

Terbit online pada laman : <http://teknosi.fti.unand.ac.id/>

Jurnal Nasional Teknologi dan Sistem Informasi

| ISSN (Print) 2460-3465 | ISSN (Online) 2476-8812 |



Artikel Penelitian

Aplikasi Alarm Cerdas Berbasis MobileNetV2 TensorFlow Lite Verifikasi Kesadaran Pengguna

Muhammad Zakfar Sodik^a, Renhat Denil Ramadhan^a, Andika Sandi Pranata^a, Zidan Alghifari^a, Dani Rofianto^a, Khusnatul Amaliah^a

^aProgram Studi Teknologi Rekayasa Perangkat Lunak, Jurusan Teknologi Informasi, Politeknik Negeri Lampung, Jl. Soekarno Hatta No. 10, Rajabasa, Bandar Lampung 35141, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 17 Juli 2026

Revisi Akhir: 15 September 2026

Diterbitkan Online: 21 September 2026

KATA KUNCI

Artificial Intelligence,
MobileNetV2,
TensorFlow Lite,
Verifikasi Kesadaran,
Android

KORESPONDENSI

E-mail: zakfarsdk@gmail.com *

ABSTRACT

Rendahnya kemampuan alarm konvensional dalam memastikan pengguna benar-benar terbangun masih menjadi penyebab keterlambatan bangun pada sebagian pelajar, mahasiswa maupun pekerja. Sebagian besar aplikasi alarm konvensional hanya mengandalkan keluaran berupa suara atau getaran tanpa mekanisme untuk memastikan bahwa pengguna benar-benar telah terbangun. Penelitian ini bertujuan merancang aplikasi WakeUp AI, yaitu aplikasi alarm cerdas berbasis Android yang memanfaatkan kecerdasan buatan untuk memverifikasi kondisi pengguna sebelum alarm dapat dihentikan. Sistem menerapkan metode Transfer Learning dengan arsitektur MobileNetV2 untuk mengklasifikasikan kondisi mata terbuka dan mata tertutup. Sebelum proses klasifikasi dilakukan, citra diproses menggunakan teknik Eye Region Cropping sehingga model hanya memproses area mata. Selanjutnya model dijalankan secara lokal menggunakan TensorFlow Lite agar dapat bekerja tanpa koneksi internet. Untuk meningkatkan kestabilan hasil deteksi diterapkan algoritma Rolling Average dan Leaky Bucket Accumulator, kemudian proses verifikasi dilanjutkan dengan mengimplementasikan tantangan Random Question Challenge sederhana sebagai verifikasi kognitif pengguna. Berdasarkan implementasi dan pengujian yang telah dilakukan, sistem mampu menghasilkan aplikasi alarm yang ringan, cepat, serta lebih efektif dibandingkan alarm konvensional dalam memastikan pengguna benar-benar berada pada kondisi sadar sebelum alarm dinonaktifkan.

1. PENDAHULUAN

Kemajuan teknologi Artificial Intelligence (AI), khususnya pada bidang computer vision, memungkinkan sistem mengenali kondisi pengguna secara otomatis melalui analisis citra digital sehingga membuka peluang penerapan pada berbagai aplikasi berbasis perangkat bergerak. Teknologi ini telah banyak dimanfaatkan pada berbagai aplikasi, seperti deteksi objek, pengenalan wajah, sistem keamanan, hingga deteksi tingkat kantuk (*driver drowsiness detection*) berbasis *deep learning*. Berbagai penelitian menunjukkan bahwa penerapan model pembelajaran mendalam mampu menghasilkan tingkat akurasi yang tinggi dalam mengenali kondisi mata terbuka dan mata tertutup secara *real-time*, sehingga

berpotensi diterapkan pada berbagai perangkat bergerak dengan keterbatasan sumber daya komputasi [1], [2], [3].

Seiring meningkatnya penggunaan telepon pintar, aplikasi alarm menjadi salah satu fitur yang paling sering dimanfaatkan untuk membantu pengguna bangun pada waktu yang telah ditentukan. Meskipun demikian, sebagian besar aplikasi tersebut masih berfokus pada pemberian notifikasi tanpa memverifikasi kondisi kesadaran pengguna. Penelitian mengenai *sleep inertia* menjelaskan bahwa seseorang masih dapat mengalami penurunan kewaspadaan, perhatian, serta kemampuan berpikir beberapa saat setelah terbangun, sehingga pengguna sering kali mematikan alarm secara refleks tanpa kesadaran penuh dan kembali tidur (*oversleeping*) [4], [1]. Kondisi tersebut menjadi salah satu

penyebab keterlambatan bangun yang masih sering dialami oleh mahasiswa maupun pekerja.

Sejumlah penelitian terkini berfokus pada pengembangan sistem deteksi kantuk berbasis *computer vision* dan *Deep Learning*. Hassan dkk. [1] mengembangkan sistem deteksi kantuk berbasis arsitektur transformer yang mampu mengenali kondisi wajah pengemudi secara *real-time*. Rajput dkk. [5] dan Pahariya dkk. [6] menerapkan *Transfer Learning* menggunakan arsitektur MobileNetV2 untuk menghasilkan model yang ringan dengan performa klasifikasi yang baik sehingga sesuai diimplementasikan pada perangkat bergerak. Sejalan dengan penelitian tersebut, Indra dkk. [7] juga menunjukkan bahwa arsitektur MobileNet efektif digunakan untuk berbagai tugas klasifikasi citra karena mampu memberikan keseimbangan antara akurasi dan efisiensi komputasi pada perangkat dengan sumber daya terbatas. Salem dkk. [2] menunjukkan bahwa kombinasi *Convolutional Neural Network* (CNN) dan *transfer learning* mampu meningkatkan akurasi deteksi kondisi mata, sedangkan Maharani dkk. [8] membuktikan bahwa deteksi kondisi mata secara *real-time* juga dapat diterapkan dengan pendekatan *machine learning* pada lingkungan nyata. Selain itu, Fonseca dkk. [3] melalui kajian sistematis menyimpulkan bahwa pemanfaatan teknologi *deep learning* menjadi pendekatan yang paling dominan dan efektif dalam penelitian deteksi kantuk beberapa tahun terakhir.

Selain pemilihan arsitektur model, kondisi pencahayaan juga menjadi salah satu tantangan utama dalam pengembangan sistem deteksi mata berbasis *Computer Vision*. Perubahan intensitas cahaya dapat memengaruhi kualitas citra yang diterima kamera sehingga berdampak pada akurasi proses klasifikasi. Oleh karena itu, diperlukan model yang tetap mampu bekerja secara andal pada berbagai kondisi lingkungan, termasuk pada pencahayaan rendah [9].

Meskipun demikian, sebagian besar penelitian tersebut masih berfokus pada deteksi kantuk pengemudi sebagai sistem peringatan keselamatan berkendara. Sistem umumnya hanya memberikan peringatan ketika kondisi kantuk terdeteksi tanpa memverifikasi apakah pengguna benar-benar telah mencapai kondisi sadar. Selain itu, penelitian terdahulu umumnya hanya mengandalkan hasil klasifikasi kondisi mata sebagai dasar pengambilan keputusan dan belum mengintegrasikan mekanisme stabilisasi prediksi maupun verifikasi kognitif setelah proses deteksi selesai [5], [3]. Padahal, berdasarkan fenomena *sleep inertia*, membuka mata belum tentu menunjukkan bahwa seseorang telah kembali memiliki tingkat kewaspadaan yang memadai untuk memulai aktivitas [4], [10].

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan *WakeUp AI*, yaitu aplikasi alarm cerdas berbasis Android yang memanfaatkan teknologi *Computer Vision* dan *Artificial Intelligence* untuk memverifikasi kondisi kesadaran pengguna sebelum alarm dapat dinonaktifkan. Sistem mengimplementasikan *Transfer Learning* menggunakan arsitektur MobileNetV2 yang dijalankan secara lokal melalui *TensorFlow Lite*, sehingga proses inferensi dapat dilakukan secara *offline* pada perangkat Android.

Berbeda dengan penelitian sebelumnya, sistem yang diusulkan mengombinasikan *Eye Region Cropping*, *Rolling Average*, *Leaky Bucket Accumulator*, serta *Random Question Challenge* sebagai mekanisme verifikasi visual dan kognitif. Kombinasi tersebut dirancang untuk meningkatkan kestabilan proses deteksi sekaligus memastikan bahwa pengguna tidak hanya membuka mata, tetapi juga mampu memberikan respons terhadap pertanyaan yang ditampilkan sebelum alarm dihentikan.

Penelitian ini bertujuan mengembangkan aplikasi alarm cerdas berbasis Android yang mampu memverifikasi tingkat kesadaran pengguna melalui integrasi *Computer Vision*, *MobileNetV2*, dan *TensorFlow Lite*. Hasil penelitian diharapkan dapat memberikan alternatif solusi terhadap keterbatasan alarm konvensional sekaligus menunjukkan bahwa penerapan model *deep learning* ringan yang dipadukan dengan mekanisme verifikasi kognitif dapat diimplementasikan secara efektif pada perangkat bergerak.

Tabel 1. Perbandingan Penelitian Terdahulu

Penulis	Metode	Fokus	Kelemahan
Hassan dkk. [1]	Transformer	Deteksi kantuk	Tidak ada verifikasi kesadaran
Salem dkk. [2]	CNN + TL	Klasifikasi mata	Tidak ada verifikasi pengguna
Rajput dkk. [5]	MobileNetV2	Deteksi kantuk	Hanya tahap deteksi
Pahariya dkk. [6]	MobileNetV2	Deteksi kantuk	Tidak ada verifikasi kognitif
Penelitian ini	MobileNetV2 + TFLite	Alarm cerdas	Verifikasi dua tahap

Berdasarkan Tabel 1, sebagian besar penelitian terdahulu berfokus pada pengembangan sistem deteksi kantuk menggunakan berbagai pendekatan *deep learning*, seperti Transformer, CNN, dan MobileNetV2. Meskipun metode tersebut mampu menghasilkan performa klasifikasi yang baik, sebagian besar penelitian masih berhenti pada tahap deteksi kondisi mata atau tingkat kantuk tanpa dilengkapi mekanisme untuk memverifikasi kondisi kesadaran pengguna setelah proses deteksi selesai. Oleh karena itu, penelitian ini mengusulkan aplikasi alarm cerdas berbasis MobileNetV2 dan TensorFlow Lite yang mengintegrasikan deteksi kondisi mata dengan *Random Question Challenge* sebagai mekanisme verifikasi kognitif sebelum alarm dapat dinonaktifkan. Pendekatan tersebut menjadi pembeda utama sekaligus memberikan kontribusi terhadap pengembangan aplikasi alarm yang tidak hanya mampu mendeteksi kondisi mata, tetapi juga membantu memastikan bahwa pengguna benar-benar telah berada pada kondisi sadar.

2. METODE

Penelitian ini menggunakan metode *Research and Development* (R&D) yang berfokus pada perancangan dan pengembangan aplikasi alarm cerdas berbasis Android dengan memanfaatkan teknologi *Artificial Intelligence*. Sistem dikembangkan menggunakan pendekatan *Transfer Learning* berbasis MobileNetV2

untuk mengklasifikasikan kondisi mata pengguna menjadi dua kelas, yaitu mata terbuka (*open eyes*) dan mata tertutup (*closed eyes*). Seluruh proses inferensi dijalankan secara lokal (*on-device inference*) menggunakan *TensorFlow Lite* sehingga aplikasi dapat digunakan tanpa memerlukan koneksi internet.

Tahapan penelitian terdiri atas proses pengumpulan dataset, pelatihan model CNN, prapemrosesan citra, implementasi inferensi pada perangkat Android, stabilisasi hasil deteksi, hingga verifikasi kognitif menggunakan tantangan *Random Question Challenge*. Alur penelitian ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

2.1. Akuisisi Dataset dan Pelatihan Model CNN

Dataset yang digunakan pada penelitian ini merupakan *self-captured eye image dataset* yang dikumpulkan secara mandiri memanfaatkan kamera depan telepon pintar. Pengambilan data dilakukan pada berbagai kondisi pencahayaan, posisi wajah, serta sudut pengambilan gambar untuk meningkatkan kemampuan generalisasi model pada kondisi penggunaan nyata.

Dataset dibagi menjadi dua kelas utama, yaitu mata terbuka (Open Eyes) sebanyak 705 citra dan mata tertutup (Closed Eyes) sebanyak 791 citra. Pelatihan model dilakukan menggunakan TensorFlow dengan resolusi citra sebesar 224×224 piksel dalam format RGB. Setelah proses pelatihan selesai, model dikonversi ke format *TensorFlow Lite* (.tflite) sehingga dapat dijalankan secara *offline* pada perangkat Android [2], [11].

Model klasifikasi kondisi mata dikembangkan menggunakan pendekatan Transfer Learning berbasis arsitektur MobileNetV2.

Arsitektur ini dipilih karena memiliki jumlah parameter yang relatif kecil, efisiensi komputasi yang tinggi, serta tetap mampu mempertahankan performa klasifikasi sehingga sesuai diterapkan pada perangkat bergerak (*mobile device*) [5], [6].

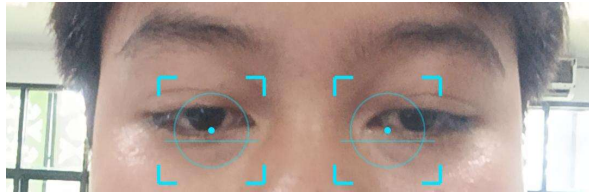
2.2. Prapemrosesan Citra Menggunakan Eye Region Cropping

Sebelum proses inferensi dilakukan, citra hasil tangkapan kamera diproses menggunakan teknik Eye Region Cropping untuk memfokuskan proses klasifikasi pada area mata. Pendekatan ini bertujuan mengurangi pengaruh objek lain, seperti latar belakang, rambut, maupun pakaian, sehingga model memperoleh masukan yang lebih relevan terhadap kondisi mata pengguna [12], [13].

Posisi area mata ditentukan berdasarkan koordinat relatif terhadap tampilan kamera. Area mata kiri berada pada koordinat sekitar 35% lebar layar dan 40% tinggi layar, sedangkan mata kanan berada pada koordinat 65% lebar layar dan 40% tinggi layar. Setelah koordinat

diperoleh, sistem melakukan proses *cropping* terhadap kedua area mata secara terpisah dengan ukuran sekitar 9% dari lebar tampilan kamera.

Selanjutnya hasil *cropping* diubah ukurannya menjadi 224×224 piksel agar sesuai dengan ukuran masukan model MobileNetV2.



Gambar 2. Proses Eye Region Cropping

2.3. Normalisasi Data dan Penyusunan Tensor

Setelah proses perubahan ukuran citra selesai, setiap nilai piksel RGB dinormalisasi ke rentang -1 hingga 1 . Normalisasi bertujuan meningkatkan kestabilan proses inferensi serta mempercepat konvergensi model.

Persamaan normalisasi yang digunakan adalah:

$$(1)$$

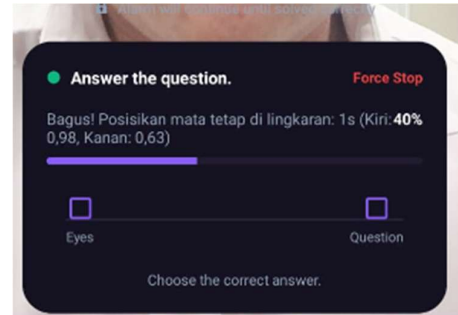
Seluruh data hasil normalisasi kemudian dimasukkan ke dalam struktur *ByteBuffer* sehingga dapat digunakan sebagai tensor masukan bagi TensorFlow Lite.

2.4. Implementasi Inferensi Berbasis TensorFlow Lite

Model MobileNetV2 yang telah selesai dilatih terlebih dahulu dikonversi ke format *TensorFlow Lite* (.tflite) agar dapat dijalankan secara efisien pada perangkat Android. TensorFlow Lite merupakan framework yang dioptimalkan untuk perangkat *mobile* dan *edge device*, sehingga mampu menjalankan proses inferensi secara lokal (*on-device inference*) dengan kebutuhan memori yang rendah dan waktu komputasi yang cepat tanpa memerlukan koneksi internet [14]. Selain itu, TensorFlow Lite menyediakan *runtime* yang mengeksekusi model menggunakan *input tensor* dan menghasilkan *output tensor* sebagai hasil inferensi, sehingga proses klasifikasi dapat dilakukan secara langsung pada perangkat Android dengan latensi yang rendah [14].

Pada aplikasi yang dikembangkan, setiap frame dari kamera depan menghasilkan dua citra hasil *cropping*, yaitu area mata kiri dan mata kanan. Kedua citra tersebut terlebih dahulu dipraproses melalui proses *resize*, normalisasi, dan penyusunan data ke dalam *ByteBuffer* sebagai *input tensor*. Selanjutnya, *TensorFlow Lite Interpreter* mengeksekusi model MobileNetV2 dan menghasilkan

probabilitas untuk masing-masing kelas, yaitu mata terbuka dan mata tertutup [14]. Keputusan klasifikasi ditentukan menggunakan nilai ambang (*threshold*) sebesar 0,50. Apabila probabilitas kelas *Open Eyes* melebihi nilai ambang tersebut maka sistem mengklasifikasikan kondisi mata sebagai mata terbuka, sedangkan apabila berada di bawah nilai ambang maka mata diklasifikasikan sebagai mata tertutup.



Gambar 3. Proses Inferensi TensorFlow Lite

2.5. Algoritma Stabilisasi Deteksi

Untuk mengurangi fluktuasi prediksi akibat kedipan mata maupun perubahan pencahayaan, sistem menerapkan dua tahap stabilisasi, yaitu *Rolling Average* dan *Leaky Bucket*.

$$x' = \frac{x - 127.5}{127.5} \text{ Accumulator.}$$

2.5.1. Rolling Average

Rolling Average digunakan untuk menghitung nilai rata-rata dari tiga prediksi terakhir sehingga perubahan nilai probabilitas menjadi lebih stabil.

$$RA = \frac{x_1 + x_2 + x_3}{3} \quad (2)$$

2.5.2. Leaky Bucket Accumulator

Hasil Rolling Average kemudian diproses dengan algoritma *Leaky Bucket*. Nilai akumulator diperbarui setiap 250 ms. Ketika kedua mata terdeteksi terbuka maka nilai akumulator bertambah, sedangkan apabila salah satu mata tertutup maka nilai tersebut berkurang secara bertahap. Verifikasi dinyatakan berhasil apabila nilai akumulator telah mencapai 3000 ms atau sekitar 3 detik, sehingga sistem tidak mudah tertipu oleh kedipan mata sesaat.

2.6. Random Question Challenge.

Setelah proses verifikasi kondisi mata berhasil, sistem mengaktifkan *Random Question Challenge* sebagai tahap verifikasi kognitif. Pengguna diminta menjawab satu pertanyaan acak menggunakan tombol pilihan jawaban sebelum alarm dapat dihentikan. Tahapan ini dirancang untuk memastikan bahwa pengguna tidak hanya

membuka mata, tetapi juga memberikan respons terhadap pertanyaan yang ditampilkan sehingga diharapkan telah mencapai tingkat kesadaran yang lebih baik. Pendekatan tersebut didasarkan pada penelitian mengenai *sleep inertia* yang menunjukkan bahwa seseorang masih dapat mengalami penurunan kewaspadaan dan fungsi kognitif sesaat setelah bangun tidur [4], [10].

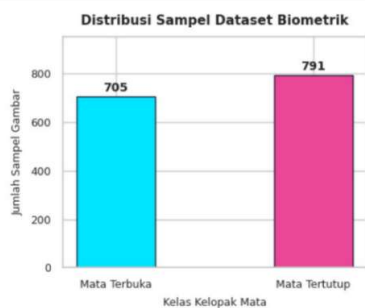
Pengguna diwajibkan memilih jawaban yang benar agar alarm dapat dihentikan sepenuhnya. Apabila jawaban benar, sistem akan menghentikan alarm, mengaktifkan layanan *Text-to-Speech (TTS)* sebagai ucapan selamat pagi, serta menyimpan riwayat penggunaan ke dalam *Room Database*. Tahapan ini bertujuan memastikan bahwa pengguna tidak hanya membuka mata, tetapi juga berada dalam kondisi sadar secara kognitif sebelum alarm dinonaktifkan.

3. HASIL

Penelitian ini menghasilkan sebuah aplikasi WakeUp AI berbasis Android yang mengintegrasikan teknologi *Computer Vision*, *Deep Learning*, dan *Artificial Intelligence* untuk memverifikasi kondisi kesadaran pengguna sebelum alarm dapat dihentikan. Implementasi sistem terdiri atas model klasifikasi kondisi mata menggunakan MobileNetV2, proses inferensi menggunakan TensorFlow Lite, algoritma stabilisasi deteksi, serta verifikasi kognitif melalui soal *Random Question Challenge* sederhana.

3.1. Hasil Pelatihan Model

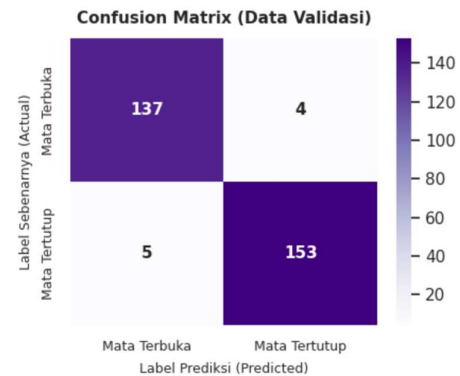
Proses pelatihan dilakukan menggunakan dataset citra mata yang terdiri atas dua kelas, yaitu Open Eyes dan Closed Eyes. Model dibangun menggunakan pendekatan Transfer Learning dengan arsitektur MobileNetV2. Seluruh citra diubah menjadi ukuran 224×224 piksel agar sesuai dengan ukuran masukan model. Selama proses pelatihan, kinerja model dievaluasi melalui metrik accuracy dan loss pada data pelatihan maupun data validasi, Distribusi dataset yang digunakan ditunjukkan pada Gambar 4.



Gambar 4. Distribusi Sampel Dataset

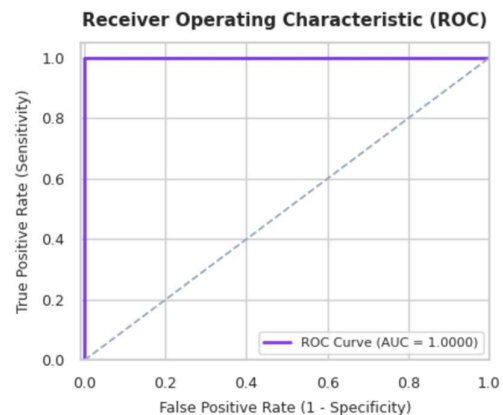
Berdasarkan Gambar 4 dataset terdiri atas 705 citra mata terbuka dan 791 citra mata tertutup sehingga total dataset yang digunakan sebanyak 1.496 citra. Jumlah data pada kedua kelas relatif seimbang meskipun kelas mata tertutup memiliki jumlah citra sedikit lebih

banyak. Distribusi yang cukup seimbang ini membantu model mempelajari karakteristik masing-masing kelas secara lebih optimal dan mengurangi kemungkinan bias selama proses pelatihan.



Gambar 5. Confusion Matrix pada Data Validasi

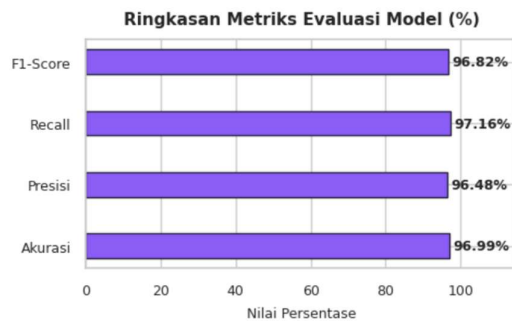
Pada Gambar 5 menunjukkan hasil evaluasi model memanfaatkan confusion matrix pada data validasi. Implementasi yang dilakukan menghasilkan, model berhasil mengklasifikasikan 137 citra mata terbuka dan 153 citra mata tertutup dengan benar. Selain itu, terdapat 4 citra mata terbuka yang salah diklasifikasikan sebagai mata tertutup (*False Negative*) dan 5 citra mata tertutup yang salah diklasifikasikan sebagai mata terbuka (*False Positive*). Jumlah kesalahan klasifikasi yang relatif kecil menunjukkan bahwa model MobileNetV2 mampu membedakan kedua kelas dengan baik sehingga layak diterapkan pada aplikasi WakeUp AI sebagai proses verifikasi kondisi mata.



Gambar 6. Kurva ROC Model MobileNetV2

Gambar 6 memperlihatkan kurva Receiver Operating Characteristic (ROC) yang digunakan untuk mengevaluasi kemampuan model dalam membedakan kelas mata terbuka dan mata tertutup. Pengujian menunjukkan pada data validasi, diperoleh nilai Area Under Curve (AUC) sebesar 1,000, yang menunjukkan bahwa model memiliki kemampuan diskriminasi yang sangat baik dalam membedakan kedua kelas pada data yang digunakan selama proses evaluasi. Kurva ROC yang berada jauh di atas garis diagonal

mengindikasikan bahwa model mampu menghasilkan True Positive Rate (TPR) yang tinggi dengan False Positive Rate (FPR) yang sangat rendah. Meskipun demikian, nilai AUC yang sangat tinggi ini merepresentasikan performa model pada data validasi penelitian dan tidak serta-merta menunjukkan bahwa model akan menghasilkan performa yang sama pada seluruh kondisi penggunaan di dunia nyata. Oleh karena itu, pengujian lebih lanjut pada dataset yang lebih beragam maupun pada perangkat Android dengan kondisi lingkungan yang berbeda tetap diperlukan untuk mengevaluasi kemampuan generalisasi model.



Gambar 7. Ringkasan Metrik Evaluasi Model

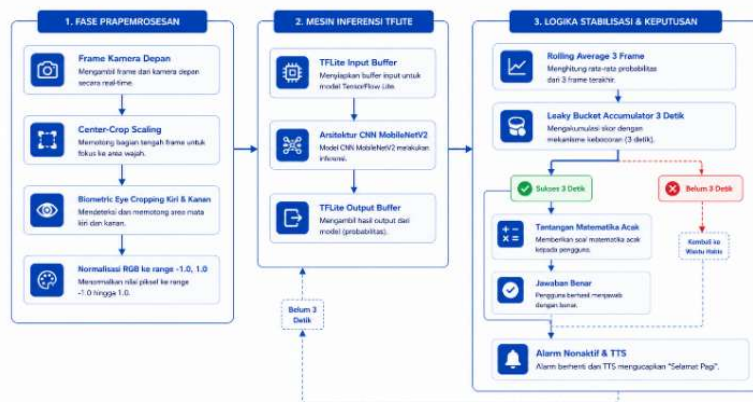
Berdasarkan Gambar 7 model menghasilkan akurasi sebesar 96,99%, presisi sebesar 96,48%, recall sebesar 97,16%, serta F1-score sebesar 96,82%. Nilai akurasi yang tinggi menunjukkan bahwa model mampu mengklasifikasikan kondisi mata terbuka dan

mata tertutup dengan baik. Selain itu, nilai precision sebesar 96,48% mengindikasikan bahwa sebagian besar prediksi yang dihasilkan model sesuai dengan kondisi sebenarnya sehingga jumlah *false positive* relatif rendah. Sementara itu, recall sebesar 97,16% menunjukkan bahwa model berhasil mengenali sebagian besar sampel pada masing-masing kelas sehingga kemungkinan terjadinya *false negative* dapat diminimalkan. Kombinasi nilai precision dan recall tersebut menghasilkan F1-score sebesar 96,82%, yang menunjukkan keseimbangan performa model dalam proses klasifikasi. Hasil evaluasi ini memperlihatkan bahwa model MobileNetV2 memiliki performa yang baik untuk diimplementasikan pada aplikasi WakeUp AI sebagai sistem verifikasi kondisi mata secara *real-time*.

3.2. Implementasi Aplikasi Android

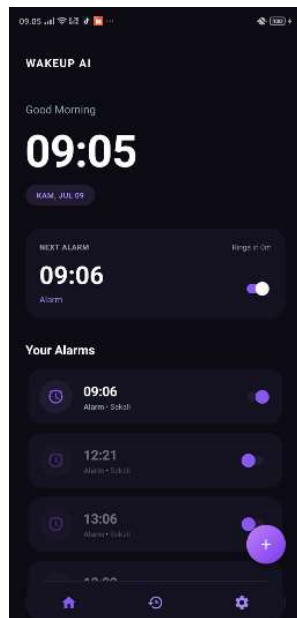
Aplikasi WakeUp AI dikembangkan menggunakan Android Studio dengan bahasa pemrograman Java. Antarmuka aplikasi terdiri atas beberapa halaman utama, yaitu halaman alarm, halaman verifikasi kondisi mata, halaman tantangan *Random Question Challenge*, dan halaman riwayat penggunaan.

Ketika alarm berbunyi, kamera depan perangkat akan aktif secara otomatis. Sistem kemudian melakukan proses *Eye Region Cropping* dan klasifikasi kondisi mata menggunakan model MobileNetV2. Seluruh proses inferensi dilakukan secara lokal sehingga tidak memerlukan koneksi internet.



Gambar 8. Workflow Sistem WakeUp AI

Gambar 9, Gambar 10, dan Gambar 11 memperlihatkan implementasi antarmuka utama aplikasi WakeUp AI yang telah dikembangkan pada perangkat Android. Setiap tampilan dirancang untuk mendukung proses verifikasi pengguna secara bertahap, mulai dari pengaturan alarm, deteksi kondisi mata memanfaatkan kamera depan, hingga penyelesaian tantangan kognitif sebelum alarm dapat dinonaktifkan.



Gambar 9. Tampilan Halaman Alarm

Pada Gambar 9 menunjukkan halaman utama aplikasi yang berfungsi sebagai pusat pengelolaan alarm. Pada halaman ini, pengguna dapat menambahkan alarm baru, mengaktifkan atau menonaktifkan alarm, serta melihat daftar alarm yang telah dibuat. Selain itu, halaman utama juga menampilkan informasi waktu saat ini dan alarm berikutnya yang akan dijalankan.



Gambar 10. Proses Deteksi Mata

Pada Gambar 10 menampilkan proses verifikasi kondisi mata menggunakan kamera depan perangkat. Ketika alarm berbunyi, aplikasi secara otomatis mengaktifkan kamera dan menampilkan penanda (*guide*) pada area mata sebagai acuan posisi pengguna.

Selanjutnya sistem melakukan proses *Eye Region Cropping*, normalisasi citra, serta klasifikasi kondisi mata dengan model MobileNetV2 yang dijalankan melalui TensorFlow Lite. Hasil klasifikasi kemudian diproses melalui algoritma Rolling Average dan Leaky Bucket Accumulator untuk memastikan kedua mata benar-benar terbuka secara stabil selama minimal tiga detik sebelum proses dilanjutkan.



Gambar 11. Halaman Random Question

Pada Gambar 11 menunjukkan halaman Random Question Challenge yang berfungsi sebagai tahap verifikasi kognitif. Setelah proses verifikasi visual berhasil diselesaikan, sistem akan menampilkan satu pertanyaan yang dipilih secara acak beserta beberapa pilihan jawaban dalam bentuk tombol (*button option*). Pengguna diwajibkan memilih jawaban yang benar agar alarm dapat dihentikan sepenuhnya. Pendekatan ini dirancang untuk memastikan bahwa pengguna tidak hanya membuka mata, tetapi juga telah melakukan aktivitas berpikir sederhana sehingga berada dalam kondisi yang lebih sadar sebelum mematikan alarm. Secara keseluruhan, implementasi antarmuka tersebut menunjukkan bahwa aplikasi telah berhasil mengintegrasikan proses verifikasi visual berbasis Computer Vision dengan verifikasi kognitif dalam satu alur kerja yang terstruktur. Seluruh proses berlangsung secara real-time pada perangkat Android tanpa memerlukan koneksi internet karena proses inferensi model dilakukan secara lokal menggunakan TensorFlow Lite.

3.3. Hasil Pengujian Fungsional

Pengujian dilakukan menggunakan metode *Black Box Testing* untuk memastikan seluruh fungsi aplikasi berjalan sesuai dengan kebutuhan sistem. Setiap fitur diuji berdasarkan masukan pengguna dan keluaran yang diharapkan.

Tabel 2. Hasil Pengujian Fungsional

No	Fitur	Hasil yang Diharapkan	Status
1	Alarm aktif	Alarm berbunyi sesuai waktu	✓
2	Kamera aktif	Kamera depan terbuka otomatis	✓
3	Deteksi mata	Mata terbuka dikenali dengan benar	✓
4	Rolling Average	Prediksi lebih stabil	✓
5	Leaky Bucket	Kedipan singkat tidak mengulang proses	✓
6	Random Question Challenge	Alarm berhenti setelah jawaban benar	✓
7	Text-to-Speech	Ucapan selamat pagi diputar	✓
8	Room Database	Riwayat tersimpan	✓

Hasil pengujian pada data Tabel 2, seluruh fitur utama aplikasi berhasil dijalankan sesuai dengan rancangan sistem. Proses aktivasi alarm, pembukaan kamera, deteksi kondisi mata, penerapan Rolling Average, Leaky Bucket Accumulator, Random Question Challenge, Text-to-Speech, serta penyimpanan riwayat menggunakan Room Database berjalan tanpa ditemukan kegagalan fungsi selama pengujian.

3.4. Pengujian pada Berbagai Kondisi

Pengujian juga dilakukan pada beberapa kondisi penggunaan untuk mengetahui kemampuan sistem dalam mengenali kondisi mata.

Tabel 3. Hasil Pengujian Berbagai Kondisi

No	Kondisi Pengujian	Hasil Deteksi	Status
1	Pencahayaannya terang	Mata berhasil dideteksi	✓
2	Pencahayaannya redup	Mata berhasil dideteksi	✓
3	Posisi wajah lurus	Mata berhasil dideteksi	✓
4	Sedikit menoleh ($\pm 15^\circ$)	Mata berhasil dideteksi	✓
5	Kedipan sesaat	Verifikasi tidak diulang	✓
6	Mata terbuka selama 3 detik	Verifikasi berhasil	✓

Pada Tabel 3, hasil pengujian menunjukkan bahwa sistem masih mampu mengenali kondisi mata pada beberapa variasi pencahayaan dan posisi wajah. Namun demikian, pada kondisi pencahayaan yang sangat minim, performa deteksi cenderung menurun karena kualitas citra yang diterima kamera menjadi kurang optimal.

4. PEMBAHASAN

Evaluasi model memperlihatkan bahwa sistem berhasil memperoleh akurasi sebesar 96,99%, *precision* sebesar 96,48%, *recall* sebesar

97,16%, serta *F1-score* sebesar 96,82%. Nilai tersebut menunjukkan bahwa model mampu melakukan klasifikasi kondisi mata terbuka dan mata tertutup dengan tingkat ketepatan yang tinggi. Tingginya nilai *recall* menunjukkan bahwa sebagian besar kondisi mata berhasil dikenali dengan benar, sedangkan nilai *precision* yang tinggi mengindikasikan rendahnya jumlah prediksi yang keliru terhadap kelas yang diprediksi. Selain itu, nilai AUC sebesar 1,000 pada data validasi menunjukkan bahwa model memiliki kemampuan diskriminasi yang sangat baik dalam membedakan kedua kelas selama proses evaluasi. Performa tersebut menunjukkan bahwa pendekatan *Transfer Learning* menggunakan arsitektur MobileNetV2 mampu menghasilkan model yang ringan tanpa mengorbankan kemampuan klasifikasi sehingga sesuai untuk diterapkan pada perangkat bergerak.

Hasil tersebut menunjukkan bahwa pemilihan MobileNetV2 bukan hanya didasarkan pada efisiensi komputasi, tetapi juga mampu mempertahankan performa klasifikasi yang kompetitif pada implementasi perangkat bergerak. dengan penelitian Rajput dkk. [5] dan Pahariya dkk. [6] yang melaporkan bahwa penggunaan MobileNetV2 mampu menghasilkan model dengan kompleksitas komputasi yang rendah namun tetap mempertahankan performa klasifikasi yang baik pada sistem deteksi kantuk. Demikian pula, penelitian Salem dkk. [2] menunjukkan bahwa kombinasi *Convolutional Neural Network* (CNN) dan *Transfer Learning* mampu meningkatkan kemampuan model dalam mengenali kondisi mata secara *real-time*. Penelitian Indra dkk. [7] juga menunjukkan bahwa arsitektur MobileNet efektif digunakan untuk tugas klasifikasi citra dengan tetap mempertahankan efisiensi komputasi. Temuan-temuan tersebut memperkuat bahwa MobileNetV2 merupakan salah satu arsitektur yang sesuai untuk implementasi sistem klasifikasi berbasis Android karena memiliki ukuran model yang relatif kecil, kebutuhan komputasi yang rendah, serta tetap mampu mempertahankan performa klasifikasi yang tinggi.

Selain itu, implementasi model menggunakan TensorFlow Lite memungkinkan seluruh proses inferensi dijalankan secara lokal pada perangkat Android tanpa memerlukan koneksi internet. Pendekatan ini memberikan keuntungan berupa waktu respons yang cepat, penggunaan sumber daya yang lebih efisien, serta menjaga privasi data pengguna karena seluruh proses klasifikasi dilakukan langsung pada perangkat. Karakteristik tersebut menjadikan aplikasi tetap dapat digunakan secara *offline* dengan latensi yang rendah sehingga sesuai untuk kebutuhan alarm yang memerlukan respons secara langsung ketika waktu alarm tercapai. Hasil tersebut juga sejalan dengan penelitian Bhanja dkk. [11] yang menerapkan model ringan pada sistem deteksi kantuk berbasis perangkat bergerak dan menunjukkan bahwa model *lightweight* memiliki performa yang memadai untuk implementasi *real-time*. Penelitian Hassan dkk. [1] dan Turki dkk. [13] juga menunjukkan bahwa pengembangan model *deep learning* yang efisien menjadi salah satu arah utama penelitian deteksi kantuk modern.

Berbeda dengan sebagian besar penelitian terdahulu yang berfokus pada deteksi kantuk pengemudi, penelitian ini menerapkan konsep

yang berbeda, yaitu verifikasi kesadaran pengguna pada aplikasi alarm. Sebagian besar penelitian sebelumnya hanya menghasilkan peringatan ketika kondisi kantuk terdeteksi [1], [15], [3], sedangkan penelitian ini mengharuskan pengguna menyelesaikan dua tahapan verifikasi sebelum alarm dapat dihentikan. Tahap pertama dilakukan melalui deteksi kondisi mata menggunakan *Computer Vision*, sedangkan tahap kedua dilakukan melalui *Random Question Challenge* sebagai verifikasi kognitif. Pendekatan ini menjadi nilai tambah penelitian karena sistem tidak hanya mendeteksi bahwa mata pengguna telah terbuka, tetapi juga memastikan pengguna mampu memberikan respons sederhana sebelum alarm dinonaktifkan. Dengan demikian, mekanisme penghentian alarm tidak hanya bergantung pada hasil klasifikasi citra, tetapi juga mempertimbangkan kondisi kesadaran pengguna.

Penerapan *Rolling Average* dan *Leaky Bucket Accumulator* juga memberikan kontribusi terhadap kestabilan proses verifikasi. Kedipan mata sesaat atau fluktuasi prediksi tidak langsung menyebabkan proses verifikasi diulang dari awal karena sistem melakukan akumulasi nilai prediksi secara bertahap. Mekanisme tersebut membantu mengurangi pengaruh *noise* pada hasil klasifikasi setiap *frame* sehingga keputusan yang dihasilkan menjadi lebih stabil. Selain itu, proses *Eye Region Cropping* mampu memfokuskan model pada area mata sehingga informasi yang tidak relevan dari latar belakang dapat diminimalkan. Strategi tersebut sejalan dengan penelitian Purba dkk. [12] dan Iswahyudi dkk. [10] yang menunjukkan bahwa pemusatan perhatian model pada area mata dapat meningkatkan efektivitas proses klasifikasi kondisi mata.

Keunggulan utama penelitian ini terletak pada integrasi antara *Computer Vision*, MobileNetV2, TensorFlow Lite, *Rolling Average*, *Leaky Bucket Accumulator*, dan *Random Question Challenge* dalam satu aplikasi Android. Berbeda dengan penelitian terdahulu yang umumnya berhenti pada tahap deteksi kondisi mata, penelitian ini menambahkan mekanisme verifikasi kognitif yang didasarkan pada konsep *sleep inertia*. Penelitian McFarlane dkk. [4] menjelaskan bahwa seseorang masih dapat mengalami penurunan kewaspadaan dan fungsi kognitif beberapa saat setelah bangun tidur meskipun telah membuka mata. Oleh karena itu, penambahan *Random Question Challenge* pada penelitian ini diharapkan mampu membantu memastikan bahwa pengguna telah mencapai kondisi yang lebih sadar sebelum alarm dihentikan. Integrasi beberapa komponen tersebut menjadi kontribusi utama penelitian ini karena menghasilkan aplikasi alarm yang tidak hanya mampu mendeteksi kondisi mata, tetapi juga memverifikasi kesiapan pengguna sebelum alarm dinonaktifkan.

Secara keseluruhan, pendekatan yang diusulkan pada penelitian ini memiliki beberapa keunggulan dibandingkan penelitian terdahulu. Implementasi MobileNetV2 menghasilkan model yang ringan dengan kebutuhan komputasi yang rendah sehingga sesuai untuk dijalankan pada perangkat Android. Integrasi TensorFlow Lite memungkinkan seluruh proses inferensi dilakukan secara lokal tanpa memerlukan koneksi internet, sehingga aplikasi memiliki

waktu respons yang cepat sekaligus menjaga privasi data pengguna. Selain itu, penerapan *Eye Region Cropping*, *Rolling Average*, dan *Leaky Bucket Accumulator* mampu meningkatkan kestabilan proses klasifikasi dengan mengurangi pengaruh fluktuasi prediksi pada setiap *frame*. Penambahan *Random Question Challenge* sebagai verifikasi kognitif juga menjadi pembeda utama penelitian ini karena sistem tidak hanya mendeteksi bahwa mata pengguna telah terbuka, tetapi juga memastikan pengguna telah memberikan respons sebelum alarm dinonaktifkan. Kombinasi seluruh komponen tersebut menghasilkan aplikasi alarm cerdas yang ringan, responsif, serta mampu memverifikasi kondisi kesadaran pengguna secara lebih komprehensif dibandingkan alarm konvensional maupun sebagian besar penelitian terdahulu.

Meskipun demikian, penelitian ini masih memiliki beberapa keterbatasan. Model dilatih menggunakan *dataset* yang dikumpulkan secara mandiri sehingga variasi data masih terbatas pada kondisi tertentu. Selain itu, pengujian aplikasi dilakukan pada perangkat Android dengan spesifikasi tertentu sehingga performa sistem pada perangkat lain belum dievaluasi secara menyeluruh. Penelitian selanjutnya dapat mengembangkan *dataset* yang lebih beragam, melakukan pengujian pada berbagai jenis perangkat Android, serta membandingkan performa MobileNetV2 dengan arsitektur *deep learning* lain, seperti EfficientNet, MobileViT, atau model ringan lainnya untuk memperoleh performa yang lebih optimal.

5. KESIMPULAN

Penelitian ini berhasil mengembangkan dan mengimplementasikan aplikasi alarm cerdas berbasis Android yang memanfaatkan Artificial Intelligence untuk memverifikasi kondisi kesadaran pengguna sebelum alarm dapat dinonaktifkan. Sistem mengimplementasikan Transfer Learning menggunakan arsitektur MobileNetV2 yang dijalankan secara **offline** melalui TensorFlow Lite, sehingga proses deteksi kondisi mata dapat dilakukan secara *real-time* tanpa memerlukan koneksi internet. Berdasarkan hasil evaluasi model, diperoleh akurasi sebesar 96,99%, precision sebesar 96,48%, recall sebesar 97,16%, dan F1-score sebesar 96,82%, yang menunjukkan bahwa model memiliki performa klasifikasi yang baik dalam membedakan kondisi mata terbuka dan mata tertutup.

Hasil pengujian fungsional menunjukkan bahwa seluruh fitur utama aplikasi, meliputi alarm, aktivasi kamera, deteksi kondisi mata, *Rolling Average*, *Leaky Bucket Accumulator*, *Random Question Challenge*, *Text-to-Speech*, dan *Room Database*, berhasil berjalan sesuai dengan rancangan sistem. Kombinasi **Eye Region Cropping**, *Rolling Average*, dan *Leaky Bucket Accumulator* mampu meningkatkan kestabilan proses deteksi, sedangkan *Random Question Challenge* berperan sebagai verifikasi kognitif untuk memastikan pengguna tidak hanya membuka mata, tetapi juga memberikan respons sebelum alarm dihentikan. Dengan demikian, aplikasi yang dikembangkan berpotensi menjadi alternatif alarm yang lebih efektif dibandingkan alarm konvensional dalam membantu pengguna bangun dalam kondisi lebih sadar.

Penelitian selanjutnya dapat difokuskan pada penggunaan dataset yang lebih beragam, pengujian pada lebih banyak tipe perangkat Android dan variasi kondisi lingkungan, serta perbandingan performa MobileNetV2 dengan arsitektur *deep learning* lain, seperti EfficientNet atau MobileViT, untuk meningkatkan kemampuan generalisasi dan kinerja sistem.

DAFTAR PUSTAKA

- [1] O. F. Hassan, A. F. Ibrahim, A. Gomaa, M. A. Makhlof, and B. Hafiz, "Real-time driver drowsiness detection using transformer architectures: a novel deep learning approach," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: [10.1038/s41598-025-02111-x](https://doi.org/10.1038/s41598-025-02111-x).
- [2] D. Salem and M. Waleed, "Drowsiness detection in real-time via convolutional neural networks and transfer learning," *Journal of Engineering and Applied Science*, vol. 71, Dec. 2024, doi: [10.1186/s44147-024-00457-z](https://doi.org/10.1186/s44147-024-00457-z).
- [3] T. Fonseca and S. Ferreira, "Drowsiness Detection in Drivers: A Systematic Review of Deep Learning-Based Models," *Applied Sciences*, vol. 15, no. 16, Aug. 2025, doi: [10.3390/app15169018](https://doi.org/10.3390/app15169018).
- [4] S. J. McFarlane, J. E. Garcia, D. S. Verhagen, and A. G. Dyer, "Alarm Tones, Voice Warnings, and Musical Treatments: A Systematic Review of Auditory Countermeasures for Sleep Inertia in Abrupt and Casual Awakenings," *Clocks & Sleep*, vol. 2, no. 4, pp. 416–433, Dec. 2020, doi: [10.3390/clockssleep2040031](https://doi.org/10.3390/clockssleep2040031).
- [5] I. S. Rajput, S. Tyagi, V. Pandey, A. Upreti, and J. K. Kukreja, "Enhancing Driver Safety with MobileNetV2-Based Transfer Learning for Drowsiness Detection," in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 1239–1248, doi: [10.1016/j.procs.2025.04.079](https://doi.org/10.1016/j.procs.2025.04.079).
- [6] S. Pahariya, P. Vats, and S. Suchitra, "Driver Drowsiness Detection using MobileNetV2 with Transfer Learning Approach," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems, ADICS 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, doi: [10.1109/ADICS58448.2024.10533606](https://doi.org/10.1109/ADICS58448.2024.10533606).
- [7] G. Indra, H. Hamida Saputri, and N. Aulia Aziza, "Penerapan MobileNet untuk Klasifikasi Kondisi Wajah Mengantuk pada Citra Wajah," *Journal of Informatics and Communications Technology*, vol. 7, no. 1, pp. 189–197, 2025, doi: [10.52661/jict.v7i1.413](https://doi.org/10.52661/jict.v7i1.413).
- [8] U. D. Maharani, A. S. Handayani, and L. Lindawati, "Analisis Deteksi Mata Kantuk Di Wajah Pengemudi Menggunakan Support Vector Machine (SVM) Berbasis Citra Real-Time," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 2, pp. 940–949, Sep. 2024, doi: [10.47065/bits.v6i2.5701](https://doi.org/10.47065/bits.v6i2.5701).
- [9] S. Saxena, Angel, M. Khurana, S. Tiwari, and H. K. Shaky, "Low-light driver drowsiness detection for real-time safety assistance using dual attention mechanisms in deep learning model," *Sci. Rep.*, vol. 16, no. 1, Dec. 2026, doi: [10.1038/s41598-026-44442-3](https://doi.org/10.1038/s41598-026-44442-3).
- [10] R. Iswahyudi, A. Fauzi Ikhsan, I. Muhamad, and M. Matin, "Fuse-Teknik Elektro Deteksi Kantuk Menggunakan Convolutional Neural Network (CNN) Berdasarkan Kedipan Mata," *Fuse-Teknik Elektro*, vol. 4, pp. 115–124, Dec. 2024.
- [11] A. Bhanja, D. Parhi, D. Gajendra, K. Sinha, and A. K. Sahoo, "Driver drowsiness shield (DDSH): a real-time driver drowsiness detection system," *ROBOMECH Journal*, vol. 12, no. 1, Dec. 2025, doi: [10.1186/s40648-025-00307-4](https://doi.org/10.1186/s40648-025-00307-4).
- [12] A. M. Purba, F. Mahardika, S. Desmana, and N. Moniroh, "Deteksi Kantuk Pengemudi Berdasarkan Keterbukaan Mata Menggunakan Model Ringan dari Spatiotemporal Pyramidal CNN," *sudo Jurnal Teknik Informatika*, vol. 4, no. 4, pp. 355–366, Jan. 2026, doi: [10.56211/sudo.v4i4.1342](https://doi.org/10.56211/sudo.v4i4.1342).
- [13] A. Turki, O. Kahouli, S. Albadran, M. Ksantini, A. Aloui, and M. Ben Amara, "A sophisticated Drowsiness Detection System via Deep Transfer Learning for real time scenarios," *AIMS Mathematics*, vol. 9, no. 2, pp. 3211–3234, 2024, doi: [10.3934/math.2024156](https://doi.org/10.3934/math.2024156).
- [14] V. H. Sahin, I. Oztel, and G. Yolcu Oztel, "Human Monkeypox Classification from Skin Lesion Images with Deep Pre-trained Network using Mobile Application," *J. Med. Syst.*, vol. 46, no. 11, Nov. 2022, doi: [10.1007/s10916-022-01863-7](https://doi.org/10.1007/s10916-022-01863-7).
- [15] Larasati, "Sistem Deteksi Kantuk Real-Time Berbasis CNN dan Landmark Wajah," *Kohesi: Jurnal Sains dan Teknologi*, vol. 8, May 2025, doi: [10.2238/zk4xn913](https://doi.org/10.2238/zk4xn913).

NOMENKLATUR

- (1) x = nilai piksel asli
 x' = nilai piksel setelah normalisasi
- (2) RA = nilai Rolling Average
 x_1, x_2, x_3 = hasil prediksi tiga frame terakhir

PENULIS



Muhammad Zakfar Sodik
Mahasiswa Program Studi Teknologi Rekayasa Perangkat Lunak, Politeknik Negeri Lampung. Memiliki minat pada bidang pengembangan aplikasi mobile, kecerdasan buatan (Artificial Intelligence), computer vision, Internet of Things (IoT), dan rekayasa perangkat lunak.



Renhat Denil Ramadhan
Mahasiswa Teknologi Rekayasa Perangkat Lunak
Politeknik Negeri Lampung. Aktif mengikuti
berbagai kegiatan akademik dan organisasi,
disiplin, bertanggung jawab



Andika Sandi Prananta
Mahasiswa Program Studi Teknologi Rekayasa
Perangkat Lunak, Politeknik Negeri Lampung.
Memiliki fokus pada pengembangan perangkat
lunak sebagai Backend Developer serta aktif
dalam organisasi kemahasiswaan.



Zidan Alghifari
Mahasiswa Program Studi Teknologi Rekayasa
Perangkat Lunak, Politeknik Negeri Lampung.
Memiliki ketertarikan pada pengembangan
teknologi, pemrograman, dan rekayasa
perangkat lunak serta berkomitmen untuk terus
meningkatkan kompetensi di bidang teknologi informasi.



Dani Rofianto, S.Mat., M.Kom.
Adalah dosen aktif di program studi Teknologi
Rekayasa Perangkat Lunak dan Kepala
Laboratorium RPL, Politeknik Negeri Lampung.
Keahlian utamanya meliputi *Artificial Intelligence*,
Computer Vision, dan *Machine Learning*, serta
aktif meneliti topik inovatif seperti pengembangan model *Deep Learning*.



Khusnatul Amaliah, S.Kom., M.Kom.
Adalah Koordinator Program Studi dan dosen
aktif pada program studi Teknologi Rekayasa
Perangkat Lunak (TRPL) di Politeknik Negeri
Lampung. Bidang keahliannya meliputi
Rekayasa Perangkat Lunak dan Kecerdasan
Buatan (AI), yang diterapkan dalam penelitian, pengabdian
masyarakat, serta metode pembelajaran berbasis proyek (*Project Based Learning*) di kampusnya.