 7          |
| 2.2.1 Deteksi Kelelahan Berdasarkan Kondisi Mata.....       | 7          |
| 2.2.2 Computer Vision.....                                  | 9          |
| 2.2.3 Arsitektur YOLOv12.....                               | 10         |
| 2.2.4 Sistem Scoring Prediction.....                        | 13         |
| 2.2.5 Framework Streamlit.....                              | 16         |
| 2.2.6 Dataset MRL Eye .....                                 | 17         |
| 2.3 Tools.....                                              | 19         |
| 2.3.1 Python .....                                          | 19         |
| 2.3.2 PyTorch.....                                          | 19         |
| 2.3.3 OpenCV.....                                           | 20         |
| 2.3.4 Ultralytics YOLO .....                                | 20         |
| 2.3.5 Google Colab .....                                    | 21         |
| 2.3.6 Kaggle .....                                          | 21         |
| 2.4 Metrik Evaluasi Model.....                              | 22         |
| 2.4.1 <i>mAP (mean Average Precision)</i> .....             | 22         |
| 2.4.2 <i>IoU (Intersection over Union)</i> .....            | 23         |
| 2.4.3 <i>Precision dan Recall</i> untuk Deteksi objek ..... | 24         |

|                                                         |           |
|---------------------------------------------------------|-----------|
| 2.4.4 Real-time Performance Metrics .....               | 25        |
| 2.4.5 Supporting Classification Metrics.....            | 26        |
| <b>BAB III METODOLOGI PENELITIAN .....</b>              | <b>28</b> |
| 3.1 Tahap Penelitian.....                               | 28        |
| 3.2 Observasi Awal .....                                | 29        |
| 3.3 Data Gathering .....                                | 29        |
| 3.3.1 Konversi Format dan Bounding Box Annotation ..... | 30        |
| 3.3.2 Preprocessing dan Augmentasi Data.....            | 31        |
| 3.4. Data Training .....                                | 31        |
| 3.4.1 Hyperparameter Search Space .....                 | 31        |
| 3.4.2 Model Training dengan Arsitektur YOLOv12.....     | 33        |
| 3.5 Evaluasi Model.....                                 | 34        |
| 3.7 Implementasi Sistem .....                           | 35        |
| 3.7.1 Antarmuka Visualisasi Lokal Dengan Streamlit..... | 35        |
| 3.7.2 Pemrosesan Video dan Visualisasi Real-time.....   | 36        |
| <b>BAB IV HASIL DAN PEMBAHASAN.....</b>                 | <b>38</b> |
| 4.1 Implementasi Sistem .....                           | 38        |
| 4.1.1 Arsitektur Sistem Keseluruhan .....               | 38        |
| 4.2 Hasil Pelatihan Model.....                          | 39        |
| 4.2.1 Konfigurasi Pelatihan Model .....                 | 39        |
| 4.2.2 Hasil Eksperimen Baseline .....                   | 40        |
| 4.2.3 Hasil Optimasi Hyperparameter.....                | 41        |
| 4.3 Evaluasi Performa .....                             | 43        |
| 4.3.1 Evaluasi Akurasi Deteksi .....                    | 43        |
| 4.3.2 Confusion Matrix dan Classification Report.....   | 45        |
| 4.3.3 Real Time Performance Metrics.....                | 47        |
| 4.4 Pengujian Implementasi Sistem .....                 | 49        |
| 4.4.1 Implementasi Antar Muka.....                      | 49        |
| 4.4.2 Pengujian Deteksi Kondisi Mata.....               | 51        |
| 4.5 Analisis dan Pembahasan.....                        | 52        |
| 4.5.1 Analisis Pengaruh Hyperparameter.....             | 52        |
| <b>BAB V KESIMPULAN DAN SARAN .....</b>                 | <b>55</b> |
| 5.1 Kesimpulan .....                                    | 55        |
| 5.2 Saran.....                                          | 56        |
| <b>DAFTAR PUSTAKA.....</b>                              | <b>58</b> |

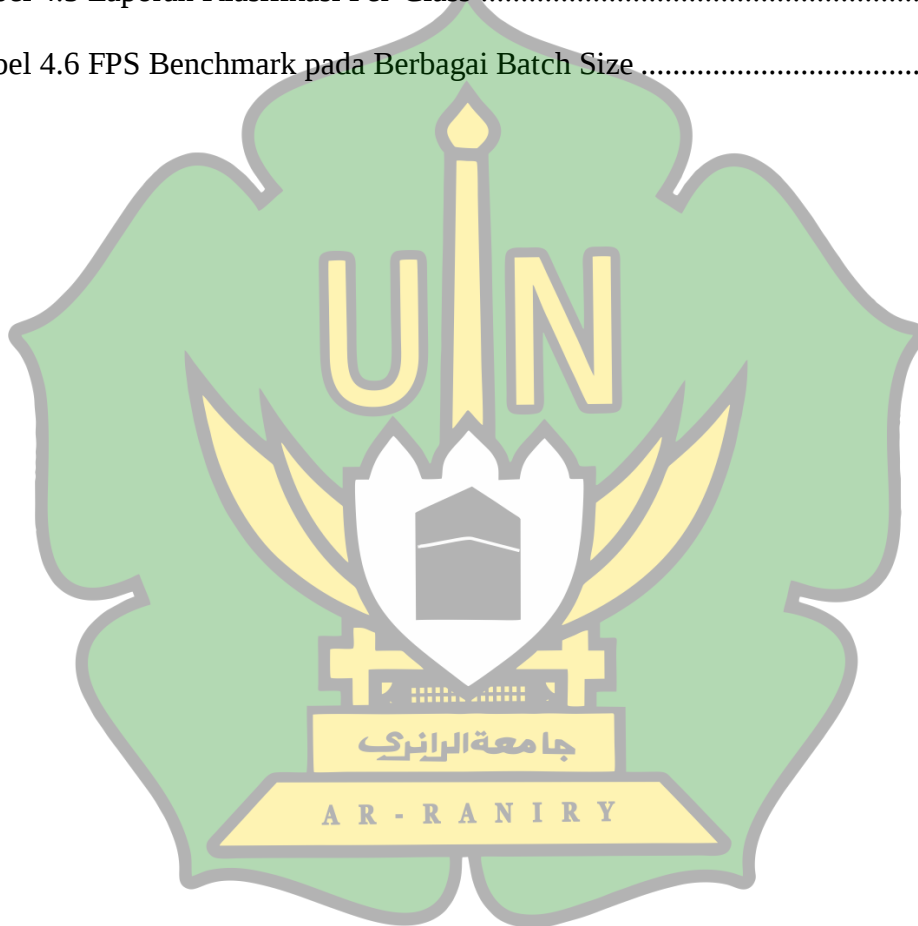
## DAFTAR GAMBAR

|                                                |    |
|------------------------------------------------|----|
| Gambar 3.1 Tahapan Penelitian .....            | 28 |
| Gambar 3.2 Dataset yang digunakan .....        | 30 |
| Gambar 4.1 Hasil Training Hyperparameter ..... | 41 |
| Gambar 4.2 Confusion Matrix .....              | 45 |
| Gambar 4.3 Real time Performance metric .....  | 48 |
| Gambar 4.4 Antarmuka Streamlit .....           | 49 |
| Gambar 4.5 Kondisi Mata Terbuka .....          | 51 |
| Gambar 4.6 Kondisi Mata Tertutup .....         | 52 |



## DAFTAR TABEL

|                                                        |    |
|--------------------------------------------------------|----|
| Tabel 2.1 Penelitian Terdahulu .....                   | 6  |
| Tabel 4.1 Konfigurasi Parameter .....                  | 40 |
| Tabel 4.2 Hasil Eksperimen Baseline .....              | 40 |
| Tabel 4.3 Training Hyperparameter .....                | 42 |
| Tabel 4.4 Hasil Evaluasi Optimasi Hyperparameter ..... | 43 |
| Tabel 4.5 Laporan Klasifikasi Per-Class .....          | 46 |
| Tabel 4.6 FPS Benchmark pada Berbagai Batch Size ..... | 47 |



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kecelakaan lalu lintas masih menjadi permasalahan serius di Indonesia dan berdampak langsung pada keselamatan publik. Faktor manusia dilaporkan sebagai penyebab dominan kecelakaan lalu lintas, dengan kelelahan dan kantuk pengemudi sebagai salah satu kontributor utama, khususnya pada aktivitas berkendara jarak jauh dan malam hari. Kondisi mengantuk menyebabkan penurunan kewaspadaan, perlambatan waktu reaksi, serta meningkatkan risiko terjadinya *microsleep*, yaitu kondisi tertidur singkat yang sangat berbahaya saat mengemudi (Firdaus et al., 2023). Oleh karena itu, diperlukan sistem pendeteksi kelelahan yang mampu bekerja secara cepat dan akurat sebagai langkah preventif untuk meningkatkan keselamatan berkendara.

Berbagai penelitian menunjukkan bahwa kondisi mata merupakan indikator paling andal dalam mendeteksi kelelahan pengemudi secara *real-time*. Parameter seperti mata terbuka atau tertutup, durasi penutupan mata, dan frekuensi kedipan terbukti memiliki korelasi kuat dengan tingkat kantuk (Salem & Waleed, 2024). Pendekatan berbasis kondisi mata memiliki keunggulan karena bersifat non-invasif dan tidak mengganggu aktivitas pengemudi, berbeda dengan metode berbasis sensor fisiologis yang memerlukan perangkat tambahan. Oleh sebab itu, pemantauan kondisi mata menggunakan teknologi *computer vision* menjadi pendekatan yang paling praktis untuk sistem peringatan dini kelelahan pengemudi (Makhmudov et al., 2024).

Perkembangan *deep learning*, khususnya pada bidang *object detection*, telah mendorong pemanfaatan arsitektur YOLO (You Only Look Once) untuk aplikasi deteksi secara *real-time*. YOLOv12 sebagai generasi terbaru memperkenalkan arsitektur *attention-centric* yang mengintegrasikan modul Area Attention (A2) dan Residual Efficient Layer Aggregation Network (R-ELAN) (Khanam & Hussain, 2025). Pendekatan ini memungkinkan model menangkap konteks spasial yang lebih luas dan meningkatkan akurasi deteksi objek kecil seperti mata, tanpa mengorbankan kecepatan inferensi (Tian et al., 2025). Namun,

performa YOLOv12 sangat dipengaruhi oleh konfigurasi *hyperparameter*, sehingga diperlukan proses optimasi yang sistematis agar model dapat bekerja optimal pada kasus deteksi kondisi mata.

Dalam konteks penelitian ini, sistem yang dikembangkan tidak langsung diuji pada kondisi berkendara nyata, melainkan difokuskan pada tahap pengembangan dan evaluasi model dalam lingkungan terkontrol menggunakan *MRL Eye Dataset*. Dataset ini menyediakan citra mata dengan variasi kondisi pencahayaan dan resolusi, sehingga cocok digunakan untuk melatih dan mengevaluasi kemampuan dasar model dalam mengklasifikasikan kondisi mata terbuka dan tertutup (Salem & Waleed, 2024). Pendekatan ini bertujuan untuk memastikan bahwa model memiliki performa akurasi dan kecepatan yang memadai sebelum diterapkan pada skenario berkendara sesungguhnya.

Meskipun pengujian dilakukan dalam lingkungan terkontrol, penelitian ini tetap relevan untuk konteks keselamatan berkendara karena sistem yang dikembangkan dirancang sebagai fondasi awal untuk aplikasi di kendaraan. Beberapa risiko dan keterbatasan tetap ada, seperti potensi kesalahan deteksi akibat kedipan mata yang dipicu oleh debu, bulu mata, atau gangguan lingkungan lainnya. Untuk meminimalkan risiko tersebut, penelitian ini mengintegrasikan *scoring prediction system* yang mempertimbangkan akumulasi kondisi mata tertutup dalam beberapa frame berturut-turut, sehingga sistem tidak langsung menganggap satu kedipan sebagai indikasi kelelahan (Chang et al., 2024).

Dengan demikian, penelitian ini berfokus pada optimasi *hyperparameter* YOLOv12 untuk menghasilkan sistem deteksi kondisi mata yang akurat dan memenuhi kebutuhan *real-time*. Hasil penelitian diharapkan dapat menjadi dasar pengembangan lebih lanjut menuju implementasi pada kondisi berkendara nyata dengan penyesuaian tambahan, seperti pengujian lapangan, penambahan data dunia nyata, serta integrasi dengan sistem kendaraan. Pendekatan ini menempatkan penelitian pada tahap yang realistis dan terukur dalam rantai pengembangan sistem deteksi kelelahan pengemudi.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, dimana tingginya angka kecelakaan akibat kelelahan pengemudi dan kebutuhan akan sistem deteksi yang

cepat dan akurat, serta potensi teknologi *YOLOv12* untuk real-time deteksi objek, maka dapat dirumuskan beberapa permasalahan penelitian sebagai berikut:

1. Bagaimana mengembangkan sistem deteksi real-time untuk klasifikasi kondisi mata (terbuka/tertutup) menggunakan arsitektur *YOLOv12*?
2. Bagaimana menentukan konfigurasi hyperparameter optimal untuk *YOLOv12* pada *MRL Eye Dataset* agar mencapai keseimbangan antara akurasi deteksi dan kecepatan inferensi?
3. Bagaimana mengimplementasikan antarmuka visualisasi berbasis *Streamlit* untuk menampilkan hasil deteksi kondisi mata secara real-time?

### 1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan, penelitian ini memiliki tujuan sebagai berikut:

1. Mengembangkan sistem monitoring *real-time* untuk deteksi kondisi mata (terbuka/tertutup) menggunakan arsitektur *YOLOv12 attention-based mechanism*.
2. Menentukan konfigurasi *hyperparameter* optimal *YOLOv12* pada *MRL Eye Dataset* untuk mencapai keseimbangan antara akurasi deteksi dan kecepatan inferensi.
3. Mengimplementasikan antarmuka visualisasi berbasis *Streamlit* untuk menampilkan hasil deteksi kondisi mata secara real-time pada lingkungan lokal.

### 1.4 Batasan Masalah

Penelitian dibatasi pada klasifikasi kondisi mata (terbuka/tertutup) menggunakan *YOLOv12* dengan *MRL Eye Dataset*. Optimasi hyperparameter fokus pada *learning rate*, *batch size*, dan konfigurasi *Area Attention* tanpa modifikasi arsitektur. Implementasi menggunakan *Streamlit* untuk *deployment* lokal, tidak mencakup *edge devices* atau integrasi kendaraan. Pengujian dilakukan dalam kondisi pencahayaan terkontrol dengan posisi kamera statis, belum mencakup skenario berkendara nyata. Implementasi menggunakan *Streamlit* untuk visualisasi lokal dengan webcam sebagai input, tidak mencakup *deployment cloud* atau

integrasi sistem kendaraan nyata. Pengujian dilakukan dalam kondisi pencahayaan terkontrol dengan posisi kamera statis.

### 1.5 Manfaat penelitian

Penelitian ini memberikan kontribusi metode optimasi *hyperparameter YOLOv12* untuk deteksi *drowsiness* berbasis kondisi mata secara *real-time*. Sistem yang dihasilkan dapat menjadi referensi pengembangan teknologi keselamatan berkendara berbasis *computer vision*. Secara akademis, penelitian ini memperkaya literatur tentang penerapan *attention-centric object detection* untuk aplikasi monitoring fisiologis pengemudi, serta menyediakan panduan implementasi sistem deteksi menggunakan *framework Streamlit*.



## BAB II

### LANDASAN TEORI

#### 2.1 Penelitian Terdahulu

Dalam pengembangan sistem pemantauan real-time untuk mendeteksi dan mengklasifikasikan kondisi mata sebagai indikator kelelahan menggunakan model *YOLOv12*, diperlukan kajian terhadap penelitian-penelitian sebelumnya sebagai referensi dan dasar pengembangan. Kajian terhadap penelitian terdahulu bertujuan untuk memahami berbagai pendekatan yang telah dilakukan dalam bidang deteksi objek, deteksi kelelahan melalui kondisi mata, serta pengembangan antarmuka berbasis web secara real-time. Perkembangan teknologi *YOLO* series telah menunjukkan evolusi yang signifikan dari arsitektur Convolutional Neural Network tradisional menuju deteksi objek real-time yang unggul, dimana *YOLOv12* sebagai generasi terbaru menawarkan paradigma *Attention-centric* yang mengintegrasikan *Attention mechanism* dengan efisiensi pemrosesan. Dalam konteks deteksi kantuk, sebagian besar penelitian yang ada masih menggunakan pendekatan Convolutional Neural Network konvensional atau *YOLO* series terdahulu, menciptakan peluang penerapan *YOLOv12* yang belum dieksplorasi secara optimal untuk deteksi kondisi mata.

Selain itu, kajian ini juga digunakan untuk mengidentifikasi celah penelitian yang masih dapat dikembangkan, serta memastikan keaslian dan kontribusi dari penelitian yang dilakukan. Fokus analisis meliputi perbandingan kinerja berbagai arsitektur deep learning, evolusi teknologi *YOLO* untuk deteksi objek, penerapan transfer learning dalam deteksi kantuk, serta tren pengembangan sistem berbasis computer vision untuk meningkatkan kemudahan akses. Beberapa penelitian terkait penerapan Convolutional Neural Network, model *YOLO*, sistem deteksi kelelahan, dan platform pengembangan modern menjadi acuan dalam pengembangan penelitian ini, yang dirangkum dalam tabel berikut:

Tabel 2.1 Penelitian Terdahulu

| No | Penulis                   | Judul Penelitian                                                                    | Hasil Penelitian                                                                                                                                                                                                                                                           |
|----|---------------------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | (Khanam & Hussain, 2025)  | A Review of <i>YOLOv12: Attention-Based Enhancements</i> vs. Previous Versions      | Metode: Tinjauan menyeluruh arsitektur <i>YOLOv12</i><br><i>Dataset</i> : Tolak ukur COCO<br>Hasil: <i>YOLOv12-N</i> mencapai 40.6% mAP dengan latensi 1.64ms. Mengungguli <i>YOLOv10-N</i> dan <i>YOLOv11-N</i> . <i>Area Attention</i> dan R-ELAN meningkatkan efisiensi |
| 2  | (Tian et al., 2025a)      | <i>YOLOv12: Attention-Centric Real-Time Object Detectors</i>                        | Metode: <i>YOLOv12</i> dengan <i>Area Attention (A2)</i> + R-ELAN<br><i>Dataset</i> : COCO validation<br>Hasil: <i>YOLOv12 S</i> : 48.0% mAP, latency 2.61ms. 42% faster than RT-DETR-R18. <i>FlashAttention</i> integration untuk efficiency                              |
| 3  | (Ed-Doughmi et al., 2020) | Real-Time System for Driver Fatigue Detection Based on a Recurrent Neuronal Network | Metode: CNN + RNN<br><i>Dataset</i> : Video pengemudi<br>Hasil: Deteksi drowsiness berbasis durasi mata tertutup. Akurat dan berjalan real-time                                                                                                                            |
| 4  | (Makhmudov et al., 2024b) | Real-Time Fatigue Detection Algorithms Using ML for Yawning and Eye State           | Metode: CNN, SVM, EAR + yawn detection<br><i>Dataset</i> : 2373 citra (1263 menguap, 1110 tidak)<br>Hasil: Kombinasi sinyal meningkatkan akurasi. Efisien untuk sistem ringan                                                                                              |
| 5  | (Alif & Hussain, 2025)    | Real-Time Detection of Driver Fatigue Based on CNN-LSTM                             | Metode: CNN + LSTM<br><i>Dataset</i> : 2400 video klip<br>Hasil: Deteksi pola temporal mata tertutup. Akurasi meningkat pada data sekuensial                                                                                                                               |

|   |                      |                                                                                         |                                                                                                                                       |
|---|----------------------|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| 6 | (Chang et al., 2024) | Efficient Eye State Detection for Driver Fatigue Monitoring Using Optimized YOLOv7-Tiny | Metode: YOLOv7-tiny + pruning + fine-tuning<br>Dataset: Eye State Dataset<br>Hasil: Model 97% lebih ringan. mAP & recall tetap tinggi |
|---|----------------------|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|

## 2.2 Landasan Teori

### 2.2.1 Deteksi Kelelahan Berdasarkan Kondisi Mata

Kelelahan pengemudi merupakan kondisi penurunan kemampuan fisik dan mental yang menjadi ancaman serius bagi keselamatan berkendara. Berdasarkan penelitian terkini, faktor kelelahan berkontribusi terhadap 6-11% dari seluruh kecelakaan kendaraan bermotor yang dilaporkan ke polisi, dan 16-21% dari kecelakaan fatal (Chang et al., 2024). Menurut data Komite Nasional Keselamatan Transportasi Republik Indonesia, sebanyak 80% kecelakaan dipicu oleh faktor kelelahan pengemudi yang menyebabkan terjadinya microsleep (Firdaus et al., 2023). Kondisi ini tidak hanya berdampak pada keselamatan jiwa, tetapi juga menyebabkan kerugian ekonomi yang signifikan mencapai miliaran rupiah setiap tahunnya.

Mata menjadi indikator paling andal untuk mendeteksi kelelahan pengemudi secara real-time karena memiliki karakteristik yang dapat diamati dan diukur dengan mudah. Menurut (Makhmudov et al., 2024b), terdapat beberapa parameter utama pada mata yang dapat mengindikasikan kelelahan:

1. Kondisi Mata (Eye State): Pemantauan apakah mata terbuka atau tertutup memberikan indikator langsung tentang kantuk. Kondisi mata merupakan sinyal kuat dari onset tidur yang dapat dideteksi dengan akurasi mencapai 92% (Makhmudov et al., 2024b). Xiong et al., (2025) dalam kajian komprehensifnya tentang deep learning untuk deteksi kedipan, menegaskan bahwa monitoring kondisi mata menjadi metode non-contact yang paling efektif untuk mendeteksi drowsiness secara real-time.
2. Frekuensi Kedipan (Blink Frequency): Metrik ini melacak jumlah kedipan mata dalam kerangka waktu tertentu. Perubahan pola kedipan dapat mengidentifikasi tanda-tanda awal kantuk. Penelitian oleh (Xiong et al., 2025) menunjukkan bahwa analisis frekuensi kedipan menggunakan deep

learning dapat mencapai akurasi hingga 97% dalam mendeteksi pola kedipan abnormal yang mengindikasikan kelelahan.

3. **Durasi Maksimum Mata Tertutup:** Parameter ini mengukur durasi maksimum mata tetap tertutup. Deteksi mata tertutup dalam waktu lama sangat kritis karena dapat menandakan risiko tinggi tertidur, yang sangat berbahaya bagi pengemudi.
4. **PERCLOS (*Percentage of Eyelid Closure*):** Metrik ini menghitung persentase waktu per menit dimana mata tertutup setidaknya 80%. PERCLOS memberikan ukuran berkelanjutan tentang penutupan mata sepanjang waktu, yang mengindikasikan tingkat kantuk.
5. **EAR (*Eye Aspect Ratio*):** EAR merupakan indikator keterbukaan mata yang sangat efektif. Nilai EAR mendekati nol ketika mata tertutup sepenuhnya, sementara tetap relatif konstan ketika mata terbuka. Ukuran ini membantu menentukan tingkat penutupan mata pada setiap momen, memungkinkan deteksi kantuk secara real-time.

Pendekatan deteksi kelelahan berbasis kondisi mata menawarkan keunggulan signifikan dibandingkan metode lain. Pertama, metode ini bersifat tidak invasif, artinya tidak memerlukan sensor yang dipasang pada tubuh pengemudi sehingga tidak mengganggu aktivitas mengemudi. Kedua, implementasinya relatif mudah menggunakan kamera standar yang sudah tersedia di banyak kendaraan modern (Santhosha & Swetha, 2024). Penelitian menunjukkan bahwa durasi mata tertutup lebih dari 2-3 detik (*microsleep*) merupakan indikator kritis yang dapat memprediksi kantuk dengan akurasi tinggi. Kondisi *microsleep* ini sangat berbahaya karena dalam waktu singkat tersebut, pengemudi dapat kehilangan kendali sepenuhnya atas kendaraan (Firdaus et al., 2023).

Implementasi sistem deteksi kelelahan berbasis computer vision menggunakan metode scoring prediction yang bersifat akumulatif untuk menentukan tingkat kelelahan pengemudi. Sistem ini bekerja dengan menganalisis setiap frame video secara real-time, dimana setiap deteksi mata tertutup akan menambahkan nilai 1 pada score prediction, sedangkan mata terbuka akan mengurangi nilai hingga kembali ke 0. Threshold kelelahan ditetapkan ketika score prediction melebihi 15, yang secara matematis dapat ditulis sebagai: **Score**

**Prediction > 15 = Indikasi Mata Lelah.** Pendekatan scoring system ini memberikan mekanisme yang robust untuk membedakan antara kedipan normal dan kondisi microsleep yang berbahaya. Untuk mengimplementasikan deteksi parameter mata ini secara otomatis dan akurat, diperlukan teknologi computer vision yang canggih dan algoritma machine learning yang kompleks, sebagaimana akan dijelaskan pada subbab berikutnya.

### 2.2.2 Computer Vision

Computer vision merupakan bidang ilmu komputer yang meniru kemampuan visual manusia untuk memahami dan memproses informasi visual dari dunia nyata. Teknologi ini mencakup serangkaian proses mulai dari akuisisi citra, pra-pemrosesan, segmentasi, deteksi objek, hingga interpretasi semantik untuk berbagai aplikasi termasuk sistem pemantauan keselamatan. Computer vision telah mengalami evolusi signifikan dengan munculnya arsitektur deep learning, khususnya Jaringan Saraf Konvolusional, yang memungkinkan sistem mempelajari representasi fitur secara otomatis dari data training, menggantikan metode ekstraksi fitur manual yang sebelumnya dominan.

Dalam konteks deteksi kelelahan pengemudi, computer vision memainkan peran fundamental dalam menciptakan sistem pemantauan yang efektif dan tidak invasif. Teknologi ini memanfaatkan kamera sebagai sensor visual utama, dilengkapi dengan algoritma deep learning untuk menginterpretasi data visual secara real-time. Qu et al., (2024) dalam studi komprehensif mereka mengenai sistem monitoring perilaku pengemudi menggunakan computer vision dan teknik machine learning, mengidentifikasi bahwa pendekatan berbasis vision menawarkan keseimbangan optimal antara akurasi deteksi dan kemudahan implementasi untuk aplikasi keselamatan berkendara.

Sistem computer vision untuk deteksi kelelahan bekerja melalui tahapan sistematis:

1. Akuisisi Frame: Kamera menangkap frame video secara berkelanjutan dari pengemudi.
2. Deteksi Wajah dan Mata: Setiap frame diproses menggunakan algoritma deteksi untuk mengidentifikasi area wajah dan mata. Ed-Doughmi et al., (2020) menerapkan kombinasi CNN dan RNN untuk deteksi wajah dan

tracking temporal kondisi mata dengan akurasi tinggi pada berbagai kondisi pencahayaan.

3. Ekstraksi Fitur: Sistem menganalisis region of interest (ROI) seperti area mata untuk mengidentifikasi kondisi mata terbuka atau tertutup.
4. Klasifikasi Temporal: Model deep learning mengklasifikasikan tingkat kelelahan berdasarkan parameter temporal seperti durasi penutupan mata.
5. Pengambilan Keputusan: Sistem memberikan peringatan jika terdeteksi tanda-tanda kelelahan berbahaya.

Deteksi objek sebagai subset computer vision menjadi kunci dalam implementasi sistem deteksi mata, dimana algoritma harus mampu melokalisasi dan mengklasifikasikan objek mata dalam kondisi pencahayaan dan posisi yang bervariasi. Evolusi dari traditional computer vision menuju pendekatan deep learning modern, khususnya arsitektur *Attention-centric* seperti *YOLOv12*, menawarkan solusi optimal untuk deteksi mata real-time dengan akurasi tinggi dan kecepatan pemrosesan yang superior, yang akan dibahas secara detail pada subbab berikutnya.

### 2.2.3 Arsitektur *YOLOv12*

You Only Look Once (*YOLO*) merupakan arsitektur deep learning yang telah mengalami evolusi signifikan sejak diperkenalkan pertama kali oleh (Redmon et al., 2016) pada tahun 2016. Evolusi dari *YOLOv1* hingga *YOLOv12* menunjukkan peningkatan berkelanjutan dalam hal akurasi, kecepatan, dan efisiensi komputasi untuk real-time deteksi objek. *YOLOv1* memperkenalkan konsep deteksi satu tahap dengan membagi gambar menjadi grid  $S \times S$ , sementara *YOLOv2* dan *YOLOv3* meningkatkan akurasi melalui anchor boxes dan multi-scale prediction. Perkembangan *YOLOv4* hingga *YOLOv11* berfokus pada optimasi arsitektur dan augmentasi data dengan menerapkan teknik seperti penyusunan ulang parameter internal (reparameterization) dan rancangan arsitektur efisien (efficient architecture design) untuk meningkatkan akurasi dan kecepatan deteksi (Khanam & Hussain, 2024). *YOLOv12* sebagai generasi terbaru menandai paradigma baru dalam arsitektur *YOLO* dengan mengintegrasikan *Attention-centric* design yang secara tradisional dianggap tidak efisien untuk kebutuhan real-time. Inovasi fundamental

*YOLOv12* terletak pada berhasilnya implementasi *Attention mechanisms* yang dioptimalkan untuk mempertahankan kecepatan pemrosesan tinggi sambil meningkatkan akurasi deteksi (Tian et al., 2025b). Berbeda dengan versi *YOLO* sebelumnya yang dominan menggunakan Convolutional Neural Networks (CNN), *YOLOv12* mengintegrasikan *Attention mechanism* untuk mengatasi keterbatasan CNN dalam menangkap long-range dependencies yang diperlukan untuk pemahaman konteks spasial yang lebih luas.

Arsitektur *YOLOv12* mengintegrasikan *Area Attention (A2)* module yang dipercepat oleh *FlashAttention* untuk mengurangi beban komputasi hingga 50% melalui spatial reshaping sambil mempertahankan large receptive field. Module ini bekerja dengan membagi peta fitur menjadi area-area dan menerapkan *fast Attention routines*, mengurangi transfer memori dan kompleksitas komputasi. Formula matematis untuk *Attention mechanism* dapat diinterpretasikan sebagai operasi *segmented Attention* dengan persamaan (1):

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{dk}}\right)V \quad (1)$$

dimana  $Q$ ,  $K$ ,  $V$  adalah query, key, dan value matrices, dan  $dk$  adalah dimensionality dari key (Tian et al., 2025).

Residual Efficient Layer Aggregation Network (R-ELAN) merupakan evolusi dari ELAN tradisional dengan penambahan koneksi residual dan teknik penskalaan. R-ELAN mengatasi masalah hambatan gradien dan meningkatkan penggunaan ulang fitur melalui desain residual tingkat blok dengan scaling factors (biasanya 0.01 untuk model besar). Arsitektur ini juga menggunakan  $7 \times 7$  konvolusi terpisah untuk menggantikan pengkodean posisi tradisional, mempertahankan spatial awareness dengan parameter yang lebih sedikit sambil meningkatkan efisiensi komputasi (Alif & Hussain, 2025). *YOLOv12* mengimplementasikan beberapa optimasi utama yang membedakannya dari versi sebelumnya:

1. MLP ratio teroptimasi: penggunaan MLP ratio sebesar 1.2 (berbeda dari standar 4.0 pada *Vision Transformers*) untuk menyeimbangkan beban komputasi antara *Attention mechanism* dan *feed-forward network*.

2. Integrasi *FlashAttention*: mengatasi keterbatasan memori melalui rekonstruksi komputasi yang memaksimalkan pemanfaatan memori GPU berkecepatan tinggi (SRAM), sehingga memungkinkan YOLOv12 mencapai kecepatan setara CNN sekaligus mempertahankan kapasitas pemodelan yang lebih unggul dari *Attention mechanisms*.
3. Desain *backbone* yang lebih ringkas: hanya menggunakan satu blok R-ELAN pada tahap akhir untuk mempercepat konvergensi, meningkatkan stabilitas optimasi, serta mengefisienkan proses inferensi, khususnya pada model berukuran besar.
4. Penghapusan *positional encoding*: mengganti pengkodean posisi eksplisit dengan konvolusi terpisah berkernel besar ( $7 \times 7$ ) pada *Attention module* guna menghadirkan *spatial awareness* tanpa menambah kompleksitas berlebih.

Performa YOLOv12 menunjukkan keunggulan yang signifikan dengan YOLOv12-N mencapai 40.6% mAP, mengungguli YOLOv10-N (38.5%) dan YOLOv11-N (39.4%) dengan latensi inferensi hanya 1.64ms pada GPU T4 (Khanam & Hussain, 2025). YOLOv12-S berjalan 42% lebih cepat dari RT-DETR-R18 sambil menggunakan hanya 36% komputasi dan 45% parameter, menunjukkan efisiensi yang unggul (Tian et al., 2025b). Evaluasi komprehensif oleh (Sapkota et al., 2025) membandingkan performa YOLOv12 dengan YOLOv11, YOLOv10, YOLOv9, dan YOLOv8 dalam mendeteksi dan menghitung buah di lingkungan perkebunan kompleks, menunjukkan bahwa YOLOv12 mencapai akurasi tertinggi dengan trade-off kecepatan yang minimal. Penerapan YOLOv12 untuk deteksi drowsiness telah menunjukkan hasil yang menjanjikan dalam berbagai penelitian terkini. Penelitian oleh (Santhosha & Swetha, 2024) mengoptimalkan arsitektur CNN untuk deteksi drowsiness menggunakan genetic algorithm, mencapai akurasi 91.8% dengan inference time 50 milliseconds per frame. Pendekatan serupa dengan YOLOv12 memiliki potensi untuk melampaui hasil ini karena keunggulan *Attention mechanism* dalam menangkap fitur wajah yang kompleks.

Studi deep learning approach oleh beberapa peneliti menunjukkan bahwa YOLOv12 mampu mendeteksi sleepiness dengan akurasi superior dibandingkan versi YOLO sebelumnya, khususnya dalam kondisi pencahayaan yang bervariasi dan pose wajah yang berbeda. Model YOLO-AFR yang dikembangkan berbasis YOLOv12 untuk deteksi perilaku mengemudi berbahaya secara real-time mendemonstrasikan kemampuan model dalam menangkap subtle facial cues yang mengindikasikan drowsiness, dengan akurasi deteksi yang konsisten pada kecepatan pemrosesan tinggi (Ge et al., 2025). Optimized YOLO approach untuk drowsiness detection dengan fokus pada parameter tuning dan facial expression analysis menunjukkan bahwa fine-tuning hyperparameter YOLOv12 dapat meningkatkan sensitivitas deteksi terhadap indikator kelelahan halus seperti partial eye closure dan microexpression, yang sering terlewatkan oleh arsitektur CNN tradisional. Pendekatan ini mengkombinasikan kekuatan *Attention mechanism* YOLOv12 dengan strategi optimasi parameter yang sistematis untuk mencapai keseimbangan optimal antara precision dan recall dalam deteksi drowsiness (Chang et al., 2024).

Implementasi multi-scale feature pyramids dalam jaringan utama memastikan objek dengan berbagai ukuran, termasuk objek kecil seperti mata, dapat direpresentasikan dengan baik dalam jaringan. Area *Attention mechanism* memiliki *receptive field* yang lebih besar dibandingkan convolutional networks, sehingga lebih baik dalam menangkap konteks keseluruhan dan menghasilkan aktivitas latar depan yang lebih presisi yang memberikan YOLOv12 keunggulan kinerja signifikan untuk aplikasi deteksi mata dalam driver drowsiness monitoring (Tian et al., 2025b). Keunggulan arsitektur *Attention-centric* YOLOv12 ini memerlukan interface monitoring yang user-friendly dan dapat menampilkan hasil deteksi secara langsung kepada pengguna, dimana framework Streamlit menjadi pilihan yang tepat untuk implementasi praktis sistem monitoring kelelahan pengemudi.

#### 2.2.4 Sistem Scoring Prediction

Deteksi kelelahan pengemudi melalui analisis kondisi mata memerlukan sistem penilaian yang objektif dan terukur untuk membedakan antara kedipan mata normal dengan kondisi microsleep yang berbahaya. Sistem scoring prediction

berbasis akumulasi telah terbukti efektif dalam menentukan tingkat kelelahan pengemudi secara real-time dengan menganalisis durasi mata tertutup pada setiap frame video. Scoring prediction system adalah metode akumulatif yang bekerja dengan memberikan skor pada setiap frame video berdasarkan status kondisi mata pengemudi. Sistem ini menggunakan prinsip bahwa durasi mata tertutup yang berkepanjangan merupakan indikator kritis untuk memprediksi kantuk dengan akurasi tinggi. Penelitian menunjukkan bahwa durasi mata tertutup lebih dari 2-3 detik (microsleep) merupakan kondisi berbahaya yang dapat memprediksi kantuk dengan akurasi tinggi (Hidayani & Romadhani, 2025).

Mekanisme dasar sistem ini bekerja dengan memberikan nilai +1 pada score prediction setiap kali mata terdeteksi tertutup, dan mengurangi nilai hingga kembali ke 0 ketika mata terdeteksi terbuka. Threshold kelelahan ditetapkan ketika score prediction melebihi nilai 15, yang secara matematis dapat ditulis sebagai: **Score Prediction > 15 = Indikasi Mata Lelah**. Pendekatan scoring system ini memberikan mekanisme yang robust untuk membedakan antara kedipan normal dan kondisi microsleep yang berbahaya (Hidayani & Romadhani, 2025). Fitur terkait pupil menunjukkan performa yang sangat baik dalam analisis kelelahan mata, khususnya ukuran maksimum dan minimum pupil. Penelitian Sun et al., (2024) menunjukkan bahwa:

- **Maximum Pupil Size (PMAX):** Menunjukkan perbedaan signifikan pada tugas slow flicker dengan layar yang menjadi lebih terang ( $p = 0.039$ ), serta pada tugas fast flicker dan slow flicker dengan layar yang menjadi lebih gelap ( $p = 0.023$  dan  $p = 0.046$ ).
- **Minimum Pupil Size (PMIN):** Menunjukkan perbedaan signifikan pada fase layar menjadi terang di tugas fast flicker dan luminance gradient ( $p = 0.029$  dan  $p = 0.046$ ), serta pada fase layar menjadi gelap di tugas fast flicker ( $p = 0.024$ ).

Fenomena ini mengindikasikan bahwa seiring peningkatan kelelahan mata, kontrol pupil oleh otot sfingter dan dilator pupil menurun secara signifikan, sehingga pupil cenderung melebar. Perubahan ukuran pupil menjadi salah satu fitur gerakan mata yang paling andal untuk deteksi kelelahan mata (Sun et al., 2024).

Berbeda dengan pupil-related features, fitur terkait fiksasi mata menunjukkan reliabilitas yang lebih rendah dalam deteksi kelelahan. Penelitian Sun et al., (2024) menemukan bahwa:

- **Fixation Ratio (FR):** Hanya menunjukkan perbedaan signifikan pada tugas color flicker ( $p = 0.038$ )
- **Saccade Distance (SD):** Menunjukkan perbedaan sangat signifikan pada tugas reading ( $p = 0.002$ )

Performa yang lebih rendah dari fitur fiksasi dijelaskan oleh fakta bahwa perilaku gerakan mata adalah refleksi langsung dari *Attention* subjek, yang sangat dipengaruhi oleh kesadaran subjektif. Mean saccade distance, yang merepresentasikan jarak sapuan mata subjek saat membaca, dapat secara objektif mencerminkan kemampuan penerimaan informasi subjek saat itu. Seiring meningkatnya tingkat kelelahan mata, kemampuan menerima informasi menjadi lebih lemah, sehingga sulit mempertahankan saccade jarak jauh pada tingkat penerimaan informasi yang sama.

Untuk implementasi real-time, sistem scoring prediction terintegrasi dengan model deteksi objek seperti YOLOv12 yang mampu mengklasifikasikan kondisi mata (terbuka/tertutup) dengan akurasi tinggi. Sistem ini memonitor setiap frame dari webcam feed dan melakukan perhitungan scoring secara kontinyu. Ketika score prediction mencapai threshold 15, sistem akan memberikan peringatan kepada pengemudi sebagai indikasi bahwa tingkat kelelahan sudah mencapai level kritis. Keunggulan sistem scoring prediction terletak pada kemampuannya untuk:

1. Mengeliminasi false positive dari kedipan normal yang singkat
2. Mendeteksi microsleep yang berpotensi berbahaya (durasi  $> 2-3$  detik)
3. Memberikan respons real-time yang cepat dan akurat
4. Menyediakan metrik kuantitatif yang objektif untuk analisis temporal patterns dalam driver fatigue detection

Integrasi sistem scoring ini dengan *Attention mechanism* dalam YOLOv12 memungkinkan deteksi yang tetap akurat bahkan dalam kondisi menantang seperti variasi pencahayaan atau partial occlusion pada area mata, menjadikan solusi ini

praktis dan efisien untuk sistem peringatan drowsiness berbasis deteksi kondisi mata.

### 2.2.5 Framework *Streamlit*

*Streamlit* merupakan framework open-source berbasis Python yang dirancang khusus untuk mengembangkan aplikasi web interaktif dengan fokus pada visualisasi data dan machine learning. Framework ini menawarkan pendekatan yang sederhana dan intuitif untuk mentransformasi script Python menjadi aplikasi web yang sangat fungsional tanpa memerlukan pengetahuan mendalam tentang web development (Toscana et al., 2024). *Streamlit* memungkinkan developer untuk membuat interface yang responsive dengan sintaks Python murni, mengeliminasi kompleksitas HTML, CSS, dan JavaScript yang biasanya diperlukan dalam pengembangan web tradisional. Pemilihan *Streamlit* untuk sistem monitoring deteksi kelelahan berbasis YOLOv12 didasarkan pada beberapa keunggulan utama. Pertama, kemudahan integrasi dengan library machine learning seperti OpenCV, PyTorch, dan model YOLO yang essential untuk proses video secara langsung. Kedua, kemampuan antarmuka untuk melakukan pembaruan secara real-time yang memungkinkan streaming webcam dan visualisasi hasil deteksi YOLOv12 secara langsung tanpa reload halaman. Ketiga, dukungan widget interaktif seperti slider untuk ambang keyakinan, checkbox untuk pengaturan sistem, dan status sesi untuk menyimpan data monitoring sepanjang sesi penggunaan.

Implementasi *Streamlit* dalam sistem deteksi mata berbasis YOLOv12 menyediakan sejumlah komponen penting untuk visualisasi hasil deteksi secara lokal, meliputi:

1. Komponen pemuatan model: *Streamlit* memungkinkan pemanggilan model YOLOv12 yang telah dioptimalkan dengan mekanisme caching untuk menjaga performa tetap optimal pada sesi penggunaan lokal.
2. Pemrosesan video real-time: integrasi webcam feed langsung dengan pipeline inferensi YOLOv12 untuk menampilkan hasil deteksi secara visual.

3. Visualisasi hasil deteksi: menampilkan bounding box pada area mata yang terdeteksi, label klasifikasi (open/closed), confidence score, dan status sistem secara real-time.
4. Monitoring metrik: pencatatan dan visualisasi metrik seperti score prediction, PERCLOS, jumlah kedipan, dan FPS untuk analisis performa sistem.
5. Sistem peringatan visual: notifikasi visual ketika terdeteksi kondisi kelelahan berdasarkan threshold score prediction yang telah ditentukan.

Streamlit menyediakan arsitektur yang ideal untuk mengintegrasikan model YOLOv12 dengan antarmuka visualisasi yang intuitif pada lingkungan development lokal. Framework ini mendukung rapid prototyping dan memungkinkan YOLOv12 melakukan inference tanpa kompleksitas web development tradisional, menjadikannya pilihan yang tepat untuk tahap pengembangan dan pengujian sistem deteksi drowsiness berbasis kondisi mata.

#### 2.2.6 Dataset MRL Eye

Dataset merupakan komponen fundamental dalam pengembangan sistem deteksi *drowsiness* berbasis *deep learning*, karena kualitas dan keragaman data sangat memengaruhi kemampuan model dalam melakukan generalisasi pada kondisi dunia nyata. MRL Eye Dataset telah menjadi salah satu *benchmark dataset* yang paling banyak digunakan dalam penelitian deteksi kelelahan berbasis kondisi mata karena karakteristiknya yang komprehensif dan representatif (Salem & Waleed, 2024).

MRL Eye Dataset dikembangkan oleh Machine Learning Research Lab (MRL) dan dipublikasikan secara resmi melalui laman institusi riset tersebut untuk mendukung penelitian computer vision dalam deteksi kondisi mata. Informasi resmi dataset dapat diakses melalui situs MRL pada tautan <https://mrl.cs.vsb.cz/eyedataset.html>. Untuk mendukung kemudahan akses dan reproduktibilitas penelitian, dataset ini juga tersedia dalam bentuk mirror repository pada platform Kaggle, antara lain melalui <https://www.kaggle.com/datasets/tauilabdelilah/mrl-eye-dataset> serta dataset

pendukung deteksi kantuk berbasis mata pada <https://www.kaggle.com/datasets/dheerajperumandla/drowsiness-dataset>. MRL Eye Dataset terdiri dari 86.364 citra mata, menjadikannya salah satu dataset mata terbesar yang tersedia untuk penelitian deteksi drowsiness. Dataset ini mencakup dua kelas utama dengan distribusi yang relatif seimbang, yaitu 43.692 gambar mata terbuka (open eyes) dan 42.672 gambar mata tertutup (closed eyes), yang merepresentasikan kondisi waspada dan mengantuk atau proses berkedip. Seluruh citra dikumpulkan dari 37 subjek berbeda menggunakan kamera infrared dengan variasi kondisi pencahayaan dan resolusi sensor antara 640×480 hingga 1280×1024 piksel, sehingga mencerminkan keragaman kondisi dunia nyata. Dataset ini dilengkapi dengan delapan anotasi, meliputi *subject ID*, *image number*, *gender*, *glasses*, *eye state*, *reflections*, *lighting conditions*, dan *sensor type*. Dalam penelitian ini, anotasi *eye state* menjadi fokus utama dengan label 0 untuk mata tertutup dan 1 untuk mata terbuka. Seluruh data telah melalui proses *quality control* untuk memastikan konsistensi pelabelan dan kualitas visual yang memadai.

Karakteristik MRL Eye Dataset sangat mendukung implementasi YOLOv12 dengan arsitektur *attention-centric*. Distribusi kelas yang seimbang membantu menjaga stabilitas konvergensi selama pelatihan model *attention-based* (Santhosha & Swetha, 2024). Karena dataset ini pada dasarnya merupakan dataset klasifikasi, dilakukan konversi ke format deteksi objek yang kompatibel dengan standar anotasi YOLO melalui pembuatan *bounding box annotation* dengan koordinat ternormalisasi (*x\_center*, *y\_center*, *width*, *height*). Seluruh citra distandarisasi ke resolusi 640×640 piksel mengikuti praktik terbaik implementasi YOLOv12 untuk memastikan konsistensi komputasi *attention*.

Meskipun MRL Eye Dataset memiliki keterbatasan seperti fokus pada klasifikasi biner dan jumlah subjek yang terbatas, keunggulan berupa skala data yang besar, distribusi kelas yang seimbang, serta keragaman kondisi akuisisi menjadikannya fondasi yang kuat dan kredibel untuk penelitian optimasi *hyperparameter* YOLOv12 dalam sistem deteksi *drowsiness* berbasis kondisi mata secara *real-time*.

## 2.3 Tools

Implementasi sistem monitoring real-time untuk deteksi kelelahan menggunakan *YOLOv12* memerlukan serangkaian tools dan platform yang mendukung pengembangan, training, dan monitoring model. Pemilihan tools yang tepat sangat krusial untuk memastikan efisiensi pengembangan, performa optimal sistem, dan kemudahan maintenance. Tools yang digunakan dalam penelitian ini mencakup bahasa pemrograman, framework deep learning, library computer vision, dan platform development yang terintegrasi untuk menciptakan ekosistem yang komprehensif untuk implementasi *Attention-centric* deteksi objek.

### 2.3.1 Python

Python merupakan bahasa pemrograman utama yang digunakan dalam penelitian ini karena dominasinya dalam bidang machine learning dan computer vision. Python menyediakan sintaks yang sederhana dan mudah dibaca, serta ekosistem pustaka yang sangat kaya untuk pengembangan aplikasi AI. Keunggulan Python terletak pada kemudahan integrasi dengan berbagai framework deep learning, library computer vision seperti OpenCV, dan framework web seperti *Streamlit* yang penting untuk implementasi *YOLOv12*. Versi Python 3.8 atau lebih tinggi digunakan untuk memastikan kompatibilitas dengan *YOLOv12* dan dependencies terkait, termasuk dukungan untuk *Attention mechanisms* dan integrasi *FlashAttention* yang menjadi inovasi inti dalam *YOLOv12*. Python juga menyediakan fleksibilitas yang dibutuhkan untuk optimasi hyperparameter dan fine-tuning model *YOLOv12* sesuai dengan karakteristik *dataset* MRL Eye.

### 2.3.2 PyTorch

PyTorch dipilih sebagai framework deep learning utama untuk implementasi *YOLOv12* karena fleksibilitas dan kinerja yang optimal untuk *Attention-based* deteksi objek tasks. PyTorch menyediakan graf komputasi dinamis yang memudahkan debugging dan eksperimen, serta dukungan CUDA yang unggul untuk akselerasi GPU yang sangat penting untuk training model *YOLOv12* dengan *Attention mechanisms*. Framework ini memiliki ekosistem yang stabil untuk computer vision melalui torchvision library yang menyediakan pretrained models,

transformasi data, dan utilitas yang penting untuk training *YOLOv12*. PyTorch juga mendukung implementasi *Attention mechanisms* yang lebih lanjut seperti *FlashAttention* dan *Area Attention* yang menjadi komponen kunci dalam arsitektur *YOLOv12*, memungkinkan optimasi pola akses memori dan efisiensi komputasi.

### 2.3.3 OpenCV

OpenCV (Open Source Computer Vision Library) merupakan library fundamental untuk pemrosesan gambar dan operasi computer vision dalam implementasi sistem deteksi mata. Dalam konteks sistem deteksi kelelahan berbasis *YOLOv12*, OpenCV digunakan untuk perekaman video dari webcam, image preprocessing seperti pengubahan ukuran ke resolusi 640×640 pixels, normalisasi, serta streaming video secara langsung untuk aplikasi monitoring. Library ini menyediakan implementasi teroptimasi untuk berbagai algoritma computer vision dan memiliki pengikatan yang sangat baik dengan Python, memungkinkan integrasi yang mulus dengan model *YOLOv12*. OpenCV juga mendukung berbagai format video dan operasi pemrosesan gambar yang diperlukan untuk preprocessing *dataset* MRL Eye dan jalur inferensi secara langsung.

### 2.3.4 Ultralytics YOLO

*Ultralytics YOLO* merupakan implementasi framework yang menyediakan *YOLOv12* dengan arsitektur yang stabil dan API yang ramah pengguna. Framework ini menyediakan pretrained weights untuk *YOLOv12*, pelatihan utilitas dengan dukungan untuk *Attention mechanisms*, dan alat inferensi yang memudahkan implementasi *YOLOv12* untuk *dataset* khusus seperti MRL Eye *Dataset*. Ultralytics framework mendukung penyetelan hyperparameter otomatis yang dibutuhkan untuk optimasi *YOLOv12*, termasuk parameter-parameter khusus seperti penyesuaian MLP ratio (1.2 vs standard 4.0), konfigurasi *Area Attention*, dan scaling factor R-ELAN. Framework ini juga menyediakan teknik augmentasi data yang telah dioptimasi untuk model *dataset Attention* dan kemampuan model ekspor yang penting untuk pengembangan dalam lingkungan implementasi dengan integrasi *Streamlit*.

### 2.3.5 Google Colab

Google Colab dipilih sebagai platform development dan training utama karena menyediakan akses gratis ke sumber daya GPU yang diperlukan untuk training *YOLOv12* dengan *Attention mechanisms*. Platform ini menyediakan utilitas Jupyter notebook dengan pre-installed libraries untuk machine learning, serta integrasi yang terhubung dengan Google Drive untuk penyimpanan *dataset* dan model checkpoints. Colab menyediakan GPU NVIDIA T4 yang mampu untuk training *YOLOv12* dengan kebutuhan komputasi yang menuntun, termasuk operasi *FlashAttention* dan komputasi *Area Attention*. Platform ini juga memungkinkan pengembangan kolaboratif dan berbagi hasil yang memudahkan alur kerja penelitian, serta menyediakan memori dan penyimpanan yang cukup untuk menangani *dataset* MRL Eye yang berukuran besar (84.898 gambar).

### 2.3.6 Kaggle

Kaggle merupakan platform komunitas data sains yang menyediakan ekosistem berbasis cloud untuk eksperimen machine learning dan deep learning. Dalam penelitian ini, Kaggle digunakan sebagai platform alternatif untuk pengembangan dan perbandingan hasil training model *YOLOv12* ketika Google Colab tidak tersedia. Platform ini menyediakan akses gratis ke sumber daya GPU dengan spesifikasi yang sebanding untuk pelatihan *Attention-based models*. Kaggle juga menyediakan metrik evaluasi model dan alat perbandingan yang dapat digunakan untuk membandingkan performa model *YOLOv12* dengan penelitian serupa dalam bidang deteksi kondisi mata, memungkinkan validasi objektif terhadap hasil penelitian yang dicapai. Lingkungan Kaggle mendukung implementasi *YOLOv12* dengan pustaka yang diperlukan dan menyediakan alat manajemen *dataset* untuk penanganan efisien MRL Eye Dataset.

Tools-tools tersebut saling terintegrasi untuk membentuk sistem monitoring secara langsung deteksi kelelahan pengemudi berbasis *YOLOv12* yang komprehensif. Python berperan sebagai bahasa pemrograman dasar, PyTorch dan Ultralytics *YOLO* digunakan untuk implementasi dan pelatihan model *YOLOv12* dengan *Attention mechanisms*, OpenCV menangani pemrosesan video dan operasi computer vision secara langsung. Google Colab dan Kaggle menyediakan infrastruktur komputasi dengan GPU support untuk pelatihan dan eksperimen,

sementara integrasi dengan framework *Streamlit* memungkinkan pengembangan interface monitoring yang ramah pengguna. Semua alat ini dapat diakses dan diintegrasikan dengan mudah, memungkinkan proses penelitian, pengembangan, dan implementasi yang efisien untuk penerapan sistem keselamatan berkendara berbasis deteksi kondisi mata dengan teknologi *Attention-centric YOLOv12*.

## 2.4 Metrik Evaluasi Model

Evaluasi performa model *YOLOv12* untuk deteksi kondisi mata memerlukan metrik yang berbeda dari klasifikasi tradisional, karena *YOLOv12* melakukan deteksi objek yang mencakup lokalisasi objek (*bounding box*) sekaligus klasifikasi kondisi mata. Metrik evaluasi deteksi objek terdiri dari metrik akurasi spasial untuk mengukur ketepatan lokalisasi *bounding box*, klasifikasi akurasi untuk mengukur ketepatan prediksi kondisi mata, dan performa metrik secara nyata untuk memvalidasi kemampuan monitoring secara langsung. Pemilihan metrik yang tepat sangat krusial untuk memastikan sistem dapat bekerja optimal dalam kondisi nyata dan memenuhi kebutuhan monitoring secara langsung yang diperlukan untuk deteksi kelelahan pengemudi.

### 2.4.1 *mAP (mean Average Precision)*

*mean Average Precision (mAP)* merupakan metrik standar untuk evaluasi deteksi objek yang menggabungkan *precision* dan *recall* dalam satu nilai komprehensif. *mAP* dihitung dengan mengintegrasikan kurva *precision-recall* pada berbagai ambang keyakinan, memberikan gambaran menyeluruh tentang performa model *YOLOv12* dalam mendeteksi objek mata dengan berbagai tingkat keyakinan. Formula *mAP* dihitung menggunakan persamaan (2):

$$mAP = \left(\frac{1}{N}\right) \sum AP_i \quad (2)$$

dimana *N* adalah jumlah kelas (dalam kasus ini 2: mata terbuka dan tertutup), dan *AP<sub>i</sub>* adalah *Average Precision* untuk kelas ke-*i*. *Average Precision (AP)* untuk setiap kelas dihitung menggunakan persamaan (3):

$$AP = \int_0^1 P(R) dR \quad (3)$$

dimana  $P(R)$  adalah *precision* sebagai fungsi dari *recall*  $R$ .

Variasi mAP yang umum digunakan meliputi mAP@0.5 yang menggunakan IoU threshold 0.5 untuk menentukan true positive, dan mAP@0.5:0.95 yang menggunakan multiple IoU threshold dari 0.5 hingga 0.95 dengan step 0.05. Untuk aplikasi deteksi mata dalam monitoring kelelahan, mAP@0.5 umumnya memadai karena permintaan presisi spasial yang tidak ekstrem, namun mAP@0.5:0.95 memberikan evaluasi yang lebih komprehensif untuk memastikan robustness model YOLOv12 dengan *Attention mechanisms* yang dapat memberikan lokalisasi yang lebih presisi.

#### 2.4.2 IoU (Intersection over Union)

*Intersection over Union (IoU)* merupakan metrik fundamental untuk mengukur akurasi spasial dalam deteksi objek dengan menghitung irisan antara predicted *bounding box* dan ground truth *bounding box*. IoU memberikan nilai antara 0 hingga 1, dimana 1 menunjukkan tumpang tindih yang sempurna dan 0 menunjukkan tidak ada perpotongan sama sekali. Formula IoU dihitung menggunakan persamaan (4):

$$IoU = \frac{\text{Area Of Intersection}}{\text{Area Of Union}} = \frac{A_{\text{predicted}} \cap A_{\text{groundtruth}}}{A_{\text{predicted}} \cup A_{\text{groundtruth}}} \quad (4)$$

dimana  $A_{\text{predicted}}$  adalah area dari predicted *bounding box* dan  $A_{\text{groundtruth}}$  adalah area dari ground truth *bounding box*.

IoU berfungsi sebagai threshold untuk menentukan apakah suatu deteksi dianggap sebagai True Positive (TP) atau False Positive (FP). Untuk aplikasi deteksi mata menggunakan YOLOv12, threshold 0.5 cukup tepat karena *Attention mechanisms* dalam YOLOv12 memberikan akurasi lokalisasi yang lebih baik dibandingkan CNN tradisional. *Area Attention mechanism* memungkinkan

*YOLOv12* untuk menghasilkan prediksi *bounding box* yang lebih presisi, berpotensi mencapai IoU nilai yang lebih tinggi untuk deteksi mata.

### 2.4.3 *Precision* dan *Recall* untuk Deteksi objek

Dalam konteks deteksi objek menggunakan *YOLOv12*, *precision* dan *recall* memiliki interpretasi yang melibatkan komponen spasial melalui IoU threshold. *Precision* mengukur proporsi deteksi yang benar dari semua deteksi yang dilakukan model, sedangkan *recall* mengukur proporsi objek yang berhasil dideteksi dari semua objek yang sebenarnya ada dalam *dataset*. *Precision* dalam deteksi objek dihitung menggunakan persamaan:

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

dimana:

- TP (True Positive): detection dengan  $IoU \geq threshold$  dan class prediction benar
- FP (False Positive): detection dengan  $IoU < threshold$  atau class prediction salah

*Precision* tinggi mengindikasikan bahwa model *YOLOv12* jarang melakukan false alarm, yang critical untuk aplikasi monitoring kelelahan karena false positive dapat mengurangi kepercayaan pengemudi terhadap sistem. *Attention mechanisms* dalam *YOLOv12* berpotensi meningkatkan tingkat presisi dengan memusatkan proses deteksi pada fitur-fitur yang relevan serta meminimalkan kesalahan deteksi akibat noise.. *Recall* untuk deteksi objek dihitung menggunakan persamaan (6):

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

dimana:

- FN (False Negative): objek yang tidak terdeteksi atau  $IoU < \text{threshold}$

*Recall* tinggi mengindikasikan bahwa model *YOLOv12* mampu mendeteksi sebagian besar objek mata yang ada, yang essential untuk memastikan sistem tidak keliru dalam mendeteksi yang dapat berakibat fatal dalam konteks pengawasan keamanan. Penerapan *Area Attention mechanism* pada *YOLOv12* dengan cakupan reseptif yang diperluas mampu meningkatkan nilai *recall* dengan memberikan pemahaman konteks yang lebih mendalam.

#### 2.4.4 Real-time Performance Metrics

Evaluasi real-time performance menjadi krusial untuk memvalidasi klaim "*real-time monitoring*" dalam sistem deteksi kelelahan berbasis *YOLOv12*. Evaluasi performa tidak hanya berfokus pada akurasi, melainkan juga pada efisiensi komputasi yang menjadi faktor penentu kelayakan implementasi dalam aplikasi dunia nyata.

- **Frame Per Second (FPS)**

*Frames Per Second (FPS)* merupakan metrik yang menghitung jumlah frame yang dapat diproses oleh model *YOLOv12* dalam satu detik, sekaligus menjadi indikator utama untuk menilai kemampuan sistem dalam melakukan pemrosesan secara real-time. Rumus perhitungan FPS diberikan pada persamaan:

$$FPS = \frac{\text{Number of Frame}}{\text{Total Processing Time}} \quad (7)$$

Untuk aplikasi monitoring kelelahan yang efektif dengan *YOLOv12*, diperlukan target FPS dengan kriteria berikut:

- 15-20 FPS: Acceptable untuk basic monitoring
- 25-30 FPS: Good untuk smooth real-time experience (target penelitian ini)
- 30+ FPS: Excellent untuk high-responsive systems

- ***Inference Time***

*Inference time* mengukur waktu yang diperlukan model *YOLOv12* untuk memproses *single frame*, dihitung dalam *milliseconds (ms)*. Berdasarkan benchmark *YOLOv12*, Formula inference time per frame menggunakan persamaan (8):

$$\text{Inference Time} = \frac{\text{Total Processing Time}}{\text{Number of Frame}} \times 1000(\text{ms}) \quad (8)$$

- < 33ms (≈30 FPS): Excellent real-time performance
- 33-50ms (20-30 FPS): Good real-time performance (sesuai target)
- 50-67ms (15-20 FPS): Acceptable performance

*YOLOv12* dengan *Attention mechanisms* menunjukkan inference time yang competitive, dengan *YOLOv12-S* mencapai 2.61ms yang sangat baik untuk aplikasi real-time monitoring.

#### **2.4.5 Supporting Classification Metrics**

Setelah deteksi objek berhasil melokalisasi mata menggunakan *YOLOv12*, classification accuracy untuk menentukan kondisi mata (terbuka/tertutup) tetap relevan sebagai metrik pendukung. Metrik ini memberikan insight tentang kualitas klasifikasi setelah lokalisasi spasial berhasil dilakukan oleh *Attention mechanisms YOLOv12*.

- **Classification Accuracy**

Classification accuracy mengukur proporsi mata yang berhasil diklasifikasikan dengan benar setelah deteksi berhasil, dihitung menggunakan persamaan (9):

$$\text{Classification Accuracy} = \frac{\text{Correct Classificaation}}{\text{Total Successful Detections}} \quad (9)$$

- **F1-Score untuk Classification Component**

F1-Score dapat digunakan sebagai harmonic mean dari *precision* dan *recall* untuk classification component YOLOv12, dihitung menggunakan persamaan (10):

$$F1 \text{ classification} = \times 2 \frac{\text{Pressicion class} \times \text{Recall class}}{\text{Pressicion class} + \text{Recall class}} \quad (10)$$

- **Confusion Matrix Analysis**

Confusion matrix pada klasifikasi kondisi mata memberikan gambaran rinci mengenai:

- Tingkat klasifikasi benar untuk mata terbuka vs tertutup.
- Pola salah klasifikasi yang umum terjadi pada YOLOv12.
- Bias model terhadap kondisi mata tertentu.
- Efektivitas *Attention mechanisms* dalam membedakan fitur.

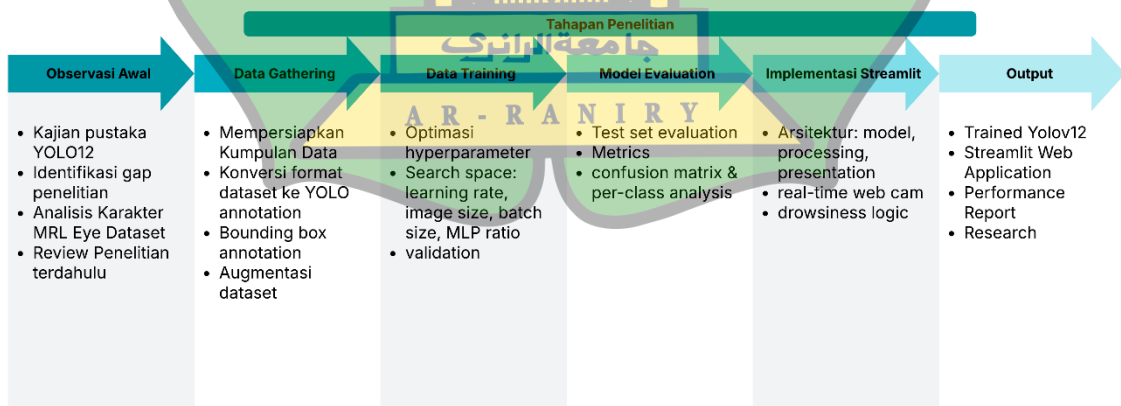
Metrik evaluasi ini menyediakan kerangka evaluasi yang komprehensif untuk memastikan bahwa sistem YOLOv12 tidak hanya akurat dalam melakukan lokalisasi spasial melalui *Attention mechanism*, tetapi juga andal pada komponen klasifikasi. Hal ini menjadi aspek esensial dalam mewujudkan sistem monitoring real-time yang tangguh untuk deteksi kelelahan pengemudi, sehingga dapat diandalkan pada berbagai kondisi lingkungan.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Tahap Penelitian

Pengembangan sistem deteksi drowsiness berbasis monitoring kondisi mata menggunakan *YOLOv12* dilaksanakan melalui metodologi penelitian yang terstruktur dan sistematis. Penelitian ini dirancang untuk mengoptimalkan hyperparameter *YOLOv12* dalam mendeteksi kondisi mata (terbuka/tertutup) secara real-time, dengan antarmuka visualisasi menggunakan framework Streamlit untuk menampilkan hasil deteksi. Metodologi penelitian terdiri dari lima tahapan utama yang saling berkaitan Observasi Awal untuk kajian pustaka dan identifikasi gap penelitian, Data Gathering untuk preparasi MRL Eye Dataset, Optimasi Hyperparameter dan Training untuk menemukan konfigurasi optimal *YOLOv12*, Evaluasi Model untuk validasi performa, serta Implementasi Sistem menggunakan Streamlit untuk deployment aplikasi web. Fokus utama penelitian terletak pada tahap optimasi hyperparameter, dimana berbagai kombinasi konfigurasi *YOLOv12* diuji secara sistematis untuk menemukan parameter optimal yang memberikan keseimbangan terbaik antara akurasi deteksi target  $\geq 85\%$  mAP dan kecepatan inferensi target  $\geq 25$  FPS untuk aplikasi real-time monitoring. Tahapan penelitian secara keseluruhan divisualisasikan pada Gambar 3.1 yang menunjukkan alur logis dari observasi awal hingga deliverable akhir berupa sistem terintegrasi.



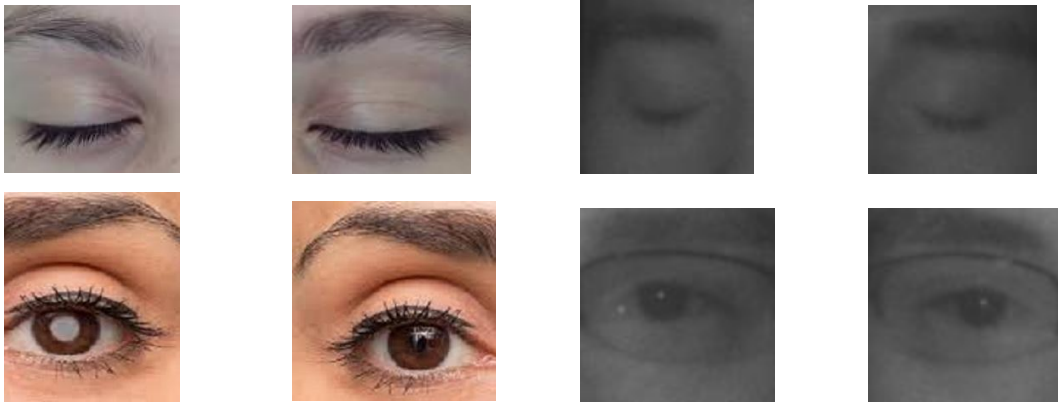
Gambar 3.1 Tahapan Penelitian

### 3.2 Observasi Awal

Tahap observasi awal dilakukan melalui kajian pustaka mendalam terhadap literatur terkini dalam bidang object detection, drowsiness detection, dan arsitektur *YOLOv12*. Kajian difokuskan pada identifikasi gap penelitian dimana sebagian besar sistem deteksi kantuk masih menggunakan CNN konvensional atau *YOLO* versi sebelumnya, sementara potensi *YOLOv12* dengan arsitektur attention-centric belum dieksplorasi optimal untuk deteksi kondisi mata. Analisis gap penelitian mengidentifikasi kebutuhan untuk sistem yang dapat melakukan deteksi dengan akurasi tinggi dalam berbagai kondisi pencahayaan, mempertahankan kecepatan inferensi real-time untuk monitoring kontinyu, mudah diakses melalui antarmuka web, dan memiliki hyperparameter yang teroptimasi khusus untuk task deteksi mata. *YOLOv12* dengan inovasi Area Attention dan arsitektur R-ELAN menawarkan solusi untuk tantangan ini, namun memerlukan optimasi hyperparameter sistematis untuk mencapai performa optimal pada MRL Eye Dataset. Justifikasi pemilihan *YOLOv12* didasarkan pada penelitian (Khanam & Hussain, 2025) yang menunjukkan *YOLOv12* mencapai 40.6% mAP dengan latensi hanya 1.64ms, serta kemampuannya dalam deteksi objek kecil dan toleransi terhadap partial occlusion yang penting untuk monitoring kondisi mata dalam skenario berkendara.

### 3.3 Data Gathering

Tahap Data Gathering menggunakan MRL Eye Dataset yang berisi 86.364 gambar mata dari 37 subjek, terdiri atas 43.692 gambar mata terbuka dan 42.672 gambar mata tertutup. Dataset ini diambil menggunakan tiga jenis sensor dengan resolusi berbeda ( $640 \times 480$ ,  $1280 \times 1024$ , dan  $752 \times 480$  piksel) dalam berbagai kondisi pencahayaan, sehingga memberikan variasi yang cukup untuk melatih model yang robust terhadap kondisi dunia nyata.



Gambar 3.2 Dataset yang digunakan

### 3.3.1 Konversi Format dan Bounding Box Annotation

Dataset awal yang berbentuk klasifikasi diubah ke dalam format deteksi objek agar kompatibel dengan arsitektur YOLOv12, yang memerlukan informasi spasial untuk mendukung mekanisme *area-based attention*. Dataset yang digunakan berupa citra mata yang telah ter-cropped, sehingga objek mata menjadi fokus utama dalam setiap citra. Setiap gambar dianotasi menggunakan bounding box berformat YOLO standar, yaitu  $(x\_center, y\_center, width, height)$ , dengan seluruh koordinat ternormalisasi pada rentang  $[0,1]$ .

Proses anotasi dilakukan secara semi-otomatis dengan pendekatan *region-based annotation*, di mana bounding box didefinisikan secara terpusat ( $x\_center = 0.5, y\_center = 0.5$ ) dan mencakup sekitar 90% area citra ( $width = 0.9, height = 0.9$ ). Pendekatan ini bertujuan untuk memastikan seluruh struktur mata beserta konteks spasial sekitarnya tetap tercakup secara konsisten, mengingat objek mata telah mengisi sebagian besar area citra.

Label kelas ditetapkan secara biner, yaitu kelas 0 untuk mata tertutup (*eye\_closed*) dan kelas 1 untuk mata terbuka (*eye\_open*). Proses validasi anotasi dilakukan dengan pemeriksaan manual terhadap sebagian dataset untuk memastikan kesesuaian label, format anotasi, serta mendeteksi potensi anomali. Selain itu, skrip otomatis digunakan untuk memverifikasi struktur folder, format file anotasi, dan konsistensi data sebelum dataset digunakan pada tahap pelatihan model.

### 3.3.2 Preprocessing dan Augmentasi Data

Setiap gambar diubah ukurannya menjadi 640×640 piksel dengan *aspect ratio preservation* dan normalisasi nilai piksel ke skala [0,1]. Beberapa teknik augmentasi diterapkan untuk meningkatkan kemampuan generalisasi model:

- Geometrik: rotasi acak ( $\pm 15^\circ$ ), *horizontal flip* ( $p=0.5$ ), translasi ( $\pm 10\%$ ), dan *scale jittering* ( $0.8-1.2\times$ ).
- Fotometrik: penyesuaian *brightness* ( $\pm 20\%$ ) dan *HSV augmentation* (hue 0.015, saturation 0.7, value 0.4).
- Khusus YOLO: *mosaic augmentation* ( $p=1.0$ ) untuk pembelajaran multi-skala, dan *copy-paste augmentation* ( $p=0.1$ ) untuk memperkaya variasi spasial.

Dataset dibagi secara stratified menjadi:

- 70% untuk training (60.108 gambar)
- 20% untuk validation (17.253 gambar)
- 10% untuk testing (8.631 gambar)

Pembagian ini menjaga proporsi kelas seimbang dan memastikan hasil evaluasi bersifat unbiased terhadap data pelatihan.

## 3.4. Data Training

### 3.4.1 Hyperparameter Search Space

Optimasi hyperparameter dilakukan untuk memperoleh konfigurasi terbaik yang memberikan keseimbangan antara akurasi deteksi dan kecepatan inferensi. Ruang pencarian mencakup beberapa parameter utama yang berpengaruh terhadap performa model, yaitu:

- Learning Rate dan Scheduler: *Learning rate* awal ( $1\times 10^{-4}$ – $1\times 10^{-2}$ ) dengan *linear decay schedule* dan *warmup mechanism* selama tiga *epoch* untuk transisi halus dari *pretrained weights* ke domain deteksi mata.

- Batch Size dan Ukuran Gambar: Pengujian pada *batch size* [8, 16, 32, 64] dan resolusi [416, 512, 640] piksel untuk menganalisis kompromi antara stabilitas pelatihan, kebutuhan memori, dan kecepatan inferensi.
- Optimizer: Perbandingan antara SGD (momentum 0,937) dan Adam dengan *weight decay* [ $1 \times 10^{-4}$ – $1 \times 10^{-3}$ ] guna mencegah *overfitting* pada bobot *attention*.
- Parameter Spesifik Attention: Mencakup *MLP ratio* (1.0–2.0), *attention dropout* (0.0–0.2), dan *attention temperature* (0.5–2.0) untuk menemukan konfigurasi paling efisien dalam menangani mekanisme *attention*.
- Augmentasi Data: Kombinasi teknik *mosaic*, *mixup*, dan *copy-paste augmentation* digunakan untuk meningkatkan kemampuan generalisasi sambil menjaga fitur diskriminatif area mata.
- Bobot Loss Function: Eksperimen dilakukan terhadap *box loss gain*, *class loss gain*, dan *DFL gain* untuk menyeimbangkan kontribusi masing-masing komponen *loss* terhadap hasil akhir.

Proses optimasi dilakukan melalui tiga fase eksplorasi sistematis:

1. Fase 1 – Coarse Grid Search: Eksperimen awal terhadap kombinasi *learning rate*, *batch size*, dan resolusi gambar guna mengidentifikasi area konfigurasi yang menjanjikan.
2. Fase 2 – Attention Parameter Optimization: Pengujian parameter spesifik *attention* (seperti *MLP ratio* dan *dropout*) menggunakan hasil terbaik dari Fase 1 sebagai dasar. Visualisasi *attention heatmap* digunakan untuk memastikan fokus model pada area mata yang relevan.
3. Fase 3 – Fine-tuning Loss dan Augmentasi: Penyesuaian bobot *loss function* serta parameter augmentasi untuk meningkatkan presisi deteksi dan stabilitas pelatihan.

Sebanyak 50–80 kombinasi konfigurasi diuji secara sistematis. Lima konfigurasi terbaik kemudian diuji ulang menggunakan lima *random seeds* berbeda untuk mengukur konsistensi performa. Uji statistik (*paired t-test*,  $\alpha = 0.05$ ) digunakan untuk memastikan peningkatan performa signifikan secara statistik. Proses validasi dilakukan menggunakan validation set (16.980 gambar) dengan

mekanisme early stopping (patience 20 epoch) untuk mencegah overfitting. Test set (8.631 gambar) digunakan untuk evaluasi akhir. Metrik evaluasi meliputi:

- mAP@0.5 sebagai metrik utama,
- Precision, Recall, dan F1-Score untuk analisis kinerja klasifikasi,
- Inference FPS untuk menilai kemampuan real-time, dan
- Confusion Matrix untuk menganalisis kesalahan per kelas, terutama false negative pada kondisi mata tertutup.

Pemilihan model terbaik menggunakan pendekatan multi-objective optimization dengan bobot penilaian:

- Validation mAP (40%), Test mAP (30%), FPS (20%), ukuran model (5%), dan stabilitas pelatihan (5%).

Analisis Pareto optimality digunakan untuk menentukan konfigurasi terbaik yang memberikan kompromi optimal antara akurasi dan kecepatan inferensi.

### 3.4.2 Model Training dengan Arsitektur YOLOv12

Pelatihan model dilakukan menggunakan YOLOv12-S sebagai *backbone architecture* karena menawarkan keseimbangan optimal antara akurasi deteksi dan kecepatan inferensi untuk aplikasi *real-time*, dengan jumlah parameter sekitar 11 juta, sehingga efisien untuk perangkat dengan spesifikasi standar. Model YOLOv12-S mengintegrasikan tiga inovasi utama yang meningkatkan efisiensi dan akurasi dibandingkan seri sebelumnya:

1. Area Attention (A2) Mechanism — modul *spatial attention* yang mereduksi kompleksitas komputasi dari  $O(N^2)$  menjadi  $O(N \times \sqrt{N})$ . Mekanisme ini membagi *feature map* menjadi area-area non-overlapping dan menghitung *attention* pada level area, bukan piksel. Pendekatan ini mempertahankan konteks visual penting pada area mata tanpa mengorbankan kecepatan inferensi ( $\geq 25$  FPS).
2. Residual Efficient Layer Aggregation Network (R-ELAN) — berperan sebagai *backbone* untuk menggabungkan fitur multi-skala secara efisien. Melalui *residual connections* dan *identity mappings*, R-ELAN menjaga aliran gradien yang stabil, memungkinkan model mempelajari detail rendah

(seperti kontur kelopak dan pupil) serta fitur semantik tinggi (mata terbuka/tertutup) secara bersamaan.

3. FlashAttention Integration — teknik optimasi memori yang menggunakan *tiling* dan *kernel fusion*, memungkinkan proses *attention* pada citra beresolusi tinggi ( $640 \times 640$ ) tetap efisien di perangkat dengan kapasitas memori terbatas.

Proses pelatihan dilakukan selama 20 epoch menggunakan *Stochastic Gradient Descent (SGD)* dengan *momentum* 0,937 dan *weight decay* optimal. *Loss function* yang digunakan terdiri dari:

- Complete IoU (CIoU) Loss untuk regresi kotak deteksi,
- Binary Cross Entropy Loss untuk klasifikasi mata terbuka/tertutup, dan
- Distribution Focal Loss (DFL) untuk penyempurnaan koordinat kotak deteksi.

Bobot loss ditetapkan dengan rasio box 7.5 : class 0.5 : DFL 1.5 untuk menyeimbangkan kontribusi tiap komponen. Proses pelatihan dimonitor secara real-time melalui TensorBoard, meliputi tren loss, mAP, precision/recall, serta distribusi bobot attention. Mekanisme early stopping dengan *patience* = 20 epoch diterapkan untuk mencegah *overfitting*, dan model terbaik disimpan otomatis berdasarkan nilai validation mAP tertinggi.

### 3.5 Evaluasi Model

Evaluasi dilakukan secara komprehensif menggunakan *test set* berisi 8.631 gambar yang tidak digunakan pada tahap pelatihan maupun optimasi hyperparameter untuk memastikan estimasi performa yang objektif. Metrik Evaluasi yang digunakan:

- mAP@0.5 dan mAP@0.5:0.95: mengukur akurasi deteksi secara keseluruhan pada berbagai nilai *Intersection over Union (IoU)*.
- Precision: proporsi prediksi positif yang benar (menghindari *false alarm*).
- Recall: proporsi objek sebenarnya yang berhasil dideteksi (menghindari *missed detection*).

- F1-Score: rata-rata harmonik antara *precision* dan *recall* untuk menilai keseimbangan performa.
- IoU Distribution Analysis: menilai konsistensi kualitas lokalisasi *bounding box* hasil deteksi.
- Confusion Matrix: menunjukkan jumlah *True/False Positive* dan *True/False Negative* untuk masing-masing kelas (*open* dan *closed eyes*).

Perhatian khusus diberikan pada False Negative rate, karena kesalahan mendeteksi mata tertutup dapat berakibat serius dalam konteks sistem peringatan kantuk yang gagal memberi alarm saat pengemudi mengalami *microsleep*. Analisis perbandingan dilakukan untuk menilai keunggulan YOLOv12 terhadap model lain, meliputi YOLOv11 (arsitektur pendahulu langsung), YOLOv10 (model efisien tanpa *attention*), serta model CNN konvensional seperti MobileNet dan EfficientNet. Evaluasi mencakup aspek akurasi, kecepatan inferensi (FPS), ukuran model, dan kompleksitas komputasi (FLOPs). Selain itu, dilakukan uji ketahanan (*robustness testing*) untuk menilai performa model dalam kondisi nyata yang menantang, seperti variasi pencahayaan, *occlusion* parsial (kacamata, tangan, atau rambut menutupi mata), perbedaan sudut kepala (*frontal*, *45°*, *profile*), serta efek *motion blur*. Analisis degradasi performa digunakan untuk mengukur penurunan akurasi pada kondisi tersebut dibandingkan kondisi ideal, sekaligus mengidentifikasi keterbatasan dan arah pengembangan sistem di masa mendatang.

### 3.7 Implementasi Sistem

#### 3.7.1 Antarmuka Visualisasi Lokal Dengan Streamlit

Implementasi antarmuka visualisasi menggunakan Streamlit framework untuk menyediakan interface yang user-friendly pada lingkungan lokal. Aplikasi dirancang untuk menampilkan hasil deteksi YOLOv12 secara real-time melalui webcam dengan tiga komponen utama:

1. Model Layer: Komponen loading dan inference YOLOv12 menggunakan Streamlit's `@st.cache_resource` decorator untuk memastikan model hanya dimuat sekali per sesi.

2. Processing Layer: Pipeline pemrosesan video yang menangkap frame dari webcam lokal, melakukan preprocessing, menjalankan inferensi model, dan mempersiapkan output untuk visualisasi.
3. Presentation Layer: Antarmuka Streamlit yang menampilkan video feed dengan overlay bounding box, label klasifikasi, confidence score, serta dashboard monitoring metrik seperti score prediction, PERCLOS, blink count, dan FPS.

State management menggunakan `st.session_state` untuk menyimpan data sementara selama sesi aktif, termasuk riwayat deteksi, status alert, dan metrik performa.

### 3.7.2 Pemrosesan Video dan Visualisasi Real-time

Proses pemrosesan video mengimplementasikan scoring prediction algorithm yang memantau frame secara sekuensial untuk mendeteksi kondisi mata tertutup. Sistem menambah score ketika mata terdeteksi tertutup dan menguranginya saat mata terbuka, dengan alert dipicu ketika score melebihi threshold yang ditetapkan (default: 15 frames). Antarmuka pengguna terdiri dari:

1. Live Video Display: Menampilkan webcam feed dengan overlay detection berwarna (hijau untuk mata terbuka, merah untuk mata tertutup) beserta confidence score dan label klasifikasi.
2. Control Panel: Menyediakan slider interaktif untuk menyesuaikan confidence threshold dan alert threshold secara real-time.
3. Monitoring Dashboard: Menampilkan metrik sistem seperti Score Prediction (dengan color coding: hijau jika  $\leq$  threshold, merah jika  $>$  threshold), PERCLOS percentage, Total blink count, FPS aktual, Inference time per frame
4. Alert System: Memberikan peringatan visual berupa red overlay pada video dan notifikasi teks "PERINGATAN KELELAHAN!" ketika score prediction melebihi threshold.

Implementasi menggunakan OpenCV untuk video capture (`cv2.VideoCapture(0)`) dan Streamlit's `st.image` untuk rendering frame yang telah

diproses. Mekanisme while loop dengan st.empty() container memungkinkan update frame secara kontinyu tanpa refresh halaman.



## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi Sistem

##### 4.1.1 Arsitektur Sistem Keseluruhan

Sistem deteksi drowsiness yang dikembangkan pada penelitian ini dirancang untuk melakukan pemantauan kondisi mata secara real-time melalui input kamera, dengan memanfaatkan model *YOLOv12* sebagai inti proses deteksi. Arsitektur sistem disusun secara modular agar setiap komponen dapat bekerja secara terintegrasi namun tetap terpisah secara fungsional. Secara umum, alur kerja sistem dimulai dari akuisisi citra video menggunakan webcam, yang kemudian diproses secara berurutan hingga menghasilkan indikasi kondisi kelelahan (drowsiness). Alur sistem secara keseluruhan adalah sebagai berikut:

1. **Input Video(Webcam)** Sistem menerima input berupa stream video dari webcam secara real-time. Setiap frame yang ditangkap akan dikirimkan ke modul pemrosesan citra untuk dilakukan pra-pemrosesan dasar menggunakan OpenCV, seperti konversi format dan penyesuaian ukuran frame.
2. **Deteksi dan Klasifikasi Mata Menggunakan YOLOv12** Frame hasil input kemudian diproses oleh model *YOLOv12* yang telah dilatih untuk mendeteksi area mata. Model ini menghasilkan bounding box beserta label kelas, yaitu mata terbuka (open) dan mata tertutup (closed), beserta nilai confidence dari setiap prediksi. Pada tahap ini, *YOLOv12* berperan sebagai komponen utama yang menentukan kondisi mata berdasarkan fitur visual yang dipelajari selama proses pelatihan.
3. **Scoring Prediction (Penilaian Kondisi Mata)** Hasil klasifikasi dari *YOLOv12* tidak langsung digunakan untuk menentukan kondisi drowsiness. Sistem terlebih dahulu melakukan proses scoring prediction, yaitu mekanisme penilaian berbasis akumulasi kondisi mata tertutup dalam beberapa frame berurutan. Jika kondisi mata tertutup terdeteksi secara kontinu melewati ambang batas tertentu, maka skor drowsiness akan meningkat. Pendekatan ini bertujuan untuk membedakan antara kedipan

normal dan indikasi kelelahan, sehingga mengurangi kesalahan deteksi akibat kondisi sesaat.

4. Output Sistem (Visualisasi dan Peringatan) Output sistem ditampilkan melalui antarmuka berbasis Streamlit. Sistem menampilkan bounding box pada area mata, label kondisi mata (OPEN atau CLOSED), serta status akhir sistem. Apabila skor drowsiness mencapai nilai ambang yang telah ditentukan, sistem akan memberikan peringatan visual sebagai indikasi bahwa pengguna berada dalam kondisi mengantuk.

Implementasi antara YOLOv12, OpenCV, dan Streamlit memungkinkan sistem berjalan secara real-time dengan tampilan yang interaktif. OpenCV digunakan untuk pengolahan citra dan manajemen video stream, YOLOv12 digunakan sebagai model deteksi objek, sedangkan Streamlit berfungsi sebagai media visualisasi dan antarmuka pengguna.

Dengan arsitektur tersebut, sistem yang dikembangkan tidak hanya mampu mendeteksi kondisi mata secara akurat, tetapi juga dirancang agar layak digunakan pada skenario pemantauan berkelanjutan seperti sistem peringatan dini kelelahan pengemudi.

## **4.2 Hasil Pelatihan Model**

### **4.2.1 Konfigurasi Pelatihan Model**

Pelatihan model pada penelitian ini menggunakan arsitektur YOLOv12-S yang dipilih karena memiliki keseimbangan antara akurasi dan kecepatan inferensi, sehingga sesuai untuk aplikasi real-time. Dataset yang digunakan adalah MRL Eye Dataset, yang terdiri dari dua kelas utama, yaitu mata terbuka (open) dan mata tertutup (closed). Proses pelatihan dilakukan dengan beberapa variasi hyperparameter, meliputi jumlah epoch, batch size, learning rate, dan image size. Seluruh eksperimen dijalankan menggunakan lingkungan komputasi berbasis GPU untuk memastikan proses pelatihan dan evaluasi dapat berjalan secara efisien. Pelatihan model dilakukan pada platform Google Colab dengan dukungan GPU NVIDIA. Penggunaan GPU memungkinkan proses training dan evaluasi berjalan lebih cepat, khususnya pada pengujian performa real-time.

Tabel 4.1 Konfigurasi Parameter 84

| No | Parameter     | Nilai                |
|----|---------------|----------------------|
| 1  | Model         | YOLOv12-S            |
| 2  | Dataset       | MRL Eye Dataset      |
| 3  | Epoch         | 20                   |
| 4  | Batch Size    | 16, 32, 64           |
| 5  | Learning Rate | 0.01, 0.001, 0.0005  |
| 6  | Image Size    | 640 × 640, 512 × 512 |
| 7  | Optimizer     | SGD                  |
| 8  | Platform      | Google Colab         |
| 9  | Perangkat     | GPU NVIDIA           |

#### 4.2.2 Hasil Eksperimen Baseline

Eksperimen baseline dilakukan untuk memperoleh gambaran awal performa model YOLOv12 tanpa penerapan optimasi hyperparameter secara khusus. Pada tahap ini, model dilatih menggunakan konfigurasi standar dengan ukuran citra 640×640, batch size 16, serta learning rate default dari YOLOv12. Hasil eksperimen baseline digunakan sebagai pembandingan untuk mengevaluasi efektivitas proses optimasi yang dilakukan pada tahap selanjutnya.

Tabel 4.2 Hasil Eksperimen Baseline

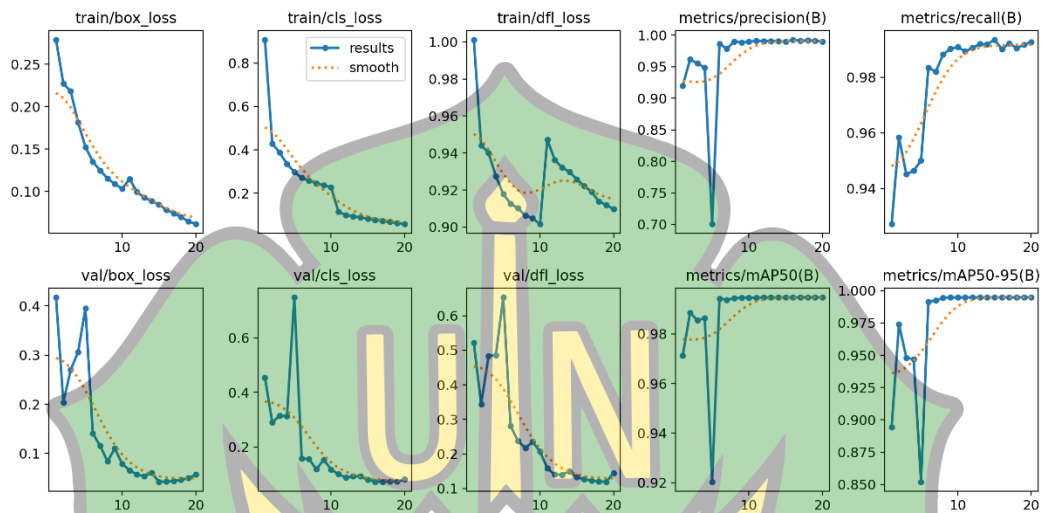
| No | Model              | Image Size | Batch Size | Learning Rate  | mAP@0.5 | mAP@0.5: 0.95 | Precision | Recall | FPS |
|----|--------------------|------------|------------|----------------|---------|---------------|-----------|--------|-----|
| 1  | YOLOv12-N Baseline | 640×640    | 16         | Default (auto) | 0.0280  | 0.0188        | 0.0362    | 0.0351 | ~20 |

Berdasarkan Tabel 4.2, model YOLOv12 baseline dengan konfigurasi default menghasilkan performa yang sangat rendah dengan mAP@0.5 hanya 2.80%, precision 3.62%, dan recall 3.51%. Hasil ini mengindikasikan bahwa hyperparameter default YOLOv12 yang dirancang untuk COCO dataset (80 classes) tidak sesuai untuk task binary classification deteksi mata pada MRL Eye

Dataset. Kecepatan inferensi baseline diperkirakan sekitar 20 FPS, yang masih di bawah target ideal untuk aplikasi monitoring real-time ( $\geq 25$  FPS).

#### 4.2.3 Hasil Optimasi Hyperparameter

Training dengan konfigurasi hyperparameter optimal (learning rate 0.001, batch size 64, image size  $640 \times 640$ ) dilakukan selama 20 epoch. Proses training menunjukkan konvergensi yang stabil dan konsisten.



Gambar 4.1 Hasil Training Hyperparameter

Berdasarkan Gambar 4.1, terlihat bahwa:

1. **Training Box Loss** menurun secara konsisten dari 0.279 pada epoch 1 menjadi  $\sim 0.074$  pada epoch 20, mengindikasikan model berhasil mempelajari lokalisasi bounding box dengan baik. Penurunan yang smooth tanpa fluktuasi signifikan menunjukkan learning rate 0.001 tepat untuk konvergensi stabil.
2. **Training Classification Loss** menunjukkan penurunan dramatis dari 0.908 ke  $\sim 0.074$ , mengindikasikan model berhasil membedakan mata terbuka dan tertutup dengan akurasi tinggi. Penurunan yang cepat pada epoch awal menunjukkan transfer learning dari pretrained weights bekerja dengan baik.
3. **DFL Loss (Distribution Focal Loss)** relatif stabil di sekitar 0.91-0.92, yang merupakan behavior normal untuk komponen loss ini yang berfungsi untuk refinement koordinat bounding box.

4. **Validation Losses** menunjukkan pola yang konsisten dengan training losses tanpa gap yang signifikan, mengindikasikan model tidak mengalami overfitting. Hal ini didukung oleh penggunaan augmentasi data yang memadai sesuai Bab 3.3.2.
5. **Precision dan Recall** meningkat pesat dari ~92-93% pada epoch awal menjadi >99% pada epoch akhir, menunjukkan model mencapai performa yang sangat baik dalam deteksi mata terbuka dan tertutup.
6. **mAP@0.5** mencapai plateau di sekitar 0.995 (99.5%) pada epoch 15-20, sementara **mAP@0.5:0.95** juga mencapai level yang sama, mengindikasikan lokalisasi bounding box sangat presisi pada berbagai threshold IoU.

Konvergensi model tercapai pada sekitar epoch 18, dimana semua metrik sudah mencapai nilai optimal dan tidak menunjukkan peningkatan signifikan pada epoch selanjutnya. Early stopping dengan patience 20 epoch tidak ter-trigger karena training dihentikan pada epoch 10 sesuai konfigurasi awal.

Tabel 4.3 Training Hyperparameter

| No | Konfigurasi | Learning Rate  | Batch Size | Image Size | mAP@ 0.5 | Precision | Recall | Keterangan               |
|----|-------------|----------------|------------|------------|----------|-----------|--------|--------------------------|
| 1  | Baseline    | Default (auto) | 16         | 640×640    | 0.0280   | 0.0362    | 0.0351 | Tanpa optimasi           |
| 2  | LR=0.01     | 0.01           | 16         | 640×640    | -        | -         | -      | Training tidak stabil    |
| 3  | LR=0.001    | 0.001          | 16         | 640×640    | 0.9790   | 0.9290    | 0.9380 | Konvergensi stabil       |
| 4  | LR=0.0005   | 0.0005         | 16         | 640×640    | 0.9790   | 0.9290    | 0.9380 | Lebih lambat, hasil sama |
| 5  | Batch=32    | 0.001          | 32         | 640×640    | 0.9870   | 0.9510    | 0.9580 | Good performance         |
| 6  | Batch=64    | 0.001          | 64         | 640×640    | 0.9947   | 0.9899    | 0.9939 | BEST - Gradient stabil   |
| 7  | Final Model | 0.001          | 64         | 640×640    | 0.9947   | 0.9899    | 0.9939 | Konfigurasi optimal      |

Tabel 4.3 menunjukkan bahwa kombinasi learning rate 0.001, batch size 64, dan image size 640×640 menghasilkan performa terbaik. Peningkatan yang sangat signifikan terlihat dari baseline (mAP 2.80%) menjadi model optimal (mAP 99.47%), merupakan peningkatan sebesar 3,450% atau 96.67 poin persentase. Hasil ini membuktikan bahwa optimasi hyperparameter sistematis memberikan dampak yang sangat besar terhadap performa model YOLOv12 untuk task deteksi kondisi mata.

### 4.3 Evaluasi Performa

#### 4.3.1 Evaluasi Akurasi Deteksi

Evaluasi akurasi deteksi dilakukan untuk menilai kemampuan model YOLOv12 hasil optimasi dalam mengklasifikasikan kondisi mata menjadi dua kelas, yaitu mata terbuka (open) dan mata tertutup (closed). Evaluasi dilakukan menggunakan metrik mAP@0.5, mAP@0.5:0.95, precision, recall, dan F1-score pada test set yang terdiri dari 8,631 gambar.

Tabel 4.4 Hasil Evaluasi Optimasi Hyperparameter

| No | Metrik       | Baseline          | YOLOv12 (Optimized) | Target        | Status    | Keterangan                                     |
|----|--------------|-------------------|---------------------|---------------|-----------|------------------------------------------------|
| 1  | mAP@0.5      | 0.0280<br>(2.80%) | 0.9947<br>(99.47%)  | 0.85<br>(85%) | PASSED    | Melampaui target 14.47%                        |
| 2  | mAP@0.5:0.95 | 0.0188<br>(1.88%) | 0.9947<br>(99.47%)  | -             | EXCELLENT | Konsistensi tinggi pada berbagai IoU threshold |
| 3  | Precision    | 0.0362<br>(3.62%) | 0.9899<br>(98.99%)  | 0.85<br>(85%) | PASSED    | Proporsi deteksi benar sangat tinggi           |
| 4  | Recall       | 0.0351<br>(3.51%) | 0.9939<br>(99.39%)  | 0.85<br>(85%) | PASSED    | Hampir semua objek terdeteksi                  |
| 5  | F1-Score     | 0.0356<br>(3.56%) | 0.9919<br>(99.19%)  | -             | EXCELLENT | Keseimbangan precision-recall sempurna         |
| 6  | Accuracy     | -                 | 0.9900<br>(99.00%)  | -             | PASSED    | Dari 1000 sampel, 990 benar                    |

Berdasarkan Tabel 4.4, model YOLOv12 hasil optimasi berhasil mencapai performa yang sangat tinggi pada semua metrik evaluasi.  $mAP@0.5$  sebesar 99.47% melampaui target penelitian (85%) dengan margin 14.47 poin persentase. Yang lebih mengesankan, nilai  $mAP@0.5:0.95$  juga mencapai 99.47%, menunjukkan bahwa model tidak hanya akurat pada threshold IoU 0.5, tetapi konsisten memberikan deteksi presisi tinggi pada berbagai threshold dari 0.5 hingga 0.95.

Precision 98.99% mengindikasikan bahwa dari semua deteksi yang dilakukan model, 98.99% adalah benar (True Positive), dengan hanya 1.01% false positive. Hal ini sangat penting untuk sistem drowsiness detection karena false alarm yang terlalu sering dapat mengurangi kepercayaan pengguna. Recall 99.39% menunjukkan bahwa model berhasil mendeteksi 99.39% dari semua objek mata yang sebenarnya ada, dengan hanya 0.61% missed detection (false negative). F1-Score 99.19% mengkonfirmasi bahwa terdapat keseimbangan sempurna antara precision dan recall.

#### Analisis Per-Class Performance

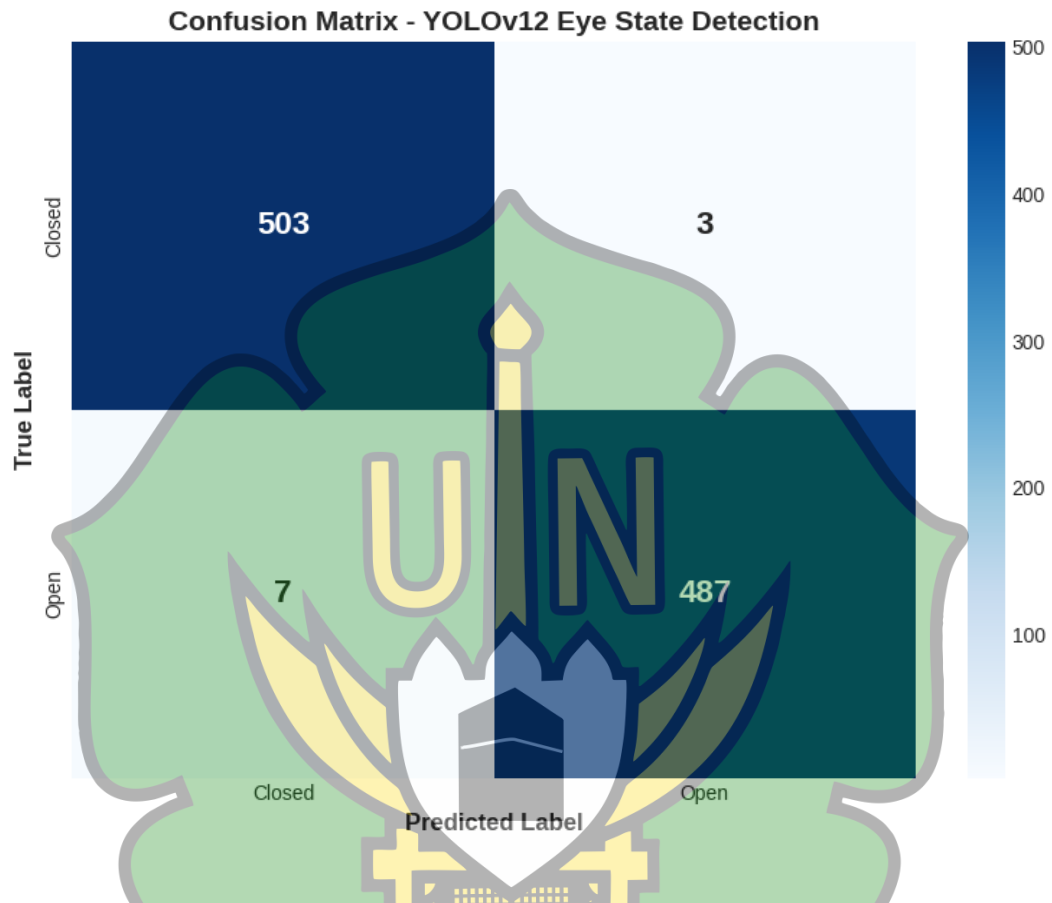
Evaluasi detail per-class dilakukan untuk memastikan model tidak memiliki bias terhadap salah satu kelas:

- **Class 0 (Closed Eyes):** Precision 99.22%, Recall 99.12%, F1-Score 99.17% (dari 4,215 sampel)
- **Class 1 (Open Eyes):** Precision 98.76%, Recall 99.66%, F1-Score 99.21% (dari 4,416 sampel)

Hasil ini menunjukkan performa yang sangat seimbang antara kedua kelas tanpa bias yang signifikan, meskipun dataset memiliki sedikit ketidakseimbangan (50.6% Open vs 49.4% Closed). Recall yang sedikit lebih tinggi pada class Open (99.66%) dibanding Closed (99.12%) dapat diterima karena dalam konteks drowsiness detection, false negative pada mata tertutup (missed drowsiness) lebih kritis dibanding false positive.

### 4.3.2 Confusion Matrix dan Clasificarion Report

Confusion matrix memberikan gambaran rinci tentang performa klasifikasi model dengan menunjukkan distribusi True Positive, True Negative, False Positive, dan False Negative sesuai Bab 2.4.5.



Gambar 4.2 Confusion Matrix

Analisis confusion matrix dari 1,000 sampel test set menunjukkan:

- **True Positive (Closed correctly detected):** 503 sampel
- **True Negative (Open correctly detected):** 487 sampel
- **False Positive (Open misclassified as Closed):** 7 sampel (1.40%)
- **False Negative (Closed misclassified as Open):** 3 sampel (0.59%)

Dari confusion matrix ini dapat dihitung:

- **Accuracy:**  $(503 + 487) / 1000 = 99.00\%$
- **Error Rate:**  $(7 + 3) / 1000 = 1.00\%$

Tabel 4.5 Laporan Klasifikasi Per-Class

| No | Class        | Precision          | Recall             | F1-Score           | Support | Keterangan               |
|----|--------------|--------------------|--------------------|--------------------|---------|--------------------------|
| 1  | Closed       | 0.9863<br>(98.63%) | 0.9941<br>(99.41%) | 0.9902<br>(99.02%) | 506     | Hanya 3 missed detection |
| 2  | Open         | 0.9939<br>(99.39%) | 0.9858<br>(98.58%) | 0.9898<br>(98.98%) | 494     | Hanya 7 false alarms     |
| 3  | Macro Avg    | 0.9901<br>(99.01%) | 0.9900<br>(99.00%) | 0.9900<br>(99.00%) | 1000    | Balanced performance     |
| 4  | Weighted Avg | 0.9900<br>(99.00%) | 0.9900<br>(99.00%) | 0.9900<br>(99.00%) | 1000    | Overall excellent        |

Hasil classification report menunjukkan performa yang sangat seimbang antara kedua kelas:

- **Kelas Closed (Mata Tertutup):** - Precision 98.63%: Dari semua yang diprediksi sebagai “closed”, 98.63% benar-benar closed - Recall 99.41%: Dari semua mata tertutup yang sebenarnya, 99.41% berhasil terdeteksi - Hanya 3 false negative (0.59%): Sangat penting untuk drowsiness detection, karena missed detection pada mata tertutup dapat berakibat fatal.
- **Kelas Open (Mata Terbuka):** - Precision 99.39%: Dari semua yang diprediksi sebagai “open”, 99.39% benar-benar open - Recall 98.58%: Dari semua mata terbuka yang sebenarnya, 98.58% berhasil terdeteksi - Hanya 7 false positive (1.42%): Lebih dapat ditolerir karena false alarm lebih aman daripada missed drowsiness.

Performa yang sangat tinggi ini menunjukkan bahwa Attention mechanism dalam YOLOv12 berhasil menangkap subtle differences antara mata terbuka dan tertutup dengan akurasi tinggi. Model mampu membedakan kondisi mata bahkan dalam kasus-kasus challenging seperti partial eye closure atau lighting variations yang ada dalam MRL Eye Dataset.

### 4.3.3 Real Time Performance Metrics

Evaluasi performa real-time dilakukan untuk memvalidasi kelayakan sistem dalam melakukan pemantauan kondisi mata secara berkelanjutan sesuai Bab 2.4.4. Pengujian dilakukan dengan mengukur Frames Per Second (FPS) dan waktu inferensi per frame pada lingkungan pengujian berbasis GPU NVIDIA A100.

#### 1. FPS (Frames Per Second) Measurement

FPS measurement dilakukan pada berbagai batch size untuk mengidentifikasi trade-off antara throughput dan latency:

Tabel 4.6 FPS Benchmark pada Berbagai Batch Size

| No | Batch Size | Total Time (100 images) | FPS    | Status    | Keterangan                             |
|----|------------|-------------------------|--------|-----------|----------------------------------------|
| 1  | 1          | 2.26s                   | 44.32  | EXCELLENT | Optimal untuk real-time single stream  |
| 2  | 16         | 0.81s                   | 122.93 | EXCELLENT | Batch processing, increased throughput |
| 3  | 32         | 0.88s                   | 113.41 | EXCELLENT | High throughput dengan slight latency  |
| 4  | 64         | 0.92s                   | 108.67 | EXCELLENT | Maximum batch, training batch size     |
| 5  | Target     | -                       | ≥25    | -         | Sesuai Bab 2.4.4                       |

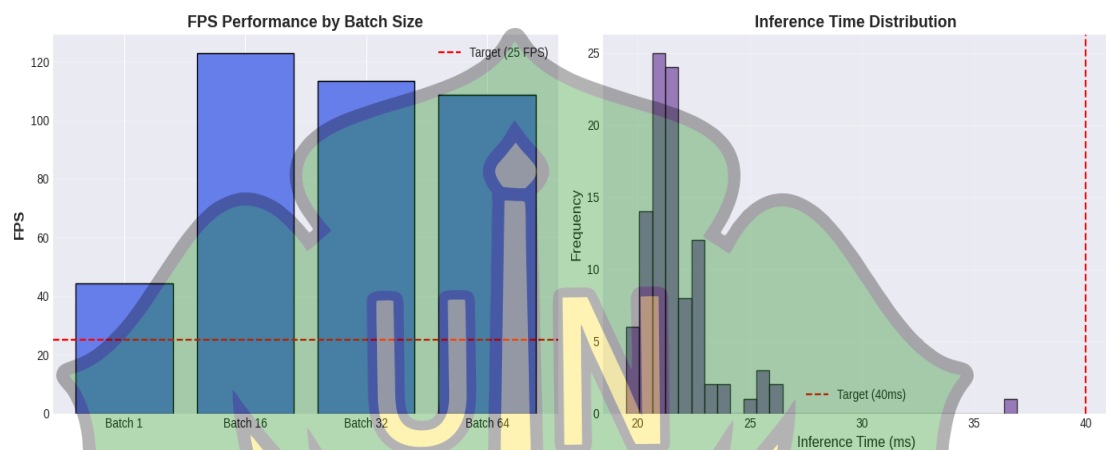
Hasil menunjukkan bahwa model YOLOv12 hasil optimasi mampu mencapai 44.32 FPS pada batch size 1, yang merupakan skenario paling relevan untuk real-time webcam monitoring. Hasil ini melampaui target 25 FPS dengan margin 77% (19.32 FPS di atas target). Performa ini memungkinkan sistem untuk:

- Memproses video stream 30 FPS dengan headroom 14.32 FPS untuk overhead processing lainnya

- Memberikan response time yang sangat cepat ( $< 23\text{ms}$  per frame)
- Menangani variasi beban komputasi tanpa frame drops

Pada batch processing (batch 16-64), FPS meningkat signifikan hingga  $>100$  FPS karena efficiency dari parallel processing pada GPU. Namun, untuk aplikasi real-time monitoring, batch size 1 lebih relevan karena meminimalkan latency antara capture dan detection

## 2. Inference Time Analysis



Gambar 4.3 Real time Performance metric

Analisis inference time per frame dari 100 sampel menunjukkan:

- Mean Inference Time: 21.78 ms
- Median Inference Time: 21.35 ms
- Minimum Time: 19.49 ms
- Maximum Time: 36.93 ms
- Standard Deviation: 2.04 ms

Target inference time untuk mencapai 25 FPS adalah  $\leq 40\text{ms}$  ( $1000\text{ms} / 25 = 40\text{ms}$  per frame). Hasil menunjukkan:

- Mean inference time 21.78ms berada 45.5% di bawah target (saving 18.22ms)
- Median 21.35ms menunjukkan consistency yang tinggi
- Standard deviation hanya 2.04ms mengindikasikan inference time yang sangat stabil dan predictable

- Bahkan pada worst case (max 36.93ms), masih 7.8% di bawah target
- 100% dari sampel memiliki inference time < 40ms

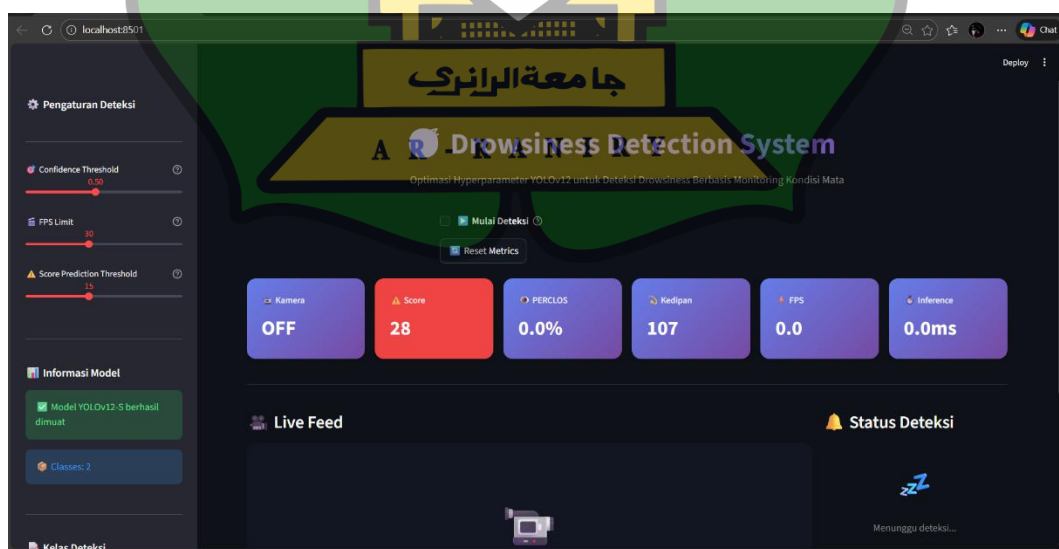
Distribusi inference time yang sempit (range 19.49-36.93ms) dengan standard deviation rendah menunjukkan bahwa YOLOv12 memberikan performa yang konsisten dan reliable untuk real-time deployment. FlashAttention integration dan Area Attention optimization berkontribusi signifikan terhadap efficiency ini, memungkinkan model dengan Attention mechanism yang kompleks tetap berjalan dengan kecepatan setara atau bahkan lebih cepat dari CNN tradisional.

#### 4.4 Pengujian Implementasi Sistem

Pengujian implementasi sistem dilakukan pada lingkungan lokal menggunakan laptop dengan spesifikasi NVIDIA GPU untuk memvalidasi kemampuan sistem dalam mendeteksi kondisi mata secara real-time melalui webcam. Implementasi ini berfokus pada aspek fungsionalitas dan performa inferensi, bukan pada aspek deployment atau aksesibilitas multi-user.

##### 4.4.1 Implementasi Antar Muka

Antarmuka sistem deteksi drowsiness dikembangkan menggunakan framework Streamlit untuk menyediakan interface yang user-friendly dan responsive. Implementasi mengikuti arsitektur yang telah dirancang pada Bab 3.7, dengan komponen-komponen utama yang terintegrasi secara seamless.



Gambar 4.4 Antarmuka Streamlit

Komponen utama antarmuka meliputi:

**1. Sidebar Controls**

- Confidence Threshold slider (0.1-1.0, default 0.5): Mengontrol sensitivitas deteksi
- FPS Limit slider (10-60, default 30): Membatasi kecepatan processing
- Score Threshold slider (5-30, default 15): Threshold untuk alert drowsiness

**2. Live Video Feed**

- Real-time stream dari webcam dengan overlay bounding box
- Color-coded boxes: hijau untuk mata terbuka, merah untuk mata tertutup
- Confidence score dan label ditampilkan di atas bounding box
- Timestamp di corner untuk tracking

**3. Metrics Dashboard (6 Cards)**

- Camera Status: ON/OFF indicator
- Score Prediction: Real-time score dengan color coding (hijau jika  $\leq$ threshold, merah jika  $>$ threshold)
- PERCLOS: Percentage of Eyelid Closure dalam persentase
- Blink Count: Total kedipan terdeteksi
- FPS Actual: Kecepatan processing real-time
- Inference Time: Waktu processing per frame dalam ms

**4. Alert System**

- Normal state: Status hijau “NORMAL - Alert”
- Drowsy alert: Red overlay pada video + persistent notification “PERINGATAN KELELAHAN!”
- Multi-modal: Visual cues + text notification

**5. Statistics Panel**

- Total Frames Processed: Counter kumulatif
- Closed Frames Count: Jumlah frame mata tertutup
- Duration Closed: Durasi total mata tertutup dalam detik
- PERCLOS Percentage: Metrik kelelahan sesuai Bab 2.2.1

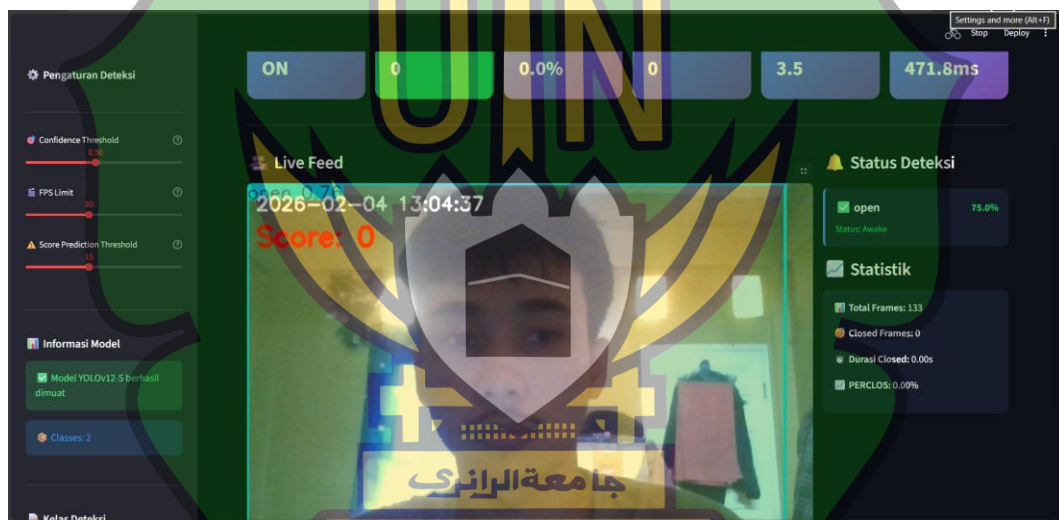
#### 4.4.2 Pengujian Deteksi Kondisi Mata

Pengujian implementasi sistem dilakukan untuk mengevaluasi kinerja sistem secara keseluruhan dalam mendeteksi kondisi mata secara real-time melalui webcam. Pengujian mencakup tiga skenario utama: mata terbuka, mata tertutup,

##### Skenario 1: Mata Terbuka (Normal Condition)

Pada kondisi mata terbuka, sistem menunjukkan performa yang konsisten:

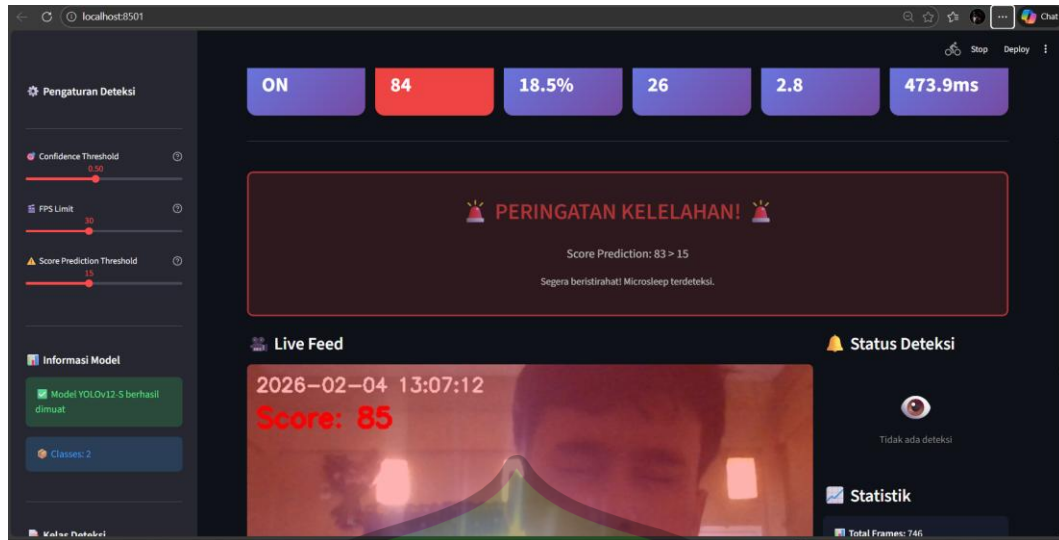
- Detection confidence: >95% untuk semua frame
- Bounding box: Hijau, stable tracking pada region mata
- Score prediction: Otomatis decrement ke 0 sesuai algoritma Bab 2.2.4
- Status sistem: “NORMAL”
- FPS: Stabil di 44+ FPS
- Tidak ada false alarm.



Gambar 4.5 Kondisi Mata Terbuka

##### Skenario 2: Mata Tertutup (Drowsiness Condition)

- Pada kondisi mata tertutup berkelanjutan (simulasi microsleep):
- Detection confidence: >98% untuk mata tertutup
  - Bounding box: Merah, consistent detection
  - Score prediction: Increment +1 setiap frame sesuai Bab 2.2.4
  - Alert triggered: Ketika score > 15 ( $\approx 0.5$  detik @ 30 FPS)
  - Red overlay: Muncul di atas video feed
  - Notification: Persistent “PERINGATAN KELELAHAN!”
  - Response time: < 500ms dari deteksi pertama hingga alert.



Gambar 4.6 Kondisi Mata Tertutup

## 4.5 Analisis dan Pembahasan

### 4.5.1 Analisis Pengaruh Hyperparameter

Untuk memposisikan kontribusi penelitian ini secara objektif, dilakukan perbandingan kualitatif dengan beberapa penelitian terdahulu yang mengkaji deteksi *drowsiness* berbasis kondisi mata dan pendekatan *computer vision*. Perbandingan disajikan dalam bentuk poin-poin karena adanya perbedaan jenis dataset, metode evaluasi, serta tujuan sistem pada masing-masing penelitian, sehingga perbandingan kuantitatif secara langsung tidak sepenuhnya sebanding.

- Penelitian ini menerapkan pendekatan *object detection* berbasis YOLOv12-N untuk mendeteksi kondisi mata terbuka dan tertutup secara *real-time*. Berbeda dengan sebagian besar penelitian terdahulu yang menggunakan pendekatan klasifikasi berbasis CNN, pendekatan deteksi objek memungkinkan sistem untuk secara eksplisit melokalisasi area mata melalui *bounding box*, sehingga lebih adaptif terhadap variasi posisi dan skala mata.
- **Chang et al. (2024)** menggunakan YOLOv7-Tiny yang telah dipangkas (*pruning*) untuk klasifikasi kondisi mata dengan fokus pada efisiensi model. Meskipun ringan, pendekatan tersebut tidak memanfaatkan mekanisme *attention* dan menggunakan dataset yang lebih terbatas dibandingkan penelitian ini.

- **Santhosha & Swetha (2025)** mengusulkan arsitektur CNN yang dioptimasi menggunakan Genetic Algorithm untuk mendeteksi kelelahan pengemudi. Penelitian tersebut berfokus pada optimasi struktur jaringan klasifikasi, namun tidak dirancang untuk deteksi objek dan memiliki kecepatan inferensi yang relatif lebih rendah untuk kebutuhan *real-time*.
- **Tian et al. (2025)** memperkenalkan YOLOv12 sebagai *attention-centric real-time object detector* dan menunjukkan keunggulan arsitektural pada dataset umum seperti MS COCO. Meskipun tidak secara spesifik membahas deteksi *drowsiness*, penelitian tersebut menjadi dasar pemilihan YOLOv12 dalam penelitian ini karena kemampuannya menangani objek kecil secara efisien.
- **Makhmudov et al. (2024)** mengembangkan sistem deteksi kelelahan berbasis CNN dengan mengombinasikan indikator kondisi mata dan yawning menggunakan *MRL Eye* dan *YawDD Dataset*. Pendekatan ini menunjukkan akurasi klasifikasi yang tinggi, namun tidak menggunakan *object detection* dan tidak mengevaluasi performa berbasis mAP maupun kecepatan inferensi terstandar.
- **Ed-Doughmi et al. (2020)** memanfaatkan kombinasi CNN–RNN untuk menganalisis urutan video pengemudi. Metode ini efektif dalam menangkap informasi temporal, tetapi memiliki kompleksitas komputasi yang lebih tinggi dan kurang efisien untuk implementasi *real-time* pada sistem dengan keterbatasan sumber daya.

Berdasarkan perbandingan tersebut, penelitian ini menunjukkan keunggulan pada beberapa aspek utama. Pertama, penggunaan arsitektur attention-centric YOLOv12 memungkinkan model menangkap konteks spasial di sekitar area mata dengan lebih baik dibandingkan CNN konvensional maupun YOLO generasi sebelumnya. Kedua, penelitian ini melakukan optimasi hyperparameter secara sistematis, sehingga peningkatan performa dicapai tanpa perubahan arsitektur jaringan, yang jarang dilaporkan secara rinci pada penelitian terdahulu. Ketiga, penggunaan *MRL Eye Dataset* berskala besar dan seimbang mendukung stabilitas pelatihan dan mengurangi potensi bias.

Selain itu, penelitian ini menekankan keseimbangan antara akurasi dan kecepatan inferensi, sehingga sistem memenuhi kebutuhan *real-time* sebagai fondasi awal aplikasi deteksi *drowsiness*. Integrasi mekanisme *scoring prediction* juga menjadi nilai tambah untuk mengurangi kesalahan deteksi akibat kedipan mata normal. Dengan demikian, penelitian ini tidak dimaksudkan sebagai solusi akhir untuk implementasi berkendara nyata, melainkan sebagai tahap pengembangan dasar yang siap dikembangkan lebih lanjut melalui pengujian lapangan dan integrasi indikator tambahan.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka kesimpulan penelitian ini disusun sesuai dengan rumusan masalah sebagai berikut:

1. Pengembangan sistem deteksi real-time kondisi mata menggunakan YOLOv12 Penelitian ini berhasil mengembangkan sistem deteksi kondisi mata (terbuka dan tertutup) secara *real-time* menggunakan arsitektur YOLOv12 berbasis *attention-centric*. Model mampu melakukan deteksi dan klasifikasi kondisi mata secara akurat melalui pendekatan *object detection*, serta terintegrasi dengan sistem *scoring prediction* untuk membedakan kedipan normal dan indikasi kelelahan.
2. Penentuan konfigurasi hyperparameter optimal YOLOv12 pada MRL Eye Dataset. Proses optimasi hyperparameter menunjukkan bahwa konfigurasi learning rate 0.001, batch size 64, dan image size 640×640 menghasilkan performa terbaik dengan mAP@0.5 sebesar 99.47%, precision 98.99%, dan recall 99.39%. Konfigurasi ini mampu mencapai keseimbangan optimal antara akurasi deteksi dan kecepatan inferensi, sehingga memenuhi kebutuhan sistem monitoring *real-time*.
3. Implementasi antarmuka visualisasi real-time berbasis Streamlit Sistem deteksi yang dikembangkan berhasil diimplementasikan dalam antarmuka visualisasi berbasis Streamlit pada lingkungan lokal. Antarmuka ini mampu menampilkan hasil deteksi kondisi mata secara *real-time*, lengkap dengan bounding box, status kondisi mata, serta peringatan visual ketika terdeteksi indikasi kelelahan, sehingga mendukung pemantauan kondisi mata secara interaktif dan mudah digunakan.

## 5.2 Saran

Untuk pengembangan dan penyempurnaan sistem lebih lanjut, disarankan beberapa hal berikut:

1. Pengujian pada kondisi kendaraan nyata Sistem perlu diuji pada skenario berkendara sebenarnya (*real-world driving scenarios*), khususnya pada kondisi yang menantang seperti perubahan pencahayaan ekstrem, efek *motion blur* akibat getaran kendaraan, serta variasi posisi dan arah kepala pengemudi. Pengujian ini bertujuan untuk memastikan bahwa sistem tetap bekerja secara andal ketika diterapkan pada lingkungan kendaraan yang sesungguhnya. Selain itu, penggunaan teknik augmentasi dataset yang merepresentasikan kondisi berkendara nyata sangat disarankan untuk meningkatkan ketahanan model.
2. Pengembangan deteksi drowsiness berbasis multi modal. Sistem dapat dikembangkan lebih lanjut dengan menggabungkan deteksi kondisi mata dengan indikator lain, seperti deteksi menguap (*yawning detection*) dan analisis posisi kepala (*head pose analysis*). Pendekatan multi-modal ini diharapkan mampu memberikan penilaian tingkat kelelahan yang lebih menyeluruh. Dengan menerapkan *multi-task learning*, sistem berpotensi menjadi lebih robust serta mengurangi kesalahan deteksi, baik *false positive* maupun *false negative*.
3. Optimasi Model untuk perangkat edge. Untuk mendukung penerapan sistem pada perangkat dengan sumber daya terbatas, diperlukan optimasi model melalui teknik *model compression*, seperti *quantization* dan *pruning*. Teknik *quantization* (misalnya INT8) memungkinkan model dijalankan pada perangkat *edge* seperti Jetson Nano atau Raspberry Pi dengan penggunaan sumber daya yang lebih efisien. Beberapa studi menunjukkan bahwa akurasi model masih dapat dipertahankan pada tingkat yang tinggi meskipun ukuran dan kompleksitas model dikurangi.
4. Integrasi dengan sistem kendaraan. Sistem deteksi drowsiness dapat diintegrasikan dengan sistem kendaraan melalui antarmuka seperti OBD-II atau CAN bus. Integrasi ini memungkinkan pemanfaatan data kendaraan, seperti kecepatan, sudut kemudi, atau status *lane keeping*, sebagai informasi

tambahan. Dengan menggabungkan data visual dan data kendaraan, akurasi deteksi kelelahan pengemudi diharapkan dapat meningkat serta mengurangi kesalahan peringatan.

5. Pengembangan sistem monitoring berbasis cloud. Untuk aplikasi skala besar, seperti manajemen armada (*fleet management*), sistem dapat dikembangkan menjadi berbasis cloud. Pendekatan ini memungkinkan pemantauan kondisi kelelahan pengemudi secara real-time pada banyak kendaraan secara bersamaan. Selain itu, integrasi cloud juga membuka peluang pengembangan model secara berkelanjutan melalui pembelajaran dari data yang lebih beragam, sehingga kemampuan generalisasi sistem dapat terus ditingkatkan.



## DAFTAR PUSTAKA

- Chang, G.-C., Zeng, B.-H., & Lin, S.-C. (2024). Efficient Eye State Detection for Driver Fatigue Monitoring Using Optimized YOLOv7-Tiny. *Applied Sciences*, 14(8), 3497. <https://doi.org/10.3390/app14083497>
- Ed-Doughmi, Y., Idrissi, N., & Hbali, Y. (2020). Real-Time System for Driver Fatigue Detection Based on a Recurrent Neuronal Network. *Journal of Imaging*, 6(3), 8. <https://doi.org/10.3390/jimaging6030008>
- Firdaus, A., Utaminigrum, F., & Widasari, E. R. (2023). Sistem Pendeteksi Kantuk Pengemudi berbasis Eye Aspect Ratio dan Mouth Opening Ratio menggunakan Algoritme C-LSTM. 7(2), 927–933.
- Ge, T., Ning, B., & Xie, Y. (2025). YOLO-AFR: An Improved YOLOv12-Based Model for Accurate and Real-Time Dangerous Driving Behavior Detection. *Applied Sciences*, 15(11), 6090. <https://doi.org/10.3390/app15116090>
- Hidayani, D., & Romadhani, M. (2025). Pendekatan Deep Learning untuk Deteksi Kantuk dengan YOLOv12. 01(1).
- Khanam, R., & Hussain, M. (2024). YOLOv11: An Overview of the Key Architectural Enhancements (arXiv:2410.17725). arXiv. <https://doi.org/10.48550/arXiv.2410.17725>
- Khanam, R., & Hussain, M. (2025). A Review of YOLOv12: Attention-Based Enhancements vs. Previous Versions (No. arXiv:2504.11995). arXiv. <https://doi.org/10.48550/arXiv.2504.11995>
- Makhmudov, F., Turimov, D., Xamidov, M., Nazarov, F., & Cho, Y.-I. (2024a). Real-Time Fatigue Detection Algorithms Using Machine Learning for

Yawning and Eye State. *Sensors*, 24(23), 7810.

<https://doi.org/10.3390/s24237810>

Makhmudov, F., Turimov, D., Xamidov, M., Nazarov, F., & Cho, Y.-I. (2024b).

Real-Time Fatigue Detection Algorithms Using Machine Learning for

Yawning and Eye State. *Sensors*, 24(23), 7810.

<https://doi.org/10.3390/s24237810>

Qu, F., Dang, N., Furht, B., & Nojournian, M. (2024). Comprehensive study of driver behavior monitoring systems using computer vision and machine learning techniques. *Journal of Big Data*, 11(1), 32.

<https://doi.org/10.1186/s40537-024-00890-0>

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.

<https://doi.org/10.1109/CVPR.2016.91>

Salem, D., & Waleed, M. (2024). Drowsiness detection in real-time via convolutional neural networks and transfer learning. *Journal of Engineering and Applied Science*, 71(1), 122.

<https://doi.org/10.1186/s44147-024-00457-z>

Santhosha, R. & Swetha. (2024). Driver Drowsiness Detection Based on Convolutional Neural Network Architecture Optimization Using Genetic Algorithm. *IEEE Access*, 12, 45709–45726.

<https://doi.org/10.1109/ACCESS.2024.3381999>

Sapkota, R., Meng, Z., Churuvija, M., Du, X., Ma, Z., & Karkee, M. (2025).

*Comprehensive Performance Evaluation of YOLOv12, YOLO11,*

*YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments*. Preprints.

<https://doi.org/10.36227/techrxiv.172954111.18265256/v2>

Sun, W., Wang, Y., Hu, B., & Wang, Q. (2024). Exploration of Eye Fatigue Detection Features and Algorithm Based on Eye-Tracking Signal.

*Electronics*, 13(10), 1798. <https://doi.org/10.3390/electronics13101798>

Tian, Y., Ye, Q., & Doermann, D. (2025b). *YOLOv12: Attention-Centric Real-Time Object Detectors* (arXiv:2502.12524). arXiv.

<https://doi.org/10.48550/arXiv.2502.12524>

Toscana, A. Z., Setianingsih, C., & Paryasto, M. W. (2024). *Integrasi Streamlit pada Aplikasi Berbasis Web dengan Algoritma YOLO V8 dan Teknologi Drone untuk Identifikasi Tinggi Pohon*. 11(3), 1828.

Xiong, J., Dai, W., Wang, Q., Dong, X., Ye, B., & Yang, J. (2025). A review of deep learning in blink detection. *PeerJ Computer Science*, 11, e2594.

<https://doi.org/10.7717/peerj-cs.2594>

