

Analisis Performa *Convolutional Neural Network* untuk Deteksi *Microsleep* Berbasis *Edge AI* Pada ESP32-CAM

Denis Prayogi¹, Hadriansa¹, Kariyamin², Muhammad Rizky Adira¹

¹Program Studi Teknik Informatika, STMIK PPKIA Tarakanita Rahmawati, Indonesia.

²Program Studi Teknologi Informasi, Institut Teknologi dan Bisnis Muhammadiyah Wakatobi, Indonesia.

Artikel Info

Kata Kunci:

CNN;
Deteksi Kantuk;
Edge AI;
ESP32-CAM;
Microsleep.

Keywords:

CNN;
Drowsiness detection;
Edge AI;
ESP32-CAM;
Microsleep.

Riwayat Artikel:

Submitted: 05 Juni 2026

Accepted: 04 Juli 2026

Published: 06 Juli 2026

Abstrak: Kantuk saat berkendara merupakan ancaman keselamatan yang serius, salah satunya melalui kondisi *microsleep* yang terjadi tanpa disadari namun dapat menyebabkan pengemudi kehilangan kendali kendaraan. Penelitian ini bertujuan untuk menganalisis performa model *Convolutional Neural Network* (CNN) ringan dalam mendeteksi kondisi *microsleep* secara real-time pada perangkat *Edge AI* berbasis ESP32-CAM sebagai solusi deteksi kantuk yang terjangkau dan mandiri tanpa ketergantungan pada komputasi eksternal. Model dikembangkan menggunakan platform *Edge Impulse* dengan arsitektur CNN ringan dan dilatih pada 3.872 citra mata serta diuji pada 974 citra dari MRL Eye Dataset. Seluruh citra dikonversi ke format grayscale berukuran 24×24 piksel sebelum proses pelatihan. Dengan konfigurasi 50 epoch, learning rate 0,0005, dan kuantisasi INT8, model mencapai akurasi validasi sebesar 94,5%. Ketika diimplementasikan pada ESP32-CAM, waktu inferensi yang diperoleh sebesar 41 ms per frame dengan konsumsi RAM 7,9 KB dan memori flash 31,2 KB. Evaluasi pada perangkat dilakukan melalui pengujian variasi jarak kamera dan kondisi pencahayaan. Deteksi *microsleep* ditentukan berdasarkan durasi mata tertutup, yaitu kurang dari satu detik tidak terdeteksi, satu hingga dua detik memberikan peringatan awal, dan lebih dari dua detik mengaktifkan alarm. Hasil pengujian menunjukkan sistem memiliki performa yang baik pada kondisi pencahayaan normal, meskipun masih mengalami keterbatasan pada kondisi cahaya ekstrem. Oleh karena itu, validasi lebih lanjut pada kondisi berkendara nyata masih diperlukan sebelum sistem dapat diterapkan secara penuh.

Abstract: Driver drowsiness is a serious safety concern, particularly due to *microsleep*, a condition that occurs unconsciously and may cause drivers to lose control of their vehicles. This study aims to evaluate the performance of a lightweight *Convolutional Neural Network* (CNN) model for real-time *microsleep* detection on an ESP32-CAM-based *Edge AI* device as an affordable and standalone drowsiness detection solution without reliance on external computing resources. The model was developed using the *Edge Impulse* platform with a lightweight CNN architecture and trained on 3,872 eye images, while 974 images from the MRL Eye Dataset were used for testing. All images were converted into 24 × 24-pixel grayscale format prior to the training process. Using a configuration of 50 epochs, a learning rate of 0.0005, and INT8 quantization, the model achieved a validation accuracy of 94.5%. When deployed on the ESP32-CAM, the model achieved an inference time of 41 ms per frame with RAM and flash memory consumption of 7.9 KB and 31.2 KB, respectively. The on-device evaluation was conducted by testing various camera distances and lighting conditions. *Microsleep* detection was determined based on eye-closure duration, where eye closure

lasting less than one second was not detected, durations between one and two seconds triggered an initial warning, and durations exceeding two seconds activated an alarm. The experimental results demonstrate that the proposed system performs well under normal lighting conditions, although its performance remains limited under extreme lighting conditions. Therefore, further validation under real-world driving conditions is required before the system can be fully deployed.

Corresponding Author:

Denis Prayogi

Email: denis@ppkia.ac.id

PENDAHULUAN

Kelelahan dan rasa kantuk saat berkendara merupakan faktor penting yang berkontribusi terhadap terjadinya kecelakaan lalu lintas (Pan et al., 2025). Data Komite Nasional Keselamatan Transportasi menunjukkan bahwa kelelahan pengemudi menjadi salah satu penyebab dominan kecelakaan, terutama pada perjalanan jarak jauh (Maulidin, 2024). Sejalan dengan hal tersebut, laporan Kepolisian Negara Republik Indonesia mencatat sebanyak 138.498 kecelakaan lalu lintas selama periode Januari–22 Desember 2024, dengan sekitar 95% di antaranya dipengaruhi oleh faktor kelalaian manusia (Admin, 2024). Data tersebut menunjukkan bahwa sebagian besar kecelakaan lalu lintas berkaitan dengan faktor manusia, termasuk penurunan konsentrasi akibat kelelahan dan kantuk selama mengemudi (Sugeng & Nizar, 2023). Salah satu kondisi yang paling berisiko adalah microsleep, yaitu episode tidur singkat yang terjadi secara tiba-tiba dan sering kali tidak disadari oleh pengemudi. Meskipun hanya berlangsung selama beberapa detik, microsleep dapat menyebabkan hilangnya kewaspadaan terhadap kondisi jalan sehingga berpotensi meningkatkan risiko kecelakaan secara signifikan (Safarov et al., 2023).

Berbagai metode telah dikembangkan untuk mengidentifikasi kondisi kantuk pada pengemudi, baik melalui pemanfaatan sinyal fisiologis seperti electroencephalogram (EEG) maupun melalui pendekatan berbasis citra menggunakan teknologi computer vision. Pendekatan berbasis sinyal fisiologis umumnya mampu menghasilkan tingkat akurasi yang tinggi, namun memerlukan perangkat tambahan yang relatif kompleks sehingga kurang praktis untuk diterapkan pada kendaraan yang digunakan dalam aktivitas sehari-hari (Ramadhani L. Q. et al., 2021). Oleh karena itu, pendekatan berbasis citra wajah menjadi alternatif yang lebih aplikatif, terutama melalui pengamatan kondisi mata, seperti terbuka atau tertutup. Metode ini bersifat non-invasif, tidak mengganggu pengemudi, serta lebih mudah diintegrasikan ke dalam sistem pemantauan kendaraan (Purba et al., 2026). Seiring dengan perkembangan teknologi deep learning, Convolutional Neural Network (CNN) semakin banyak dimanfaatkan dalam klasifikasi citra karena memiliki kemampuan ekstraksi fitur yang baik, termasuk dalam mendeteksi kondisi mata secara real-time (Iswahyudi et al., 2024).

Beberapa penelitian sebelumnya menunjukkan bahwa pendekatan berbasis CNN mampu memberikan performa yang baik dalam mendeteksi kondisi mata pengemudi. Akbar et al. (2024) melaporkan tingkat akurasi yang tinggi pada klasifikasi mata terbuka dan tertutup menggunakan model deep learning berbasis YOLOv5. Selain itu, pendekatan berbasis deteksi objek menggunakan algoritma YOLOv8 juga menunjukkan performa yang baik dalam mendeteksi kondisi mata secara real-time, namun model tersebut dijalankan pada perangkat dengan kemampuan komputasi yang tinggi (Saputri et al., 2025). Ziliwu et al. (2024) berhasil mengimplementasikan TinyML pada ESP32-CAM dengan hasil yang meyakinkan, tetapi penerapannya masih terbatas pada klasifikasi objek pertanian dan belum ditujukan untuk deteksi kantuk pengemudi. Sementara itu, Alajlan dan Ibrahim (2023) mengembangkan model TinyML berbasis MobileNet untuk deteksi kantuk pada mikrokontroler, namun tidak melaporkan secara menyeluruh penggunaan RAM, kapasitas memori flash, maupun waktu inferensi pada perangkat nyata. Secara umum, sebagian besar penelitian tersebut masih berfokus pada peningkatan akurasi klasifikasi dengan menggunakan model yang relatif besar dan memerlukan sumber daya komputasi yang tinggi. Model yang digunakan umumnya belum menerapkan teknik

kompresi seperti kuantisasi sehingga sulit diimplementasikan secara langsung pada mikrokontroler dengan kapasitas memori yang terbatas. Selain itu, evaluasi terhadap konsumsi RAM, kapasitas penyimpanan flash, dan waktu inferensi pada perangkat embedded masih jarang dilaporkan secara komprehensif, padahal aspek-aspek tersebut merupakan parameter penting dalam menentukan kelayakan implementasi sistem *Edge AI* berbiaya rendah.

Perkembangan konsep *Edge AI* membuka peluang untuk mengimplementasikan model kecerdasan buatan secara langsung pada perangkat embedded tanpa bergantung pada komputasi awan maupun perangkat GPU. Pendekatan ini memungkinkan sistem bekerja secara mandiri dengan latensi rendah, konsumsi daya yang lebih efisien, serta mampu memberikan respons secara real-time (Ngo et al., 2025). Salah satu perangkat yang banyak digunakan dalam implementasi *Edge AI* adalah ESP32-CAM karena memiliki kemampuan akuisisi citra, harga yang relatif terjangkau, serta mendukung implementasi model machine learning berukuran ringan meskipun memiliki keterbatasan sumber daya komputasi (Gaffar et al., 2025). Beberapa penelitian menunjukkan bahwa implementasi CNN pada perangkat *Edge AI* mampu menghasilkan performa yang baik dengan konsumsi daya rendah dan waktu inferensi yang relatif cepat (Fanariotis et al., 2023; Pantho et al., 2021; Donskoy et al., 2025; Ziliwu et al., 2024). Meskipun demikian, penelitian yang secara khusus mengintegrasikan arsitektur custom lightweight CNN, kuantisasi INT8, serta evaluasi komprehensif terhadap penggunaan RAM, memori flash, dan waktu inferensi pada ESP32-CAM untuk kasus deteksi *microsleep* masih sangat terbatas. Sebagian besar penelitian terdahulu hanya melaporkan akurasi klasifikasi tanpa mengevaluasi efisiensi penggunaan sumber daya perangkat, sehingga belum dapat menunjukkan kelayakan implementasi model pada sistem embedded berbiaya rendah untuk aplikasi real-time.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk menganalisis performa model custom lightweight *Convolutional Neural Network* (CNN) dengan kuantisasi INT8 untuk deteksi *microsleep* pada perangkat *Edge AI* berbasis ESP32-CAM. Analisis dilakukan terhadap akurasi klasifikasi, konsumsi RAM, penggunaan memori flash, waktu inferensi, serta kemampuan sistem dalam beroperasi secara real-time pada perangkat dengan sumber daya terbatas. Kontribusi penelitian ini terletak pada pengembangan model CNN ringan yang dioptimalkan untuk implementasi langsung pada ESP32-CAM disertai evaluasi menyeluruh terhadap efisiensi komputasi pada perangkat embedded. Sistem yang dikembangkan diharapkan mampu mendeteksi kondisi mata secara real-time serta memberikan peringatan ketika terjadi indikasi *microsleep* berdasarkan durasi mata tertutup.

METODE

Penelitian ini menggunakan pendekatan eksperimen untuk menganalisis performa *Convolutional Neural Network* (CNN) dalam mendeteksi kondisi *microsleep* pada perangkat *Edge AI* berbasis ESP32-CAM. Sistem dikembangkan untuk melakukan klasifikasi kondisi mata terbuka (*open eye*) dan mata tertutup (*closed eye*) secara real-time menggunakan model CNN yang diimplementasikan pada perangkat embedded. Tahapan penelitian meliputi pengumpulan dataset, prapemrosesan data, pelatihan model CNN, implementasi model pada ESP32-CAM, serta pengujian performa sistem.

Dataset

Dataset yang digunakan merupakan dataset publik, yaitu MRL Eye Dataset, yang berisi citra mata manusia dengan berbagai kondisi pencahayaan, posisi wajah, dan tingkat keterbukaan mata. Pada penelitian ini, dataset dikelompokkan menjadi dua kelas, yaitu mata terbuka (*open eye*) dan mata tertutup (*closed eye*). Seluruh citra kemudian diseleksi sehingga hanya memuat area mata sebagai objek utama klasifikasi, sehingga proses pelatihan model dapat difokuskan pada fitur visual yang berkaitan dengan kondisi mata.

Dataset dibagi menjadi data pelatihan (*training*) dan data pengujian (*testing*) dengan rasio 80:20 menggunakan fitur *train/test split* pada platform *Edge Impulse*. Data pelatihan berjumlah 3.872 citra yang terdiri dari 1.970 citra mata terbuka dan 1.902 citra mata tertutup, sedangkan data pengujian berjumlah 974 citra yang terdiri dari 492 citra mata terbuka dan 482 citra mata tertutup. Data pelatihan

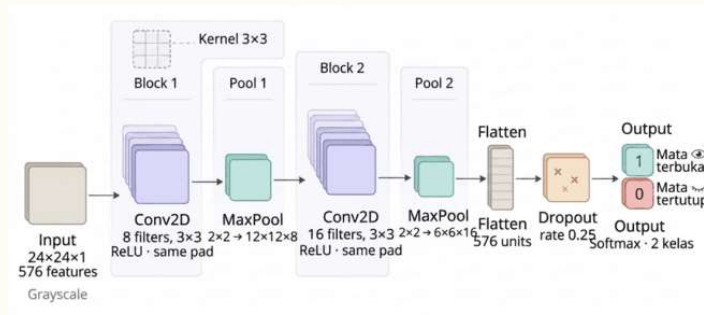
digunakan untuk membangun model CNN, sedangkan data pengujian digunakan untuk mengevaluasi performa model terhadap data yang tidak diikutsertakan dalam proses pelatihan.

Prapemrosesan Data

Tahap prapemrosesan dilakukan untuk mempersiapkan data sebelum proses pelatihan model. Seluruh citra telah tersedia dalam format grayscale dengan ukuran 24×24 piksel sehingga tidak diperlukan proses konversi warna maupun perubahan ukuran citra. Untuk meningkatkan kemampuan generalisasi model terhadap kondisi nyata saat sistem diimplementasikan secara real-time, dilakukan proses augmentasi data secara otomatis menggunakan platform Edge Impulse (Rodiah et al., 2025; Xu et al., 2023). Teknik augmentasi yang digunakan meliputi horizontal flip, perubahan tingkat kecerahan sebesar 15%, zoom sebesar 10%, serta rotasi hingga 10° . Proses augmentasi ini bertujuan meningkatkan variasi data pelatihan sehingga model menjadi lebih robust terhadap perubahan posisi dan pencahayaan.

Arsitektur CNN

Gambar 1 menampilkan arsitektur CNN yang digunakan pada penelitian. Model yang digunakan pada penelitian ini merupakan Convolutional Neural Network (CNN) dengan arsitektur ringan (lightweight CNN) yang dirancang agar dapat diimplementasikan pada perangkat ESP32-CAM yang memiliki keterbatasan sumber daya komputasi. Arsitektur CNN disusun secara kustom dan terdiri atas dua convolution layer yang diikuti oleh lapisan max pooling, satu fully connected layer, satu lapisan dropout, serta lapisan keluaran (output layer) menggunakan fungsi aktivasi Softmax. Convolution layer berfungsi mengekstraksi fitur penting dari citra mata, sedangkan max pooling digunakan untuk mengurangi dimensi data sehingga proses komputasi menjadi lebih efisien. Lapisan dropout diterapkan untuk mengurangi risiko overfitting, sedangkan fungsi aktivasi Softmax digunakan untuk mengklasifikasikan citra ke dalam dua kelas, yaitu mata terbuka dan mata tertutup. Arsitektur CNN yang digunakan ditunjukkan pada Gambar 1.



Gambar 1. Arsitektur CNN yang Digunakan

Total parameter model yang dilatih berjumlah 3.842 parameter. Setelah dilakukan kuantisasi INT8, ukuran model menjadi 31,2 KB sehingga dapat dimuat pada kapasitas memori flash ESP32-CAM. Proses pelatihan model dilakukan menggunakan platform Edge Impulse melalui fitur Image Classification dengan arsitektur kustom yang mengacu pada penelitian Alajlan & Ibrahim (2023). Parameter pelatihan yang digunakan meliputi jumlah epoch, learning rate, dan ukuran batch yang disesuaikan untuk memperoleh performa model yang optimal.

Implementasi pada ESP32-CAM

Setelah proses pelatihan selesai, model CNN diekspor dalam format Arduino Library dan diimplementasikan pada perangkat ESP32-CAM. Perangkat ESP32-CAM berfungsi sebagai sistem utama yang melakukan akuisisi citra secara real-time menggunakan kamera OV2640 bawaan (Nurdiansyah et al., 2024).

Pada proses implementasi, kamera OV2640 mengambil citra secara kontinu dengan resolusi awal 320×240 piksel dalam format JPEG. Selanjutnya citra dikonversi ke format grayscale dan diperkecil secara otomatis menjadi 24×24 piksel menggunakan fungsi crop and interpolate yang tersedia pada

Edge Impulse SDK sebelum diproses oleh model CNN. Implementasi sistem tidak menggunakan mekanisme face detection maupun eye detection secara eksplisit. Area mata diasumsikan berada pada bagian tengah frame kamera yang diarahkan langsung ke wajah pengemudi pada jarak tertentu. Pendekatan ini dipilih untuk mempertahankan kecepatan inferensi sekaligus meminimalkan beban komputasi pada perangkat ESP32-CAM sehingga sistem dapat bekerja secara real-time.

Pengujian Sistem

Pengujian dilakukan untuk menganalisis performa model CNN baik pada tahap klasifikasi maupun implementasi pada perangkat ESP32-CAM. Evaluasi model dilakukan menggunakan parameter precision, recall, loss, serta waktu inferensi (Abdurrahman et al., 2023). Selain itu, pengujian implementasi dilakukan secara real-time dengan memvariasikan kondisi pencahayaan dan jarak wajah terhadap kamera untuk mengetahui kemampuan sistem dalam mendeteksi kondisi mata pada berbagai kondisi penggunaan.

Selama proses pengujian, ESP32-CAM melakukan klasifikasi terhadap setiap citra yang ditangkap kamera menjadi kondisi mata terbuka atau mata tertutup. Indikasi *microsleep* ditentukan berdasarkan durasi mata tertutup secara berturut-turut, yaitu mata tertutup kurang dari satu detik tidak menghasilkan peringatan, durasi satu hingga dua detik menghasilkan peringatan awal, sedangkan durasi lebih dari dua detik mengaktifkan alarm. Hasil pengujian kemudian dianalisis untuk mengevaluasi akurasi klasifikasi, kecepatan inferensi, serta kemampuan sistem dalam mendeteksi *microsleep* secara real-time pada perangkat *Edge AI* dengan sumber daya komputasi terbatas.

HASIL DAN PEMBAHASAN

Hasil Pelatihan Model CNN

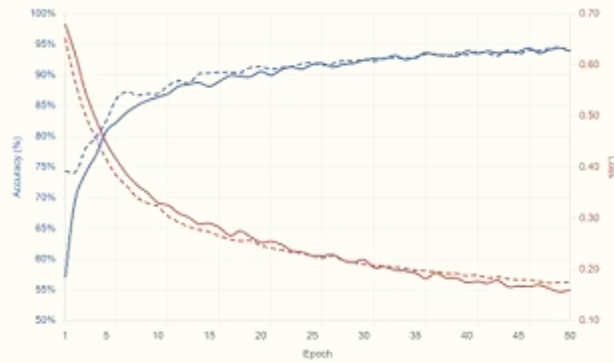
Proses pelatihan model dilakukan menggunakan platform Edge Impulse dengan metode *image classification*. Dataset yang digunakan terdiri dari dua kelas utama, yaitu mata terbuka dan mata tertutup. Sebelum proses pelatihan dilakukan, seluruh citra telah melalui tahap prapemrosesan berupa normalisasi data.

Model CNN dilatih menggunakan beberapa parameter pelatihan seperti jumlah *epoch*, *learning rate*, dan *batch size* (Mahmud et al., 2026) yang disesuaikan dengan kemampuan komputasi perangkat *embedded*. Arsitektur model dirancang agar tetap ringan sehingga dapat dijalankan pada perangkat ESP32-CAM tanpa memerlukan komputasi yang besar.

Tabel 1. Konfigurasi Parameter

Parameter	Keterangan
Citra input	24x24
Epoch	50
Batch Size	32
Learning rate	0.0005
Optimizer	Adam
Aktivasi output	Softmax
Training processor	GPU

Berdasarkan hasil pelatihan, model mampu mencapai tingkat akurasi yang cukup baik pada data validasi. Selain itu, nilai loss mengalami penurunan secara bertahap selama proses pelatihan berlangsung seperti yang terlihat pada Gambar 2. Hal tersebut menunjukkan bahwa model mampu mempelajari pola citra mata terbuka dan mata tertutup dengan cukup baik.



Gambar 2. Grafik Hasil Pelatihan

Hasil pelatihan pada Edge Impulse juga menunjukkan bahwa model memiliki ukuran yang relatif kecil yaitu flash usage sebesar 31.2 KB, dengan pemakaian ram tertinggi saat inferensi sebesar 7.9 KB, serta waktu inferensi tertinggi sekitar 41ms, sehingga memungkinkan untuk diimplementasikan pada perangkat ESP32-CAM. Ukuran model yang ringan menjadi salah satu faktor penting dalam implementasi Edge AI karena perangkat memiliki keterbatasan memori dan kemampuan pemrosesan.

Hasil Pengujian Model

Pengujian model dilakukan menggunakan data uji yang tidak digunakan pada proses pelatihan. Evaluasi dilakukan untuk mengetahui kemampuan model dalam mengklasifikasikan kondisi mata terbuka dan mata tertutup. Tabel 2 menampilkan hasil evaluasi dari parameter yang digunakan.

Tabel 2. Hasil Pengujian Model	
Parameter Evaluasi	Nilai (%)
Akurasi	94.5
Presisi	94
Recall	94
F1-Score	94
Loss	0.17

Berdasarkan hasil pengujian, model mampu melakukan klasifikasi dengan tingkat akurasi yang cukup baik yaitu 94.5%. Nilai presisi dan *recall* menunjukkan bahwa model dapat mengenali pola mata terbuka maupun mata tertutup secara konsisten. Namun, masih terdapat beberapa kesalahan klasifikasi terutama pada kondisi pencahayaan rendah dan citra dengan kualitas kurang baik. Gambar 3 merupakan confusion matrix yang menampilkan hasil pengujian.

	MATA TERBUKA	MATA TERTUTUP
MATA TERBUKA	94.7%	5.3%
MATA TERTUTUP	5.8%	94.2%
F1 SCORE	0.95	0.94

Gambar 3. Confusion Matrix Hasil Pengujian

Berdasarkan Gambar 3, dari 492 citra mata terbuka pada data uji, sebanyak 466 citra (94,7%) berhasil diklasifikasikan dengan benar sebagai mata terbuka, sedangkan 26 citra (5,3%) salah diklasifikasikan sebagai mata tertutup (false positive terhadap kelas mata terbuka). Sementara itu, dari 482 citra mata tertutup, sebanyak 454 citra (94,2%) berhasil dikenali dengan benar, sedangkan 28 citra (5,8%) salah diklasifikasikan sebagai mata terbuka (false negative terhadap kondisi mata tertutup). Dalam konteks deteksi microsleep, kesalahan false negative merupakan kondisi yang lebih kritis karena sistem gagal mengenali mata tertutup sehingga berpotensi tidak memberikan peringatan ketika pengemudi mengalami kantuk. Sebaliknya, false positive menyebabkan sistem memberikan peringatan ketika pengemudi sebenarnya masih terjaga, yang dapat mengurangi kenyamanan pengguna.

Meskipun demikian, tingkat kesalahan klasifikasi pada kedua kelas relatif seimbang, sehingga menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap kedua kategori citra.

Hasil Implementasi pada ESP32-CAM dan Analisis Deteksi *Microsleep*

Setelah proses pelatihan selesai, model CNN diekspor dalam format Arduino Library dan diimplementasikan pada perangkat ESP32-CAM. Pengujian dilakukan secara real-time menggunakan kamera bawaan ESP32-CAM untuk mendeteksi kondisi mata pengguna.

Pada proses pengujian, sistem melakukan pengambilan citra secara kontinu kemudian melakukan klasifikasi kondisi mata menjadi dua kelas, yaitu mata terbuka dan mata tertutup. Hasil klasifikasi ditampilkan melalui serial monitor dan digunakan sebagai dasar dalam menentukan kondisi *microsleep*.

Pada penelitian ini, kondisi *microsleep* ditentukan berdasarkan durasi mata tertutup dalam rentang waktu tertentu. Sistem akan menghitung jumlah frame yang terdeteksi sebagai mata tertutup secara berurutan. Jika mata tertutup melebihi batas waktu yang telah ditentukan, maka sistem akan memberikan peringatan sebagai indikasi terjadinya *microsleep*. Gambar 4 menunjukkan hasil pembacaan deteksi pada serial monitor di Arduino IDE.

```
(START) Deteksi dimulai.

Capture OK - frame 1
Capture OK - frame 2
Capture OK - frame 3
Capture OK - frame 4
[ 5] Mata: TERTUTUP | Open: 0.125 | Closed: 0.875 | Tertutup: 5 frame (0.20s) | Status: NORMAL
Capture OK - frame 5
Capture OK - frame 6
Capture OK - frame 7
Capture OK - frame 8
Capture OK - frame 9
[ 10] Mata: TERTUTUP | Open: 0.117 | Closed: 0.883 | Tertutup: 10 frame (0.41s) | Status: NORMAL
Capture OK - frame 10
Capture OK - frame 11
Capture OK - frame 12

[PERINGATAN] Mata tertutup 1-2 detik - waspada!
[PERINGATAN] Frame tertutup: 24 (~1.0 detik)
[ 24] Mata: TERTUTUP | Open: 0.141 | Closed: 0.859 | Tertutup: 24 frame (0.98s) | Status: PERINGATAN
Capture OK - frame 24
[ 25] Mata: TERTUTUP | Open: 0.117 | Closed: 0.883 | Tertutup: 25 frame (1.02s) | Status: PERINGATAN
Capture OK - frame 25
[ 26] Mata: TERTUTUP | Open: 0.129 | Closed: 0.871 | Tertutup: 26 frame (1.07s) | Status: PERINGATAN
Capture OK - frame 26
[ 27] Mata: TERTUTUP | Open: 0.141 | Closed: 0.859 | Tertutup: 27 frame (1.11s) | Status: PERINGATAN
Capture OK - frame 27
[ 28] Mata: TERTUTUP | Open: 0.137 | Closed: 0.863 | Tertutup: 28 frame (1.15s) | Status: PERINGATAN
Capture OK - frame 28
[ 29] Mata: TERTUTUP | Open: 0.129 | Closed: 0.871 | Tertutup: 29 frame (1.19s) | Status: PERINGATAN

Output Serial Monitor X
Message (Enter to send message to 'AI Thinker ESP32-CAM' on 'COM14') Both NL & CR 115200 baud

⚠ MICROSLEEP TERDETEKSI! ⚠

[ALERT] Mata tertutup: 48 frame (~2.0 detik)
[ALERT] Total kejadian microsleep: 1
[ALERT] Total frame diproses : 48
[ 48] Mata: TERTUTUP | Open: 0.160 | Closed: 0.840 | Tertutup: 48 frame (1.97s) | Status: MICROSLEEP
Capture OK - frame 48
[ 49] Mata: TERTUTUP | Open: 0.168 | Closed: 0.832 | Tertutup: 49 frame (2.01s) | Status: MICROSLEEP
Capture OK - frame 49
[ 50] Mata: TERTUTUP | Open: 0.047 | Closed: 0.953 | Tertutup: 50 frame (2.05s) | Status: MICROSLEEP
Capture OK - frame 50
[ 51] Mata: TERTUTUP | Open: 0.008 | Closed: 0.992 | Tertutup: 51 frame (2.09s) | Status: MICROSLEEP
Capture OK - frame 51
[ 52] Mata: TERTUTUP | Open: 0.004 | Closed: 0.996 | Tertutup: 52 frame (2.13s) | Status: MICROSLEEP
```

Gambar 4. Hasil Deteksi

Gambar 3 merupakan beberapa percobaan yang dilakukan dengan kondisi tertentu untuk mengetahui hasil dari deteksi menggunakan ESP32-CAM. Secara keseluruhan sistem mampu mendeteksi mata tertutup dan terbuka, serta kondisi *microsleep* berdasarkan durasi lamanya mata pengemudi tertutup.

Tabel 3. Pengujian Deteksi *Microsleep*

Durasi Mata Tertutup	Hasil Sistem	Status yang Diharapkan	Keterangan
< 1 detik	Normal	Normal	Benar
1-2 detik	Peringatan awal	Memberikan peringatan awal	Benar
> 2 detik	Microsleep terdeteksi	Peringatan Microsleep terdeteksi	Benar

Berdasarkan Tabel 3, Pendekatan berbasis durasi dipilih karena kondisi mata tertutup sesaat selama kurang dari satu detik belum tentu menunjukkan pengemudi mengalami kantuk. Dengan menggunakan parameter durasi, sistem menjadi lebih stabil dan mampu mengurangi kesalahan deteksi akibat kedipan mata normal.

Tabel 4 merupakan ujicoba dengan kondisi jarak yang berbeda serta variasi pencahayaan normal, redup dan terang, dengan posisi wajah agak miring disesuaikan dengan kondisi yang nyata.

Tabel 4. Pengujian Berdasarkan Jarak dan Cahaya

No	Jarak (cm)	Cahaya	Kondisi	Hasil Deteksi	Keterangan
1	30	Redup	Mata terbuka	Mata terbuka	Benar
2			Mata tertutup	Mata terbuka	Salah
3		Normal	Mata terbuka	Mata terbuka	Benar
4			Mata tertutup	Mata tertutup	Benar
5		Terang	Mata terbuka	Mata terbuka	Benar
6			Mata tertutup	Mata terbuka	Salah
7	40	Redup	Mata terbuka	Mata terbuka	Benar
8			Mata tertutup	Mata terbuka	Salah
9		Normal	Mata terbuka	Mata terbuka	Benar
10			Mata tertutup	Mata tertutup	Benar
11		Terang	Mata terbuka	Mata terbuka	Benar
12			Mata tertutup	Mata tertutup	Benar

Berdasarkan hasil pengujian pada Tabel 4, sistem mampu mendeteksi kondisi mata dengan baik pada jarak 30–40 cm ketika kondisi pencahayaan berada pada tingkat normal. Sebaliknya, pada kondisi pencahayaan redup maupun terlalu terang masih ditemukan kesalahan klasifikasi, terutama ketika mata tertutup terdeteksi sebagai mata terbuka. Hal tersebut menunjukkan bahwa intensitas cahaya berpengaruh terhadap kualitas citra yang diterima kamera ESP32-CAM sehingga memengaruhi proses ekstraksi fitur oleh model CNN. Meskipun demikian, sistem tetap mampu melakukan inferensi secara real-time serta memberikan respons berupa peringatan sesuai durasi mata tertutup yang telah ditentukan. Hasil ini menunjukkan bahwa pendekatan CNN ringan berbasis Edge AI memiliki potensi untuk diterapkan pada sistem deteksi kantuk berbiaya rendah, meskipun masih memerlukan validasi lebih lanjut pada kondisi berkendara yang lebih beragam.

Pembahasan

Hasil penelitian menunjukkan bahwa model Convolutional Neural Network (CNN) kustom dengan kuantisasi INT8 mampu diimplementasikan pada perangkat ESP32-CAM untuk mendeteksi kondisi mata secara real-time. Model yang dikembangkan mencapai akurasi sebesar 94,5% dengan waktu inferensi 41 ms, penggunaan RAM sebesar 7,9 KB, dan ukuran flash sebesar 31,2 KB. Hasil tersebut menunjukkan bahwa model memiliki performa klasifikasi yang baik sekaligus memenuhi keterbatasan sumber daya komputasi pada perangkat embedded. Kemampuan melakukan proses inferensi secara langsung pada ESP32-CAM menunjukkan bahwa pendekatan Edge AI dapat

diterapkan sebagai solusi deteksi *microsleep* tanpa bergantung pada komputasi berbasis cloud maupun perangkat dengan kemampuan komputasi tinggi.

Tabel 5. Pengujian Berdasarkan Jarak dan Cahaya

Penelitian	Metode	Platform	Akurasi (%)	Inferensi	RAM/Flash
(Akbar et al., 2024)	CNN/YOLOv5	PC + ESP32	-	-	-
(Alajlan & Ibrahim, 2023)	MobileNet TinyML DRQ		99.6	-	0.05 MB
(Saputri et al., 2025)	YOLOv8n	PC/GPU	mAP@0.5 97.9	-	-
(Ziliwu et al., 2024)	TinyML ESP32-CAM	ESP32-CAM	66-91	-	-
Penelitian ini	CNN kustom INT8	ESP32-CAM	94.5	41 ms	7,9 KB / 31,2 KB

Jika dibandingkan dengan penelitian terdahulu sebagaimana ditunjukkan pada Tabel 5, model yang dikembangkan pada penelitian ini tidak menghasilkan akurasi tertinggi. Alajlan & Ibrahim (2023) melaporkan akurasi sebesar 99,6% menggunakan MobileNet berbasis TinyML dengan kuantisasi DRQ. Namun, penelitian tersebut tidak melaporkan penggunaan RAM maupun waktu inferensi pada perangkat secara rinci sehingga efisiensi implementasinya pada mikrokontroler sulit dibandingkan secara langsung. Penelitian Akbar et al. (2024) menggunakan pendekatan CNN berbasis YOLOv5 juga menunjukkan hasil yang baik untuk klasifikasi kondisi mata, tetapi tidak melaporkan performa implementasi pada perangkat embedded. Sementara itu, Saputri et al. (2025) memperoleh nilai mAP@0.5 sebesar 97,9% menggunakan YOLOv8n, namun proses inferensi dilakukan pada komputer dengan dukungan GPU yang memiliki kemampuan komputasi jauh lebih tinggi dibandingkan ESP32-CAM. Penelitian Ziliwu et al. (2024) berhasil mengimplementasikan TinyML pada ESP32-CAM dengan akurasi antara 66% hingga 91%, tetapi diterapkan pada kasus klasifikasi objek pertanian. Dibandingkan penelitian tersebut, penelitian ini memberikan kontribusi berupa implementasi model CNN kustom yang dirancang khusus untuk deteksi *microsleep* dengan evaluasi yang lebih komprehensif, meliputi akurasi, waktu inferensi, penggunaan RAM, dan kapasitas memori flash pada perangkat embedded.

Hasil penelitian juga menunjukkan adanya kompromi (trade-off) antara akurasi model dan efisiensi penggunaan sumber daya perangkat. Model deep learning dengan arsitektur yang lebih kompleks umumnya mampu menghasilkan akurasi yang lebih tinggi, tetapi memerlukan kapasitas memori dan daya komputasi yang lebih besar sehingga kurang sesuai untuk diimplementasikan pada ESP32-CAM. Sebaliknya, CNN kustom yang digunakan pada penelitian ini hanya memiliki 3.842 parameter dan setelah dilakukan kuantisasi INT8 menghasilkan ukuran model sebesar 31,2 KB dengan penggunaan RAM inferensi sebesar 7,9 KB. Dengan ukuran model yang relatif kecil, sistem tetap mampu mencapai akurasi sebesar 94,5% dan melakukan inferensi dalam waktu 41 ms. Hasil tersebut menunjukkan bahwa penyederhanaan arsitektur dan kuantisasi INT8 mampu menghasilkan keseimbangan antara performa klasifikasi dan efisiensi penggunaan sumber daya, sehingga model layak diterapkan pada perangkat *Edge AI* berbiaya rendah.

Implementasi model secara langsung pada perangkat *Edge AI* juga memberikan beberapa keuntungan dibandingkan pendekatan berbasis cloud. Seluruh proses klasifikasi dilakukan pada ESP32-CAM sehingga sistem tidak memerlukan koneksi internet maupun pengiriman data ke server eksternal. Pendekatan ini menghasilkan latensi yang lebih rendah, meningkatkan kecepatan respons sistem, serta mengurangi ketergantungan terhadap infrastruktur jaringan. Karakteristik tersebut menjadikan sistem lebih sesuai untuk diterapkan pada aplikasi pemantauan pengemudi yang membutuhkan respons cepat dalam memberikan peringatan ketika terdeteksi indikasi *microsleep*.

Meskipun demikian, hasil pengujian menunjukkan bahwa performa sistem masih dipengaruhi oleh kondisi pencahayaan dan posisi wajah terhadap kamera. Pengujian menunjukkan bahwa sistem

bekerja paling baik pada pencahayaan normal dengan jarak sekitar 30–40 cm, sedangkan pada kondisi pencahayaan redup maupun terlalu terang masih ditemukan kesalahan klasifikasi, terutama ketika mata tertutup terdeteksi sebagai mata terbuka. Dalam konteks keselamatan berkendara, kesalahan tersebut perlu menjadi perhatian karena berpotensi menyebabkan sistem gagal memberikan peringatan ketika pengemudi mengalami *microsleep*. Selain itu, pendekatan yang digunakan belum menerapkan mekanisme *face detection* maupun *eye detection*, sehingga posisi wajah harus berada pada area tertentu agar proses klasifikasi dapat dilakukan secara optimal. Kualitas kamera ESP32-CAM yang relatif terbatas juga memengaruhi detail citra yang diperoleh, terutama pada kondisi pencahayaan yang tidak ideal.

Penelitian ini masih memiliki beberapa keterbatasan yang perlu diperhatikan. Pengujian dilakukan pada kondisi yang masih terkontrol dengan variasi jarak dan pencahayaan yang terbatas sehingga belum sepenuhnya merepresentasikan kondisi berkendara yang sebenarnya. Variasi wajah pengguna, penggunaan kacamata, perubahan posisi kepala, serta pengaruh getaran kendaraan belum dievaluasi dalam penelitian ini. Oleh karena itu, penelitian selanjutnya perlu melakukan validasi pada kondisi berkendara nyata dengan variasi lingkungan yang lebih beragam. Selain itu, penggunaan kamera inframerah, penerapan mekanisme *face detection* atau *eye detection*, serta penambahan dataset dengan variasi kondisi pencahayaan dan karakteristik pengguna yang lebih luas diharapkan dapat meningkatkan keandalan sistem dalam mendeteksi *microsleep* pada aplikasi keselamatan berkendara.

KESIMPULAN

Berdasarkan hasil penelitian, model Convolutional Neural Network (CNN) kustom dengan kuantisasi INT8 berhasil diimplementasikan pada perangkat ESP32-CAM untuk mendeteksi kondisi mata terbuka dan mata tertutup sebagai indikator *microsleep*. Model yang dikembangkan menggunakan arsitektur ringan dengan total 3.842 parameter dan menghasilkan akurasi sebesar 94,5% pada data pengujian. Setelah diterapkan pada perangkat ESP32-CAM, model mampu melakukan inferensi secara real-time dengan waktu inferensi sebesar 41 ms, penggunaan RAM sebesar 7,9 KB, dan ukuran flash sebesar 31,2 KB. Hasil tersebut menunjukkan bahwa pendekatan Edge AI memungkinkan proses deteksi dilakukan secara langsung pada perangkat tanpa memerlukan koneksi internet maupun komputasi berbasis cloud, sehingga berpotensi menjadi solusi deteksi *microsleep* yang ringan dan berbiaya rendah.

Pengujian implementasi pada perangkat menunjukkan bahwa sistem bekerja dengan baik pada kondisi pencahayaan normal dengan jarak pengambilan citra sekitar 30–40 cm. Namun, performa masih mengalami penurunan pada kondisi pencahayaan redup maupun terlalu terang, sehingga menunjukkan bahwa kualitas pencahayaan masih menjadi salah satu faktor yang memengaruhi akurasi deteksi. Oleh karena itu, hasil penelitian ini menunjukkan kelayakan implementasi model pada kondisi pengujian yang telah dilakukan, tetapi belum dapat digeneralisasikan secara langsung untuk seluruh kondisi berkendara nyata yang memiliki variasi lingkungan lebih kompleks.

Penelitian selanjutnya disarankan melakukan pengujian pada kondisi luar ruangan maupun di dalam kendaraan yang sedang bergerak untuk mengevaluasi pengaruh getaran, perubahan posisi kepala, dan variasi pencahayaan dinamis terhadap performa sistem. Selain itu, penggunaan dataset yang lebih beragam, penerapan mekanisme *face detection* atau *eye detection*, serta pemanfaatan kamera inframerah pada kondisi cahaya rendah diharapkan dapat meningkatkan keandalan sistem dalam mendeteksi *microsleep* pada aplikasi keselamatan berkendara.

DAFTAR PUSTAKA

- Abdurrahman, N., Rahmat, B., & Sihananto, A. N. (2023). Perbandingan Performa Klasifikasi Citra Ikan Menggunakan Metode K-Nearest Neighbor (K-NN) Dan Convolutional Neural Network (CNN). *Jurnal Sistem Informasi Dan Informatika (JUSIFOR)*, 2(2), 84–93. <https://doi.org/10.33379/jusifor.v2i2.3728>
- Admin. (2024). *Kecelakaan Lalu Lintas Paling Sering Terjadi saat Kendaraan Melaju 40 Km/Jam*. Pusiknas Polri. https://pusiknas.polri.go.id/detail_artikel/kecelakaan_lalu_lintas_paling_serang_terjadi_saar_kendaraan_melaju_40_km_jam
- Akbar, Y., Bintoro, B., Mulyana, D. I., & Lestari, S. (2024). Deteksi Microsleep Menggunakan Convolutional Neural Network Dengan Sensor ESP32 Pada Pengemudi. *INTECOMS: Journal of Information Technology and Computer Science*, 7(5), 1668–1674. <https://doi.org/10.31539/intecom.v7i5.12142>
- Alajlan, N. N., & Ibrahim, D. M. (2023). DDD TinyML: A TinyML-Based Driver Drowsiness Detection Model Using Deep Learning. *Sensors*, 23(12), 1–35. <https://doi.org/https://doi.org/10.3390/s23125696>
- Donskoy, D., Gvindjiliya, V., & Ivliev, E. (2025). TinyML Classification for Agriculture Objects with ESP32. *Digital*, 5(4), 1–14. <https://doi.org/10.3390/digital5040048>
- Fanariotis, A., Orphanoudakis, T., Kotrotsios, K., Fotopoulos, V., Keramidas, G., & Karkazis, P. (2023). Power Efficient Machine Learning Models Deployment on Edge IoT Devices. *Sensors*, 23(3), 1–20. <https://doi.org/10.3390/s23031595>
- Gaffar, A. W. M., Widyawati, D., Fahmi, F., & Ulfiani, S. (2025). Deteksi dan Klasifikasi Jenis Kacang-Kacangan Menggunakan CNN Berbasis ESP32-Cam. *Jurnal Minfo Polgan*, 14(2), 2234–2241. <https://doi.org/https://doi.org/10.33395/jmp.v14i2.15359>
- Iswahyudi, R., Ikhsan, A. F., & Matin, I. M. M. (2024). Deteksi Kantuk Menggunakan Convolutional Neural Network (CNN) Berdasarkan Kedipan Mata. *Fuse-Teknik Elektro*, 4(2), 115–124.
- Mahmud, Ysuandi, I. A., Amir, A. A., Sulaiman, M. A., & Humera, A. P. (2026). Sistem Inventarisasi Kerusakan Perkerasan Jalan Deep Learning Dengan Arsitektur Convolutional Neural Network (CNN) YOLOv8. *DECODE: Jurnal Pendidikan Teknologi Informasi*, 6(1), 213–222. <https://doi.org/http://dx.doi.org/10.51454/decode.v6i1.1511>
- Maulidin, M. A. (2024). *80 Persen Kecelakaan Lalu Lintas Akibat Human Error*. Isafetymagazine. <https://isafetymagazine.com/80-persen-kecelakaan-lalu-lintas-akibat-human-error/>
- Ngo, D., Park, H. C., & Kang, B. (2025). Edge Intelligence: A Review of Deep Neural Network Inference in Resource-Limited Environments. *Electronics*, 14(12), 1–54. <https://doi.org/10.3390/electronics14122495>
- Nurdiansyah, D., Satrianansyah, S., & Sobri, A. (2024). Smart Robot Object Detection Menggunakan Esp-32 Cam. *Jurnal Teknik Informasi Dan Komputer (Tekinkom)*, 7(1), 272–279. <https://doi.org/10.37600/tekinkom.v7i1.1296>
- Pan, H., Logan, D. B., Stephens, A. N., Payre, W., Wang, Y., Peng, Z., Qin, Y., & Koppel, S. (2025). Exploring The Effect Of Driver Drowsiness On Takeover Performance During Automated Driving: An Updated Literature Review. *Accident Analysis and Prevention*, 216(March). <https://doi.org/10.1016/j.aap.2025.108023>
- Pantho, M. J. H., Bhowmik, P., & Bobda, C. (2021). Towards an Efficient CNN Inference Architecture

- Enabling In-Sensor Processing. *Sensors*, 21(6), 1–20. <https://doi.org/https://doi.org/10.3390/s21061955>
- Purba, A. M., Mahardika, F., Desmana, S., & Moniroh, N. (2026). Deteksi Kantuk Pengemudi Berdasarkan Keterbukaan Mata Menggunakan Model Ringan dari Spatiotemporal Pyramidal CNN. *Sudo Jurnal Teknik Informatika*, 4(4), 355–366. <https://doi.org/10.56211/sudo.v4i4.1342>
- Ramadhani L. Q., N., Suhartono, E., Aulia, S., & Hadiyoso, S. (2021). Deteksi Kantuk pada Pengemudi Berdasarkan Penginderaan Wajah Menggunakan PCA dan SVM. *Jurnal Rekayasa Elektrika*, 17(2), 129–136. <https://doi.org/10.17529/jre.v17i2.19884>
- Rodiah, Susetianingtias, D. T., & Patriya, E. (2025). Model Convolutional Neural Network (CNN) Custom Sequential untuk Klasifikasi Citra Ikan Hias Carassius Auratus pada Industri Akuakultur. *Decode: Jurnal Pendidikan Teknologi Informasi*, 5(3), 813–825. <https://doi.org/http://dx.doi.org/10.51454/decode.v5i3.1229>
- Safarov, F., Akhmedov, F., Abdusalomov, A. B., Nasimov, R., & Cho, Y. I. (2023). Real-Time Deep Learning-Based Drowsiness Detection: Leveraging Computer-Vision and Eye-Blink Analyses for Enhanced Road Safety. *Sensors*, 23(14), 1–15. <https://doi.org/10.3390/s23146459>
- Saputri, R. S., Apriliani, A., Amar, R. S., & Astri, L. Y. (2025). Deteksi Dini Microsleep Pada Pengemudi Kendaraan Roda Empat Melalui Citra Mata Menggunakan Algoritma YOLOv8. *Jurnal Processor*, 20(2), 165–179. <https://doi.org/10.33998/processor.2025.20.2.2529>
- Sugeng, S., & Nizar, T. N. (2023). Deteksi Aktivitas Mata, Mulut Dan Kemiringan Kepala Sebagai Fitur Untuk Deteksi Kantuk Pada Pengendara Mobil. *Komputika : Jurnal Sistem Komputer*, 12(1), 91–99. <https://doi.org/10.34010/komputika.v12i1.9688>
- Xu, M., Yoon, S., Fuentes, A., & Park, D. S. (2023). A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. *Pattern Recognition*, 137, 1–12. <https://doi.org/10.1016/j.patcog.2023.109347>
- Ziliwu, J. R., Setyawan, G. C., & Budiati, H. (2024). Penerapan ESP32-CAM dan TinyML dalam Klasifikasi Gambar Buah dan Sayuran. *Jutisi: Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, 13(1), 584. <https://doi.org/10.35889/jutisi.v13i1.1869>