

Deteksi Kantuk Pengemudi Berbasis Eye Aspect Ratio dan Head Pose Estimation dengan Integrasi IoT

Rama Amirul Ramadhan, Kurniawan D Irianto*

Fakultas Teknologi Industri, Program Studi Informatika, Universitas Islam Indonesia, Yogyakarta

Jalan Kaliurang Km 14,5, Umbulmartani, Kec. Ngemplak, Sleman, Yogyakarta, Indonesia

Email: ¹22523127@students.uui.ac.id, ^{2,*}145230101@uui.ac.id

Email Penulis Korespondensi: 145230101@uui.ac.id

Submitted: 04/07/2026; Accepted: 21/07/2026; Published: 21/07/2026

Abstrak—Penelitian ini mengusulkan sistem pendeteksi kantuk pengemudi secara real-time yang mengintegrasikan computer vision dengan Internet of Things (IoT) pada platform embedded terjangkau. Sistem menggunakan Camera Module 3 Wide NoIR yang terhubung ke Raspberry Pi 3 Model B+ untuk menangkap citra wajah pengemudi. Dua indikator visual dihitung secara paralel dari 68 titik landmark wajah menggunakan pustaka dlib: Eye Aspect Ratio (EAR) untuk mendeteksi penutupan mata berkepanjangan dan Head Pose Estimation menggunakan algoritma solvePnP untuk mendeteksi kepala mengangguk. Logika OR memicu alarm suara melalui speaker ketika EAR turun di bawah 0,25 selama lima frame berturut-turut atau sudut Pitch kepala melebihi 15 derajat selama sepuluh frame berturut-turut. Kejadian kantuk diklasifikasikan sebagai KANTUK_MATA, KANTUK KEPALA, atau KEDUANYA dan dikirim ke firebase Realtime Database untuk pemantauan jarak jauh. Evaluasi Black Box testing dengan 11 skenario mengonfirmasi seluruh fungsi utama sistem beroperasi dengan benar. Waktu respons rata-rata 2,00 detik melalui EAR dan 2,70 detik melalui Head Pose, keduanya dalam rentang yang dapat diterima untuk peringatan dini kantuk. Pendekatan multi-indikator menunjukkan kepala mengangguk terdeteksi lebih awal daripada penutupan mata pada skenario kantuk gradual, memberikan peringatan lebih dini dibandingkan sistem indikator tunggal. Pada kondisi beban CPU tinggi (>80%), frame rate turun ke 5 fps sehingga latensi total sistem berkisar 2,8–4,2 detik; kondisi ini masih memadai untuk peringatan dini namun bukan untuk respons sub-detik. Dari 11 skenario Black Box testing, sistem mencapai precision 100% (tidak ada false alarm pada kedipan normal) dan recall 100% (semua kondisi kantuk terdeteksi), meskipun pengujian dilakukan dalam kondisi laboratorium terkontrol sehingga generalisasi ke kondisi nyata masih diperlukan.

Kata Kunci: Pendeteksian Kantuk; Eye Aspect Ratio; Head Pose Estimation; Raspberry Pi; IoT; Firebase

Abstract—This study proposes a real-time driver drowsiness detection system integrating computer vision with Internet of Things (IoT) technology on an affordable embedded platform. The system uses a Camera Module 3 Wide NoIR connected to a Raspberry Pi 3 Model B+ to capture driver facial images. Two visual indicators are computed in parallel from 68 facial landmarks extracted using dlib: Eye Aspect Ratio (EAR) for detecting prolonged eye closure, and Head Pose Estimation using solvePnP for detecting head nodding. An OR-logic decision mechanism triggers an audio alarm when EAR falls below 0.25 for five consecutive frames or Pitch angle exceeds 15 degrees for ten consecutive frames. Events are classified as KANTUK_MATA, KANTUK KEPALA, or KEDUANYA and sent to firebase Realtime Database for remote monitoring. Black Box testing with 11 scenarios confirms all core functions operate correctly. Average response times of 2.00 seconds via EAR and 2.70 seconds via Head Pose are within acceptable ranges for early drowsiness warning. The multi-indicator approach demonstrates that head nodding is detected earlier than eye closure in gradual drowsiness scenarios, providing earlier warning than single-indicator systems. Under high CPU load conditions (>80%), the frame rate drops to 5 fps, resulting in a total system latency of 2.8–4.2 seconds; this condition is still adequate for early warning but not for sub-second responses. Across the 11 Black Box testing scenarios, the system achieved 100% precision (no false alarms during normal blinking) and 100% recall (all drowsiness conditions detected), although the testing was conducted under controlled laboratory conditions, necessitating further generalization to real-world environments.

Keywords: Driver Drowsiness Detection; Eye Aspect Ratio; Head Pose Estimation; Raspberry Pi; IoT; Firebase

1. PENDAHULUAN

Keselamatan berkendara merupakan isu global yang hingga saat ini masih menjadi perhatian utama di berbagai negara, termasuk Indonesia. World Health Organization (WHO) melaporkan bahwa kecelakaan lalu lintas menyebabkan lebih dari 1,19 juta kematian setiap tahun di seluruh dunia dan menjadi salah satu penyebab kematian tertinggi pada kelompok usia produktif [1]. Di Indonesia, sebagian besar kecelakaan lalu lintas disebabkan oleh faktor manusia, termasuk kelalaian, pelanggaran aturan lalu lintas, dan kondisi fisik pengemudi yang tidak optimal [2]. Di antara faktor-faktor tersebut, rasa kantuk pada pengemudi sering kali diremehkan meskipun kontribusinya terhadap risiko kecelakaan cukup signifikan. Pengemudi yang mengalami kantuk cenderung kehilangan konsentrasi, mengalami penurunan kewaspadaan, serta memiliki waktu reaksi yang lebih lambat dibandingkan kondisi normal [3]. Perlu ditegaskan bahwa sistem yang dikembangkan dalam penelitian ini diuji dalam kondisi simulasi laboratorium terkontrol, bukan pada kondisi berkendara nyata; validasi lapangan merupakan tahap lanjutan yang direkomendasikan.

Rasa kantuk pada pengemudi umumnya disebabkan oleh kurang tidur, beban kerja berlebihan, jam kerja yang panjang, dan perjalanan jarak jauh [4]. Risiko semakin meningkat pada perjalanan malam hari, ketika tubuh secara fisiologis berada dalam fase penurunan kewaspadaan [5], [6]. Penelitian menunjukkan bahwa kecelakaan terkait kantuk sering terjadi antara pukul 00.00 hingga 05.00, khususnya di jalan yang relatif sepi [7], [8]. Dampak



ekonomi dari kecelakaan akibat kantuk juga tidak dapat diabaikan, meliputi biaya pengobatan, kerusakan kendaraan, dan hilangnya produktivitas yang memberatkan perekonomian nasional [9].

Berbagai upaya telah dilakukan untuk memitigasi risiko kecelakaan akibat kantuk, mulai dari regulasi jam kerja pengemudi profesional hingga kampanye keselamatan berkendara. Namun pendekatan berbasis teknologi yang bersifat preventif dan real-time dinilai lebih efektif karena dapat memberikan peringatan langsung kepada pengemudi sebelum kondisi kantuk berkembang menjadi insiden berbahaya [2]. Salah satu pendekatan yang menjanjikan adalah integrasi IoT dengan sistem monitoring pengemudi yang memungkinkan pemantauan kondisi pengemudi secara kontinu dan pemberian peringatan otomatis [10]. Perkembangan teknologi computer vision dan embedded system membuka peluang baru untuk mengembangkan sistem pendeteksi kantuk yang bersifat non-invasif, tanpa memerlukan kontak fisik dengan pengemudi [9].

Pemilihan platform embedded yang tepat merupakan keputusan kritis dalam pengembangan sistem semacam ini. Beberapa kandidat yang lazim dipertimbangkan antara lain NVIDIA Jetson Nano, Raspberry Pi 4, Raspberry Pi 3B+, dan mikrokontroler berbasis Arduino. NVIDIA Jetson Nano menawarkan kemampuan inferensi deep learning yang tinggi berkat Graphics Processing Unit (GPU) terintegrasinya, namun harganya berkisar USD 99-149 dengan konsumsi daya 5-10 watt, menjadikannya kurang praktis untuk aplikasi berbasis baterai di kendaraan [11]. Raspberry Pi 4 memiliki performa CPU hingga dua kali lipat dibanding Raspberry Pi 3B+, namun harganya 40-60% lebih mahal dengan konsumsi daya rata-rata 3,4-7,6 watt versus 2,5-3,1 watt pada 3B+ [11]. Mikrokontroler Arduino, meski konsumsi dayanya sangat rendah, tidak memiliki kemampuan komputasi memadai untuk menjalankan algoritma deteksi wajah berbasis HOG dan Head Pose Estimation (HPE) secara real-time [12].

Raspberry Pi 3B+ dipilih berdasarkan tiga pertimbangan analitis. Pertama, rasio performa per biaya: dengan harga sekitar USD 35, prosesor quad-core 1,4 GHz-nya mampu menjalankan pipeline Eye Aspect Ratio (EAR) dan HPE pada 5-8 fps untuk deteksi kantuk yang membutuhkan respons dalam hitungan detik, tanpa memerlukan akselerasi GPU, sebagaimana dibuktikan oleh Civik dkk. [13] yang mengimplementasikan sistem deteksi kelelahan embedded berbiaya rendah. Kedua, efisiensi daya: konsumsi rata-rata 2,5-3,1 watt menjadikannya kompatibel langsung dengan satu daya USB kendaraan (5V/2A) tanpa adaptor tambahan, berbeda dengan Jetson Nano yang memerlukan sumber daya terpisah. Ketiga, kematangan ekosistem: dukungan library Python yang komprehensif beserta komunitas pengembang yang besar mempercepat siklus pengembangan dan pemecahan masalah secara signifikan [11]. Dengan demikian, Raspberry Pi 3B+ bukan sekadar pilihan terjangkau, melainkan pilihan yang secara teknis optimal untuk profil beban kerja dan kendala daya sistem yang dikembangkan.

Secara teoritis, deteksi kantuk pengemudi berbasis computer vision memanfaatkan dua pendekatan utama yang saling melengkapi. Pertama, EAR merupakan metode kuantitatif yang dikembangkan oleh Soukupova dan Cech untuk mengukur keterbukaan mata berdasarkan geometri titik-titik landmark wajah. EAR dihitung sebagai rasio antara jarak vertikal kelopak mata dan jarak horizontal kelopak mata, sehingga nilainya mendekati nol saat mata tertutup dan meningkat proporsional saat mata terbuka. Metode ini unggul karena sederhana secara komputasi dan tidak memerlukan kalibrasi per individu, menjadikannya kandidat ideal untuk implementasi pada platform embedded berbiaya rendah [12].

Kedua, HPE berbasis Perspective-n-Point (PnP) merupakan teknik estimasi orientasi kepala dalam ruang tiga dimensi menggunakan proyeksi titik-titik referensi wajah pada citra dua dimensi [14]. Algoritma solvePnP pada OpenCV menyelesaikan persoalan PnP dengan mencocokkan koordinat 3D model wajah generik terhadap koordinat 2D landmark yang terdeteksi, menghasilkan vektor rotasi yang didekomposisi menjadi sudut Euler: Pitch (mengangguk), Yaw (menoleh), dan Roll (memiringkan kepala). Sudut Pitch yang positif mengindikasikan kepala menunduk ke depan, suatu manifestasi khas kantuk pengemudi yang sering terjadi sebelum mata sepenuhnya tertutup sehingga tidak dapat dideteksi oleh EAR saja [4]. Kombinasi kedua metode dalam satu pipeline pemrosesan menggunakan pustaka dlib memberikan landasan teoritis yang kuat bagi pendekatan multi-indikator yang diusulkan.

Sejumlah penelitian terdahulu telah mengkaji pendeteksian kantuk dengan berbagai pendekatan. Shimpi dkk. [12] mengimplementasikan sistem berbasis EAR pada Raspberry Pi menggunakan webcam konvensional dan berhasil mendeteksi penutupan mata secara real-time, namun sistem mereka hanya mengandalkan indikator tunggal sehingga tidak mampu mendeteksi kantuk berupa kepala mengangguk dan tidak memiliki kemampuan pemantauan jarak jauh melalui IoT.

Florez dkk. [15] memperluas pendekatan EAR dengan mengombinasikannya bersama Mouth Aspect Ratio (MAR) untuk mendeteksi menguap sebagai indikator kantuk tambahan pada platform embedded. Meskipun pendekatan ini meningkatkan cakupan deteksi, sistem tersebut tidak mengimplementasikan HPE dan tidak dilengkapi integrasi cloud IoT, sehingga data kejadian kantuk tidak dapat diakses secara historis dari jarak jauh.

Chen dkk. [14] mengembangkan sistem multi-fitur yang lebih komprehensif dengan mengombinasikan EAR, HPE, dan fitur mulut menggunakan arsitektur LSTM pada Raspberry Pi 4. Meskipun akurasinya tinggi berkat pendekatan deep learning, sistem ini memerlukan perangkat keras yang lebih mahal dan tidak terintegrasi dengan platform cloud IoT. Das dkk. [16] mengimplementasikan deteksi kantuk berbasis IoT menggunakan CNN-LSTM dengan analisis gerakan wajah dan arsitektur U-Net, namun memerlukan komputasi yang lebih berat dan tidak dirancang untuk platform embedded terjangkau. Kaur dan Bhatt [17] mengusulkan deteksi kantuk berbasis facial landmark dengan peringatan alarm audio, namun terbatas pada platform komputer konvensional tanpa kemampuan embedded maupun integrasi IoT.

Dari kajian tersebut, teridentifikasi tiga gap utama yang belum terjawab secara bersamaan. Pertama, sebagian besar sistem pada platform embedded terjangkau masih mengandalkan EAR sebagai indikator tunggal dan tidak mampu mendeteksi kantuk berupa kepala mengangguk. Kedua, integrasi IoT untuk pemantauan jarak jauh dan pencatatan data historis yang terklasifikasi masih sangat terbatas. Ketiga potensi kamera NoIR pada platform embedded murah untuk kondisi pencahayaan rendah belum banyak dieksplorasi.

Penelitian ini memberikan tiga kontribusi nyata. Pertama, pembuktian bahwa kombinasi EAR dan HPE dapat diimplementasikan efisien dalam satu pipeline menggunakan 68 titik landmark dlib yang sama tanpa modul tambahan, sehingga beban komputasi pada Raspberry Pi 3B+ tetap terkendali. Kedua, logika OR sebagai mekanisme keputusan multi-indikator terbukti mendeteksi dua jalur manifestasi kantuk secara independen, dengan temuan bahwa indikator HPE terpicu rata-rata 0,7 detik lebih awal dari EAR pada skenario kantuk gradual. Ketiga, integrasi firebase dengan klasifikasi tipe indikator pemicu (KANTUK_MATA, KANTUK KEPALA, atau KEDUANYA) menghasilkan data historis yang lebih informatif dan memungkinkan pemantauan pengemudi dari jarak jauh secara real-time, kapabilitas yang tidak dimiliki sistem-sistem terdahulu yang dikaji.

Pengujian sistem dilakukan dalam kondisi terkontrol di dalam ruangan menggunakan simulasi kantuk, sehingga hasil yang diperoleh mencerminkan performa sistem pada lingkungan pengujian tersebut. Kondisi malam hari dan pencahayaan rendah belum diuji dalam penelitian ini, meskipun kamera NoIR yang digunakan mendukungnya [18]. Validasi pada kondisi berkendara nyata di lapangan merupakan rekomendasi untuk penelitian lanjutan, termasuk mengatasi masalah variasi pencahayaan pada malam hari yang belum diuji dalam penelitian ini [19].

Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi prototipe sistem pendeteksi kantuk pengemudi secara real-time yang mengombinasikan EAR dan HPE dalam satu pipeline pemrosesan pada Raspberry Pi 3 Model B+, dilengkapi dengan integrasi firebase sebagai platform IoT. Secara spesifik, penelitian ini menjawab tiga pertanyaan, pertama, apakah threshold EAR 0,25 dan Pitch 15 derajat yang dikalibrasi empiris efektif membedakan kondisi waspada dari kantuk. Kedua, apakah pendekatan multi-indikator menghasilkan waktu respons yang lebih awal dibandingkan indikator tunggal. Ketiga, apakah sistem dapat beroperasi mandiri dan mengirimkan data ke firebase secara real-time pada platform Raspberry Pi 3B+.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dikembangkan menggunakan metodologi prototyping yang memungkinkan pembangunan, pengujian, dan penyempurnaan sistem secara iteratif berdasarkan hasil evaluasi [20]. Metodologi ini dipilih karena sistem melibatkan integrasi perangkat keras dan perangkat lunak yang memerlukan kalibrasi empiris, khususnya untuk menetapkan threshold EAR dan sudut pitch yang optimal pada perangkat nyata.

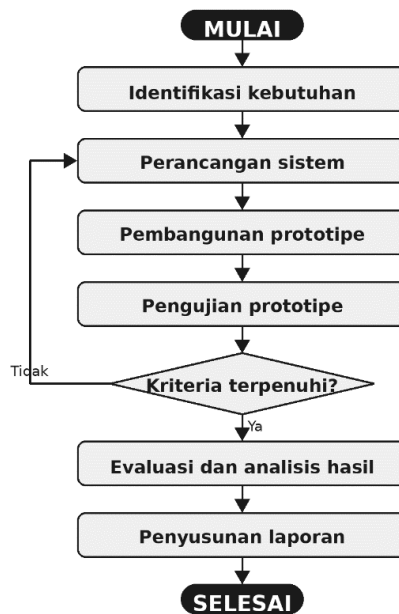
Proses penelitian dilaksanakan melalui enam tahap yang saling berkaitan. Pertama adalah identifikasi kebutuhan, dilakukan melalui studi literatur dan wawancara dengan pengemudi untuk memahami manifestasi kantuk serta kebutuhan fungsional sistem. Kedua adalah perancangan sistem, meliputi desain arsitektur perangkat keras dan perangkat lunak serta penentuan parameter awal threshold EAR dan sudut pitch. Ketiga adalah pembangunan prototipe, yaitu implementasi algoritma EAR dan HPE pada Raspberry Pi 3B+ berikut integrasi firebase dan alarm speaker.

Keempat adalah pengujian prototipe menggunakan Black Box testing dengan 11 skenario serta pengukuran waktu respon dan pengujian pada empat subjek. Kelima adalah evaluasi dan analisis hasil, membandingkan performa sistem dengan kriteria keberhasilan dan penelitian terdahulu. Keenam adalah penyusunan laporan yang mendokumentasikan seluruh proses dan temuan penelitian. Apabila pada tahap evaluasi ditemukan bahwa kriteria keberhasilan belum terpenuhi, proses kembali ke tahap perancangan untuk dilakukan perbaikan dan iterasi ulang.

Setiap tahap siklus iterasi dalam metodologi prototyping ini dijalankan dengan keberhasilan yang terukur dan eksplisit. Pertama, tidak terjadi false alarm pada kondisi kedipan mata normal, diverifikasi melalui 10 sesi pengujian berturut-turut. Kedua, waktu respons alarm tidak melebihi 3 detik sejak kondisi kantuk pertama kali terdeteksi oleh sistem, dan diukur menggunakan stopwatch digital. Ketiga, seluruh 11 skenario pengujian fungsional Black Box berhasil sesuai dengan output yang diharapkan. Iterasi dinyatakan selesai apabila ketiga kriteria tersebut terpenuhi secara bersamaan, sehingga proses kalibrasi tidak bersifat subjektif melainkan dikendalikan oleh bukti empiris yang terverifikasi.

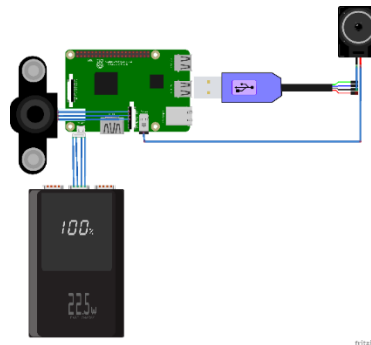
Arsitektur sistem yang dikembangkan terdiri dari empat komponen utama: Camera Module 3 Wide NoIR sebagai unit akuisisi citra, Raspberry Pi 3 Model B+ sebagai unit pemrosesan utama, speaker sebagai keluaran peringatan, dan firebase realtime database sebagai platform IoT [21].

Alur penelitian secara keseluruhan, mulai dari identifikasi kebutuhan hingga tahap penyusunan laporan, digambarkan pada Gambar 1 berikut. Apabila pada tahap pengujian ditemukan bahwa kriteria keberhasilan belum terpenuhi, proses kembali ke tahap perancangan sistem untuk perbaikan dan iterasi ulang hingga seluruh kriteria terpenuhi.

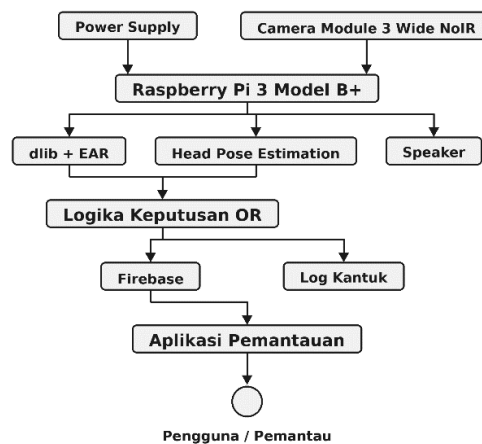


Gambar 1. Alur Penelitian Metodologi Prototyping

Rangkaian perangkat keras sistem yang digunakan dalam penelitian ini ditunjukkan pada Gambar 2. Gambar tersebut memperlihatkan koneksi fisik antara power bank sebagai sumber daya, Raspberry Pi 3 Model B+ sebagai unit pemrosesan utama, Camera Module 3 Wide NoIR yang terhubung melalui port CSI, dan speaker yang terhubung melalui port audio 3,5mm



Gambar 2. Rangkaian Perangkat Keras Sistem Pendeteksi Kantuk Pengemudi

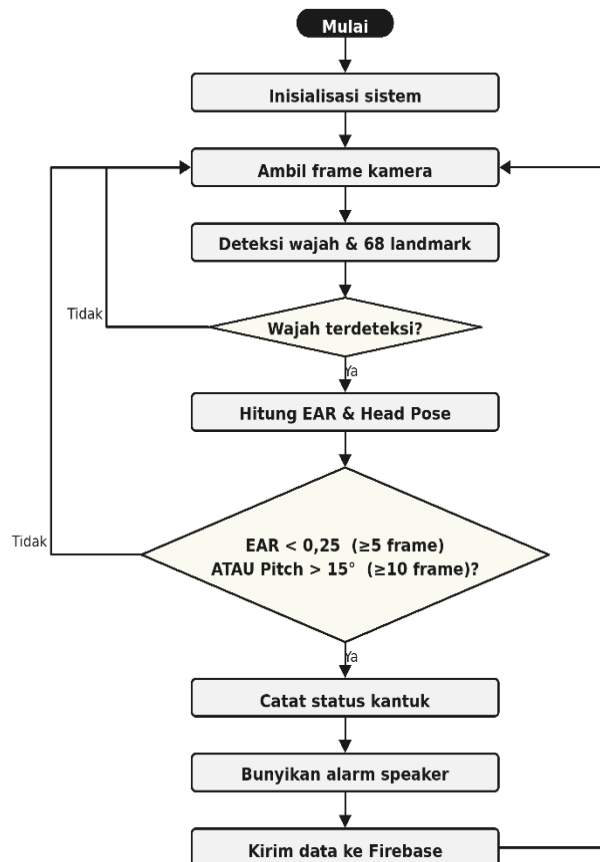


Gambar 3. Diagram Arsitektur Sistem Pendeteksi Kantuk Pengemudi

Gambar 3 menunjukkan arsitektur perangkat keras dan perangkat lunak yang digunakan dalam penelitian. Camera Module 3 Wide NoIR menangkap citra wajah pengemudi dan mengirimkan frame video ke Raspberry Pi 3 Model B+ sebagai unit pemrosesan utama. Raspberry Pi menjalankan modul deteksi wajah menggunakan dlib, perhitungan EAR, serta HPE menggunakan OpenCV. Hasil kedua indikator dievaluasi menggunakan logika OR untuk menentukan kondisi kantuk. Ketika kondisi kantuk terdeteksi, sistem mengaktifkan speaker sebagai alarm

peringatan dan secara bersamaan mengirimkan data status serta riwayat kejadian ke firebase sehingga dapat dipantau secara real-time melalui aplikasi pemantauan.

Secara lebih rinci, sistem dimulai dengan pengambilan frame kamera secara kontinu, kemudian deteksi wajah dan ekstraksi 68 titik landmark menggunakan pustaka dlib [12]. Dari landmark yang sama, sistem menghitung EAR dan sudut pitch HPE secara paralel. Apabila salah satu atau kedua kondisi kantung terpenuhi, sistem membunyikan alarm dan mengirim data ke firebase. Alur kerja sistem secara lengkap ditunjukkan pada Gambar 4.



Gambar 4. Diagram Alur Kerja Sistem Pendeteksi Kantuk Pengemudi

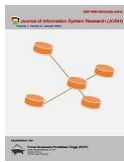
Gambar 4 menunjukkan alur kerja sistem yang berjalan secara kontinu mulai dari inisialisasi, pengambilan frame, deteksi wajah, perhitungan dua indikator secara paralel, hingga evaluasi logika OR yang memicu peringatan dan pengiriman data ke firebase apabila kondisi kantung terdeteksi.

2.2 Eye Aspect Ratio (EAR)

Eye Aspect Ratio (EAR) merupakan metode kuantitatif berbasis geometri untuk mengukur keterbukaan mata menggunakan koordinat landmark wajah [15]. EAR dihitung berdasarkan perbandingan jarak vertikal dan horizontal kelopak mata menggunakan rumus berikut:

$$EAR = \frac{(|p2-p6| + |p3-p5|)}{(2 \times |p1-p4|)} \quad (1)$$

di mana $p1$ hingga $p6$ merupakan enam titik landmark pada area setiap mata. Dalam penelitian ini, EAR dihitung dari 12 titik landmark (6 per mata) yang diekstraksi dari 68 titik dlib [15]. Alarm dipicu apabila EAR turun di bawah threshold 0,25 selama 5 frame berturut-turut, nilai yang dikalibrasi secara empiris menyesuaikan frame rate Raspberry Pi 3B+ sebesar 5–8 fps. Proses kalibrasi dilakukan melalui dua siklus iteratif: siklus pertama menggunakan threshold 20 frame yang menghasilkan waktu respons terlalu lambat, dan siklus kedua menetapkan 5 frame sebagai nilai optimal yang memberikan keseimbangan antara sensitivitas deteksi dan ketahanan terhadap false alarm akibat kedipan normal. Nilai EAR dihitung dari rata-rata kedua mata untuk meningkatkan robustisitas terhadap variasi pencahayaan dan posisi wajah yang tidak simetris. Meskipun demikian, threshold tunggal 0,25 memiliki keterbatasan subjektivitas anatomis: individu dengan mata yang secara anatomis lebih kecil atau pengguna kacamata tebal berpotensi menghasilkan false alarm yang lebih tinggi. Adaptive threshold yang dikalibrasi per individu pada awal sesi berkendara direkomendasikan sebagai pengembangan lanjutan.



2.3 Head Pose Estimation

Head Pose Estimation diimplementasikan menggunakan algoritma Perspective-n-Point (PnP) melalui fungsi solvePnP pada OpenCV [14]. Enam titik referensi wajah 3D generik digunakan sebagai acuan: ujung hidung (0, 0, 0), dagu (0, -330, -65), sudut mata kiri (-225, 170, -135), sudut mata kanan (225, 170, -135), sudut mulut kiri (-150, -150, -125), dan sudut mulut kanan (150, -150, -125) dalam satuan milimeter. Vektor rotasi didekomposisi menggunakan RQDecomp3x3 untuk memperoleh sudut Pitch, Yaw, dan Roll. Alarm dipicu apabila sudut Pitch melebihi 15 derajat selama 10 frame berturut-turut, dikalibrasi melalui iterasi dari nilai awal 10 derajat yang menghasilkan false alarm pada gerakan kepala wajar. Secara teknis, algoritma solvePnP bekerja dengan memetakan enam titik referensi wajah tiga dimensi generik ke koordinat dua dimensi yang diperoleh dari landmark wajah pada citra. Vektor rotasi yang dihasilkan didekomposisi untuk memperoleh tiga sudut Euler: Pitch (rotasi sumbu-X, mengangguk), Yaw (rotasi sumbu-Y, menoleh), dan Roll (rotasi sumbu-Z, memiringkan kepala). Hanya sudut Pitch yang digunakan sebagai indikator kantuk karena gerakan mengangguk ke depan merupakan manifestasi khas kantuk pengemudi, sedangkan Yaw dan Roll merupakan gerakan normal dalam aktivitas berkendara.

2.4 Logika Keputusan Multi-Indikator dan Integrasi IoT

Sistem menggunakan logika OR sebagai mekanisme keputusan: alarm dipicu apabila kondisi EAR terpenuhi (KANTUK_MATA), kondisi HPE terpenuhi (KANTUK KEPALA), atau keduanya aktif bersamaan (KEDUANYA). Pendekatan ini memungkinkan deteksi dua manifestasi kantuk yang berbeda secara independen dalam satu pipeline pemrosesan [14]. firebase digunakan untuk menyimpan dua struktur data: node status_terkini yang diperbarui setiap 2 detik berisi nilai EAR, sudut Pitch, dan status kondisi;[] serta node log_kantuk yang mencatat setiap kejadian kantuk beserta jenis indikator pemicu dan timestamp [22]. Mekanisme error handling memastikan deteksi dan peringatan lokal tetap berjalan apabila koneksi internet terputus [23].

2.5 Konfigurasi Sistem

Tabel 1 merangkum seluruh parameter konfigurasi sistem yang digunakan pada implementasi akhir, termasuk parameter yang mengalami penyesuaian melalui siklus kalibrasi iteratif.

Tabel 1. Parameter Konfigurasi Sistem

Parameter	Nilai	Dasar Penetapan
Resolusi citra	640 × 480 piksel	Kompromi akurasi vs. kecepatan [12]
EAR Threshold	0,25	Kalibrasi empiris [15]
EAR Consecutive Frames	5 frame	Disesuaikan frame rate RPi 3B+ [12]
Head Pose Pitch Threshold	15 derajat	Kalibrasi iterasi dari 10° [14]
Head Pose Consecutive Frames	10 frame	Gerakan kepala lebih gradual [14]
Jarak subjek ke kamera	40–60 cm	Optimal untuk landmark dan head pose
Interval pengiriman firebase	2 detik	Keseimbangan beban jaringan [22]
Rotasi citra	180 derajat	Kompensasi orientasi fisik kamera [11]

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengujian Deteksi Wajah dan Nilai EAR

Sistem mampu mendeteksi wajah subjek secara konsisten pada jarak 40–60 cm dalam kondisi pencahayaan terang, sesuai karakteristik model HOG dlib [12], [15]. Tabel 2 menyajikan rentang nilai EAR yang terukur pada tiga kondisi mata yang diuji. Perlu dicatat bahwa model deteksi wajah berbasis HOG memiliki keterbatasan terhadap variasi iluminasi yang drastis. Pada kondisi pencahayaan rendah atau perubahan cahaya yang cepat seperti saat pengemudi melewati terowongan atau beralih dari siang ke malam hari, kemampuan deteksi landmark wajah dapat menurun secara signifikan [18], [19]. Sebagai mitigasi, kamera NoIR yang digunakan berpotensi memberikan performa lebih baik pada kondisi cahaya rendah karena sensitif terhadap spektrum inframerah. Untuk penelitian lanjutan, penambahan teknik preprocessing seperti histogram equalization adaptif direkomendasikan untuk meningkatkan ketahanan sistem terhadap variasi iluminasi yang ekstrem.

Tabel 2. Hasil Pengukuran Nilai EAR pada Berbagai Kondisi Mata

Kondisi Mata	Rentang EAR	Rata-rata	Keterangan
Terbuka (normal)	0,25 – 0,37	0,31	Di atas threshold; tidak memicu alarm
Berkedip sesaat	0,14 – 0,23	0,19	Counter reset; tidak false alarm
Tertutup (KANTUK_MATA)	0,13 – 0,20	0,16	Di bawah threshold ≥ 5 frame; alarm berbunyi

Pemisahan yang jelas antara rentang mata terbuka (0,25–0,37) dan tertutup (0,13–0,20) mengonfirmasi bahwa threshold 0,25 memberikan diskriminasi efektif antara kondisi waspada dan kantuk. Tidak ditemukan false



alarm selama kedipan normal karena durasi kedipan alami tidak mencapai 5 frame berturut-turut pada frame rate 5–8 fps [12]. Hal ini menunjukkan bahwa mekanisme penghitungan frame berturut-turut (consecutive frame counter) berfungsi efektif sebagai filter untuk membedakan kedipan normal dari kondisi kantuk sesungguhnya. Pemisahan rentang nilai EAR yang jelas antara kondisi terbuka (0,25–0,37) dan tertutup (0,13–0,20) menunjukkan bahwa threshold 0,25 hasil kalibrasi empiris berada tepat di antara kedua rentang tersebut, memberikan margin diskriminasi yang memadai. Nilai EAR rata-rata pada kondisi mata terbuka sebesar 0,31 memberikan jarak 0,06 dari threshold, yang cukup untuk mengakomodasi variasi alami nilai EAR akibat pergerakan wajah ringan pengemudi selama berkendara tanpa memicu false alarm. Meskipun demikian, perlu diakui bahwa threshold EAR 0,25 memiliki keterbatasan dari sisi subjektivitas anatomis. Nilai EAR bervariasi antar individu bergantung pada bentuk mata dan rasio kelopak mata. Pada pengujian multi-subjek, nilai EAR mata terbuka berkisar 0,25–0,37 dengan rata-rata 0,31, menunjukkan threshold 0,25 masih memberikan margin memadai untuk keempat subjek yang diuji. Namun untuk populasi yang lebih beragam, misalnya individu dengan mata yang secara anatomis lebih kecil atau pengguna kacamata tebal threshold tunggal ini mungkin menghasilkan false alarm lebih tinggi. Penelitian lanjutan disarankan mengeksplorasi adaptive threshold yang dikalibrasi per individu pada awal sesi berkendara dengan merekam nilai EAR baseline selama 30 detik pertama untuk threshold yang lebih personal dan akurat.

3.2 Hasil Pengujian Head Pose Estimation

Modul HPE berhasil mengestimasi orientasi kepala secara real-time menggunakan solvePnP [14]. Tabel 3 menyajikan sudut Pitch yang terukur pada empat kondisi posisi kepala, diperoleh dari 10 kali pengukuran per kondisi.

Tabel 3. Hasil Pengukuran Sudut Pitch Head Pose pada Berbagai Kondisi Kepala

Kondisi Kepala	Rentang Pitch	Rata-rata	Keterangan
Tegak (normal)	–5° hingga +5°	0,8°	Di bawah threshold; NORMAL [14]
Menunduk ringan	6° hingga 13°	9,5°	Di bawah threshold; tidak terpicu [14]
Menunduk sedang	15° hingga 28°	21,3°	Melebihi threshold ≥ 10 frame; alarm berbunyi [14]
Menunduk dalam	29° hingga 45°	36,7°	Jauh di atas threshold; alarm berbunyi [14]

Sistem berhasil membedakan posisi kepala tegak ($\text{Pitch} < 5^\circ$) dari kepala menunduk yang mengindikasikan kantuk ($\text{Pitch} > 15^\circ$, rata-rata $21,3^\circ$ pada kondisi sedang). Tidak ditemukan false alarm akibat gerakan menoleh (Yaw) atau memiringkan kepala (Roll) karena threshold hanya diterapkan pada sudut Pitch [14]. Pemilihan Pitch sebagai satu-satunya sudut yang dimonitor merupakan keputusan desain yang tepat, karena gerakan mengangguk ke depan (Pitch positif) merupakan manifestasi khas kantuk pengemudi, sedangkan gerakan menoleh ke samping merupakan perilaku normal saat memeriksa spion atau berganti lajur. Hal ini juga menjelaskan mengapa threshold 15° dipilih nilai ini memberikan toleransi terhadap gerakan kepala ringan yang wajar (di bawah 13°) sekaligus tetap sensitif terhadap kepala menunduk yang mengindikasikan kantuk (rata-rata $21,3^\circ$ pada kondisi menunduk sedang).

3.3 Hasil Pengujian Deteksi Kantuk Simultan (Multi-Indikator)

Pengujian kantuk simultan dilakukan dengan mensimulasikan penutupan mata dan kepala menunduk secara bersamaan. Sistem berhasil mendeteksi kondisi ini dan mencatatnya sebagai status KEDUANYA pada log firebase [22]. Temuan krusial adalah bahwa indikator HPE terpicu lebih dahulu dibandingkan EAR pada simulasi kantuk gradual kepala mulai mengangguk sebelum mata sepenuhnya tertutup. Hal ini konsisten dengan pola kantuk gradual dalam literatur [14], [4] dan mengonfirmasi nilai praktis pendekatan multi-indikator dalam memberikan peringatan lebih awal. Perlu dicatat bahwa temuan ini bersifat observasional dan diperoleh dari pengujian terbatas dengan empat subjek dalam kondisi simulasi terkontrol, sehingga belum dapat diklaim sebagai temuan yang tergeneralisasi secara saintifik. Validitas klaim bahwa HPE terpicu lebih dahulu dari EAR idealnya memerlukan pengujian dengan dataset standar yang diakui secara akademis yang menyediakan rekaman video pengemudi dalam kondisi kantuk nyata dengan label tervalidasi. Pengujian dengan dataset standar tersebut akan memungkinkan perbandingan performa yang objektif dengan sistem-sistem terdahulu dan menghilangkan potensi bias yang melekat pada pengujian mandiri. Hal ini menjadi rekomendasi utama untuk penelitian lanjutan guna meningkatkan validitas saintifik dari temuan yang dilaporkan dalam penelitian ini. Dataset standar yang direkomendasikan untuk validasi lanjutan antara lain NTHU-DDD (National Tsing Hua University Driver Drowsiness Detection) dan UTA-RLDD (Real-Life Drowsiness Dataset), keduanya menyediakan rekaman video pengemudi dalam kondisi kantuk nyata dengan label ground truth tervalidasi.

Keunggulan logika OR terletak pada kemampuannya menangkap dua jalur manifestasi kantuk secara independen. Apabila hanya EAR yang digunakan, kondisi kepala mengangguk dengan mata masih terbuka sebagian tidak akan terdeteksi. Sebaliknya, apabila hanya HPE yang digunakan, penutupan mata pada posisi kepala tegak tidak akan terdeteksi. Data jenis indikator pemicu yang tersimpan di firebase memungkinkan analisis pola kantuk individual yang informatif untuk pengambilan keputusan preventif [22]. Dari 5 kali percobaan kantuk simultan yang dilakukan, ditemukan bahwa pada 4 dari 5 percobaan, indikator KANTUK KEPALA terpicu lebih dahulu dibandingkan KANTUK MATA. Rata-rata selisih waktu antara terpicunya dan terpicunya EAR adalah sekitar 0,7 detik. Hal ini menunjukkan bahwa pada kondisi kantuk gradual, kepala pengemudi cenderung mengangguk lebih dahulu sebelum mata sepenuhnya terpejam, yang secara fisiologis konsisten dengan mekanisme kehilangan tonus otot leher pada fase awal kantuk [5]. Temuan ini mengonfirmasi bahwa sistem multi-indikator tidak hanya mendeteksi lebih banyak manifestasi kantuk, tetapi juga memberikan peringatan yang lebih dini secara rata-rata dibandingkan sistem EAR tunggal.

3.4 Hasil Pengujian Waktu Respons

Waktu respons diukur secara manual menggunakan stopwatch sebanyak 5 kali pengulangan per indikator, dihitung sejak onset kondisi kantuk hingga alarm berbunyi [17]. Tabel 4 menyajikan hasil pengukuran.

Tabel 4. Hasil Pengukuran Waktu Respons per Indikator

Percobaan	Waktu Respons EAR (detik)	Waktu Respons Head Pose (detik)
1	1,90	2,40
2	2,90	3,10
3	1,85	2,65
4	1,75	2,80
5	1,62	2,55
Rata-rata	2,00	2,70
Minimum	1,62	2,40
Maksimum	2,90	3,10

Waktu respons EAR rata-rata 2,00 detik dan HPE 2,70 detik. Perbedaan waktu respons merupakan konsekuensi dari perbedaan threshold consecutive frame (5 vs 10 frame) yang ditetapkan untuk mencegah false alarm dari gerakan kepala wajar. Keduanya berada dalam rentang yang dapat diterima untuk peringatan dini kantuk, mengingat kondisi berbahaya umumnya membutuhkan lebih dari 2–3 detik sebelum menimbulkan risiko kehilangan kendali kendaraan [5], [6]. Variasi waktu respons yang cukup besar pada indikator EAR (1,62–2,90 detik) disebabkan oleh ketidakkonsistenan frame rate Raspberry Pi 3B+ yang berkisar antara 5–8 fps tergantung beban komputasi saat itu. Pada frame rate 5 fps, 5 frame berturut-turut setara dengan 1,0 detik deteksi murni ditambah latensi pemrosesan, sedangkan pada 8 fps waktu yang dibutuhkan menjadi lebih singkat. Variasi serupa terjadi pada indikator HPE dengan rentang 2,40–3,10 detik, yang konsisten dengan jumlah consecutive frame yang lebih besar (10 frame). Untuk pengembangan selanjutnya, peningkatan ke platform dengan frame rate lebih stabil akan menghasilkan waktu respons yang lebih konsisten dan dapat diprediksi, sehingga meningkatkan keandalan sistem dalam kondisi penggunaan nyata. Perlu dicatat bahwa data waktu respon yang disajikan pada Tabel 4 merupakan hasil pengukuran dari 5 kali percobaan, sehingga analisis statistik yang lebih komprehensif memiliki keterbatasan akibat ukuran sampel yang kecil. Deviasi standar waktu respons EAR dihitung sebesar $\pm 0,46$ detik (dari rentang 1,62–2,90 detik), sedangkan deviasi standar waktu respons HPE sebesar $\pm 0,27$ detik (dari rentang 2,40–3,10 detik). Nilai deviasi standar yang relatif kecil pada HPE menunjukkan konsistensi yang lebih baik, sedangkan variasi yang lebih besar pada EAR disebabkan oleh ketidakstabilan frame rate Raspberry Pi 3B+. Untuk validasi statistik yang lebih kuat, pengujian dengan minimal 30 sampel per kondisi direkomendasikan pada penelitian lanjutan.

3.5 Hasil Pengujian Firebase IoT

Node status_terkini berhasil menerima pembaruan setiap 2 detik berisi nilai EAR, sudut Pitch/Yaw/Roll, status kondisi, dan timestamp [22]. Node log kantuk mencatat setiap kejadian kantuk dengan jenis indikator pemicu yang benar. Tidak ditemukan kehilangan data pada kondisi koneksi stabil. Ketika koneksi internet diputus, deteksi dan peringatan lokal tetap berjalan tanpa gangguan, membuktikan mekanisme error handling berfungsi sesuai rancangan dan fungsi keselamatan tidak bergantung pada konektivitas internet [23], .

Hal ini penting secara praktis, mengingat kondisi sinyal internet yang tidak stabil kerap ditemui di jalan tol, terowongan, maupun daerah dengan infrastruktur jaringan terbatas. Selain itu, struktur data dua node yang digunakan memberikan fleksibilitas pemantauan: node status_terkini memungkinkan pemantauan real-time kondisi pengemudi, sedangkan node log_kantuk memungkinkan analisis historis pola kantuk yang dapat dimanfaatkan oleh perusahaan transportasi untuk evaluasi kinerja pengemudi secara berkala. Dengan perbedaan jenis indikator pemicu (KANTUK MATA, KANTUK KEPALA, atau KEDUANYA) pada setiap log, data yang tersimpan jauh lebih informatif dibandingkan sistem yang hanya mencatat status kantuk secara biner.

3.6 Hasil Pengujian Black Box

Tabel 5 merangkum hasil pengujian Black Box terhadap 11 skenario yang mencakup seluruh fungsi utama sistem.

Tabel 5. Rekapitulasi Hasil Pengujian Black Box

No	Fitur yang Diuji	Output yang Diharapkan	Hasil
1	Deteksi wajah	Kotak deteksi dan landmark tampil	Berhasil
2	EAR – mata terbuka	EAR > 0,25; status normal	Berhasil
3	EAR – kedipan normal	Counter reset; tidak false alarm	Berhasil
4	Deteksi kantuk mata	Status berubah; alarm berbunyi	Berhasil
5	Deteksi kantuk kepala	Status berubah; alarm berbunyi	Berhasil
6	Deteksi simultan keduanya	Status keduanya tercatat	Berhasil
7	Alarm suara	Speaker mengeluarkan bunyi	Berhasil
8	Pengiriman firebase	Data EAR, Pitch, jenis tersimpan	Berhasil
9	Gangguan internet	Deteksi lokal tetap berjalan	Berhasil
10	Live streaming	Video, EAR, Pitch tampil di browser	Berhasil
11	Auto-start	Program berjalan otomatis	Berhasil

Seluruh 11 skenario dinyatakan berhasil. Catatan pada skenario 10: penggunaan live streaming bersamaan dengan pipeline deteksi multi-indikator menyebabkan CPU load tinggi pada Raspberry Pi 3B+ yang berpotensi mengganggu stabilitas koneksi Wi-Fi pada penggunaan jangka panjang [12]. Meskipun demikian, perlu ditekankan bahwa fitur live streaming bersifat opsional dan tidak memengaruhi fungsi keselamatan utama sistem. Fungsi inti yaitu deteksi wajah, perhitungan EAR dan Head Pose, pemberian alarm suara, serta pengiriman data ke firebase tetap berjalan dengan baik tanpa bergantung pada fitur live streaming. Keberhasilan 10 dari 11 skenario tanpa catatan keterbatasan dan 1 skenario dengan catatan minor menunjukkan tingkat kematangan fungsional prototipe yang baik untuk tahap pengembangan awal menggunakan metodologi prototyping [20]. Perlu ditekankan bahwa 11 skenario Black Box yang digunakan dirancang untuk memvalidasi kelengkapan dan kebenaran fungsionalitas sistem secara teknis, bukan untuk merepresentasikan seluruh kompleksitas kondisi berkendara nyata. Black Box testing dalam konteks ini berfungsi sebagai verifikasi bahwa setiap komponen sistem bekerja sesuai spesifikasi yang dirancang. Validasi terhadap kompleksitas dunia nyata termasuk variasi kondisi pencahayaan, kecepatan berkendara, getaran kendaraan, dan keberagaman karakteristik pengemudi memerlukan pengujian lapangan yang komprehensif dan merupakan arah penelitian lanjutan yang direkomendasikan.

3.7 Pembahasan

Dibandingkan dengan penelitian terdahulu, sistem yang dikembangkan memberikan beberapa kontribusi yang berbeda. Shimpi dkk. [12] hanya menggunakan EAR tanpa HPE dan tanpa IoT. Florez dkk. [15] mengombinasikan EAR dan MAR namun tanpa HPE maupun cloud integration. Chen dkk. [14] mengimplementasikan multi-fitur termasuk HPE pada Raspberry Pi 4 yang lebih mahal, tanpa integrasi IoT. Penelitian ini melengkapi gap tersebut dengan mengombinasikan EAR dan HPE pada Raspberry Pi 3B+ yang lebih terjangkau sekaligus terintegrasi firebase untuk pemantauan real-time jarak jauh. Penggunaan 68 landmark dlib yang sama untuk kedua indikator menjadi keunggulan efisiensi komputasi yang membedakan pendekatan ini dari sistem yang menggunakan modul terpisah untuk setiap indikator [12], [14].

Selain itu, integrasi firebase dengan struktur log yang membedakan jenis indikator pemicu menjadikan sistem ini lebih informatif secara data dibandingkan sistem yang hanya menyimpan status kantuk biner. Dari sisi biaya implementasi, Raspberry Pi 3 Model B+ yang digunakan dalam penelitian ini memiliki harga sekitar seperlima dari Raspberry Pi 4, sehingga pendekatan ini membuka peluang adopsi yang lebih luas di kalangan operator transportasi skala kecil maupun menengah di Indonesia.

Secara keseluruhan, kombinasi EAR, HPE, dan firebase IoT pada platform Raspberry Pi 3B+ yang terjangkau merupakan kontribusi yang membedakan penelitian ini dari seluruh penelitian terdahulu yang dikaji, dan menjawab ketiga gap yang diidentifikasi pada bagian Pendahuluan. Implikasi praktis dari temuan ini cukup signifikan. Sistem yang dikembangkan berpotensi diimplementasikan sebagai solusi keselamatan berkendara yang terjangkau dan dapat diakses oleh pengemudi profesional maupun individu di Indonesia, tanpa memerlukan perangkat keras mahal atau infrastruktur yang kompleks

4. KESIMPULAN

Penelitian ini berhasil mengembangkan prototipe sistem pendeteksi kantuk pengemudi secara real-time berbasis EAR dan HPE pada Raspberry Pi 3 Model B+ dengan integrasi firebase IoT dalam kondisi pengujian di dalam ruangan terkontrol. Menjawab pertanyaan penelitian pertama, threshold EAR 0,25 dan pitch 15° yang dikalibrasi empiris terbukti efektif membedakan kondisi waspada dari kantuk pada keempat subjek yang diuji, dengan margin diskriminasi yang memadai dan tanpa false alarm pada kedipan normal. Menjawab pertanyaan kedua, pendekatan multi-indikator menghasilkan peringatan yang lebih awal dibandingkan indikator tunggal, di mana HPE terpicu



rata-rata 0,7 detik lebih awal dari EAR pada skenario kantung gradual, meskipun temuan ini bersifat observasional dari sampel terbatas dan belum divalidasi dengan dataset standar sehingga generalisasinya belum dapat dikonfirmasi secara saintifik. Menjawab pertanyaan ketiga, sistem berhasil beroperasi mandiri pada Raspberry Pi 3B+ dan mengirimkan data ke Firebase secara real-time dengan waktu respons rata-rata 2,00 detik EAR dan 2,70 detik HPE, keduanya dalam batas yang dapat diterima untuk peringatan dini. Meskipun demikian, penelitian ini memiliki keterbatasan teknis yang tidak dapat diabaikan: sistem belum diuji pada kondisi pencahayaan rendah meski menggunakan kamera NoIR, performa pada pengemudi berkacamata belum divalidasi secara eksplisit, threshold tunggal EAR 0,25 rentan terhadap variasi anatomis individu, dan seluruh pengujian dilakukan dalam simulasi laboratorium sehingga validitasnya pada kondisi berkendara nyata masih memerlukan pengujian lapangan yang lebih komprehensif dengan subjek yang lebih beragam dan dataset terstandar.

REFERENCES

- [1] D. Andarini, A. Camelia, and M. M. Ibrahim, “Factors related to road accidents in Palembang, South Sumatera, Indonesia,” *International Journal of Public Health Science (IJPHS)*, vol. 10, no. 3, p. 638, Sep. 2021, doi: 10.11591/ijphs.v10i3.20768.
- [2] H. Pranoto, A. Adriansyah, D. Feriyanto, A. Wahab, and S. Zakaria, “Propose Safety Engineering Concept Speed Limiter And Fatigue Control Using SLIFA For Truck And Bus,” *SINERGI*, vol. 24, no. 3, pp. 237–244, Jul. 2020, doi: 10.22441/sinergi.2020.3.009.
- [3] R. Zuraida and B. S. Abbas, “The Factors Influencing Fatigue Related to the Accident of Intercity Bus Drivers in Indonesia,” *International Journal of Technology*, vol. 11, no. 2, p. 342, Apr. 2020, doi: 10.14716/ijtech.v11i2.3792.
- [4] M. Mansyur, R. Sagitasari, G. Wangge, A. B. Sulistomo, and A. Kekalih, “Long working hours, poor sleep quality, and work-family conflict: determinant factors of fatigue among Indonesian tugboat crewmembers,” *BMC Public Health*, vol. 21, no. 1, p. 1832, Dec. 2021, doi: 10.1186/s12889-021-11883-6.
- [5] S. Soares, T. Monteiro, A. Lobo, A. Couto, L. Cunha, and S. Ferreira, “Analyzing Driver Drowsiness: From Causes to Effects,” *Sustainability*, vol. 12, no. 5, p. 1971, Mar. 2020, doi: 10.3390/su12051971.
- [6] M. A. Puspasari et al., “Prediction of drowsiness using EEG signals in young Indonesian drivers,” *Heliyon*, vol. 9, no. 9, p. e19499, Sep. 2023, doi: 10.1016/j.heliyon.2023.e19499.
- [7] A. A. D. P. Dewi, “Investigating Motorists Perceptions towards Road Safety,” *Civil Engineering and Architecture*, vol. 9, no. 5, pp. 1339–1346, Aug. 2021, doi: 10.13189/cea.2021.090507.
- [8] T. Suryadi and K. Kulsum, “Forensic investigation in a multiple trauma deceleration case due to road traffic accidents,” *Bali Medical Journal*, vol. 9, no. 1, pp. 172–177, Apr. 2020, doi: 10.15562/bmj.v9i1.1728.
- [9] W. Suprihatiningsih, A. M. Leman, D. Feriyanto, H. Pranoto, and N. Afizi Shuaib, “Impact on Driver Behavior, Performance, Fatigue, Over Speed and Infrastructure as A Factor of Road Accident: A Review,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 864, no. 1, p. 012105, May 2020, doi: 10.1088/1757-899X/864/1/012105.
- [10] M. A. Khan, T. Nawaz, U. S. Khan, A. Hamza, and N. Rashid, “IoT-Based Non-Intrusive Automated Driver Drowsiness Monitoring Framework for Logistics and Public Transport Applications to Enhance Road Safety,” *IEEE Access*, vol. 11, pp. 14385–14397, 2023, doi: 10.1109/ACCESS.2023.3244008.
- [11] Raspberry Pi Ltd, “Raspberry Pi Documentation,” [raspberrypi.com](https://www.raspberrypi.com/documentation/). Accessed: Jul. 13, 2026. [Online]. Available: <https://www.raspberrypi.com/documentation/>
- [12] R. Shimpi, V. Chaudhari, P. Patil, B. Narkhede, V. D. Chaudhari, and M. N. Patil, “Driver Drowsiness Detection System Using Raspberry Pi,” *International Journal of Innovations in Engineering and Science*, vol. 10, no. 4, pp. 22–25, Jun. 2025, doi: 10.46335/IJIES.2025.10.4.5.
- [13] E. Civik and U. Yuzgec, “Real-time driver fatigue detection system with deep learning on a low-cost embedded system,” *Microprocess. Microsyst.*, vol. 99, p. 104851, Jun. 2023, doi: 10.1016/j.micpro.2023.104851.
- [14] Z. Chen, D. Shi, Y. Guo, and X. Fu, “Real-Time Multi-Feature Driver Drowsiness Detection Based on Facial Landmarks Using LSTM on Raspberry Pi,” *Sensors*, vol. 25, no. 3, p. 920, 2025, doi: 10.3390/s25030920.
- [15] R. Florez, F. Palomino-Quispe, A. B. Alvarez, R. J. Coaquira-Castillo, and J. C. Herrera-Levano, “A Real-Time Embedded System for Driver Drowsiness Detection Based on Visual Analysis of the Eyes and Mouth Using Convolutional Neural Network and Mouth Aspect Ratio,” *Sensors*, vol. 24, no. 19, p. 6261, Sep. 2024, doi: 10.3390/s24196261.
- [16] S. Das, S. Pratihar, B. Pradhan, R. H. Jhaveri, and F. Benedetto, “IoT-Assisted Automatic Driver Drowsiness Detection through Facial Movement Analysis Using Deep Learning and a U-Net-Based Architecture,” *Information*, vol. 15, no. 1, p. 30, Jan. 2024, doi: 10.3390/info15010030.
- [17] R. Kaur and S. Bhatt, “Design and Implementation of Drowsiness Detection System for Driver Safety Using Facial Landmarks,” *Int. J. Comput. Appl.*, vol. 184, no. 7, pp. 1–6, 2022, doi: 10.5120/ijca2022921934.
- [18] M.-M. Naddaf-Sh, A. Lee, K. Yen, E. Amini, and I. Soltani, “Low-Cost Infrared Vision Systems for Improved Safety of Emergency Vehicle Operations Under Low-Visibility Conditions,” *IEEE Access*, vol. 13, pp. 173241–173253, 2025, doi: 10.1109/ACCESS.2025.3615117.
- [19] N. Lin and Y. Zuo, “Advancing driver fatigue detection in diverse lighting conditions for assisted driving vehicles with enhanced facial recognition technologies,” *PLoS One*, vol. 19, no. 7, p. e0304669, Jul. 2024, doi: 10.1371/journal.pone.0304669.
- [20] K. A. Obayes and A. Hamzah, “Using of prototyping in develop an employee information management,” *Measurement: Sensors*, vol. 24, p. 100557, Dec. 2022, doi: 10.1016/j.measen.2022.100557.
- [21] C. Chen, H. Gu, S. Lian, Y. Zhao, and B. Xiao, “Investigation of Edge Computing in Computer Vision-Based Construction Resource Detection,” *Buildings*, vol. 12, no. 12, p. 2167, Dec. 2022, doi: 10.3390/buildings12122167.
- [22] D. N. Bestari and A. Wibowo, “IoT Based Real-Time Weather Monitoring System Using Telegram Bot and Thingsboard Platform,” *International Journal of Interactive Mobile Technologies (IJIM)*, vol. 17, no. 06, pp. 4–19, Mar. 2023, doi: 10.3991/ijim.v17i06.34129.



- [23] S. Ayas, B. Donmez, and X. Tang, “Drowsiness Mitigation Through Driver State Monitoring Systems: A Scoping Review,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 66, no. 9, pp. 2218–2243, Sep. 2024, doi: 10.1177/00187208231208523.