

BAB II

TINJAUAN PUSTAKA

2.1 Keselamatan Berkendara dan Kecelakaan Lalu Lintas

Kecelakaan lalu lintas merupakan peristiwa tidak terduga dan tidak disengaja di jalan yang melibatkan kendaraan, yang dapat menyebabkan kerugian harta benda atau korban jiwa. UU No. 22 Tahun 2009 mendefinisikan kecelakaan sebagai kejadian di jalan yang melibatkan kendaraan dengan atau tanpa pengguna jalan lain, dan mengakibatkan korban manusia atau kerugian harta benda. Berdasarkan tingkat keparahannya, kecelakaan dapat dikategorikan menjadi ringan, sedang, dan berat. Sedangkan berdasarkan jenis tabrakannya, dapat berupa tabrakan depan-belakang, samping, kecelakaan tunggal, dan kendaraan terguling (Sri Susanti et al., 2024).

Keselamatan berkendara merujuk pada upaya meminimalkan risiko kecelakaan melalui pendekatan multidimensi. Peraturan Pemerintah No. 37 Tahun 2017 menyebutkan bahwa aspek keselamatan mencakup kondisi fisik dan mental pengemudi, kelayakan teknis kendaraan, kepatuhan terhadap peraturan lalu lintas, serta faktor lingkungan seperti kondisi jalan, cuaca, dan kepadatan lalu lintas (Manullang et al., 2023).

Berdasarkan hasil analisis kecelakaan lalu lintas di Indonesia (Santosa et al., 2017) menyatakan bahwa sekitar 88% kejadian kecelakaan disebabkan oleh faktor manusia, sementara faktor lingkungan dan kendaraan masing-masing berkontribusi sebesar 8% dan 3%. Temuan tersebut mengindikasikan bahwa aspek perilaku dan

kondisi pengemudi merupakan faktor dominan dalam penyebab kecelakaan lalu lintas di Indonesia.

Mengingat dominasi faktor manusia sebagai penyebab utama kecelakaan, berbagai pendekatan teknologi telah dikembangkan untuk mendeteksi kejadian kecelakaan secara otomatis. Salah satu pendekatan yang banyak digunakan adalah pemanfaatan sensor akselerometer dan giroskop (Dilshad et al., 2018) menyatakan bahwa percepatan kendaraan yang melebihi 4 g dapat dijadikan indikator kuat terjadinya tabrakan. Sementara itu, (Ilahi, 2025) menyatakan bahwa kemiringan kendaraan yang melebihi 45 derajat dapat digunakan sebagai parameter untuk mendeteksi kecelakaan jenis rollover, karena sudut tersebut tidak ditemukan dalam kondisi jalan normal yang umumnya berada di bawah 17,5 derajat.

2.2 Faktor Manusia dalam Kecelakaan

Beberapa perilaku manusia terbukti berkontribusi langsung terhadap terjadinya kecelakaan lalu lintas. Mengemudi dalam pengaruh alkohol secara signifikan meningkatkan risiko kecelakaan fatal. Kecepatan berkendara yang melebihi batas juga memperbesar potensi terjadinya kecelakaan serius. Selain itu, tidak menggunakan sabuk pengaman dapat meningkatkan risiko kematian, terutama bagi penumpang di kursi depan. Pelanggaran terhadap rambu dan aturan lalu lintas pun menjadi penyebab utama kecelakaan yang sering terjadi di jalan raya (Cociu et al., 2023).

Penelitian (Widyastuti et al., 2024) menyatakan bahwa kantuk memiliki hubungan yang signifikan terhadap peningkatan risiko kecelakaan lalu lintas, khususnya pada malam hingga dini hari, terutama ketika pengemudi berkendara

lebih dari lima jam tanpa istirahat. Selain itu, distraksi pengemudi menjadi faktor signifikan yang menyebabkan kecelakaan. Berdasarkan analisis data, sekitar 60% kecelakaan lalu lintas di Indonesia berkaitan dengan perilaku pengemudi, termasuk distraksi (Sutanto et al., 2022).

Berdasarkan proporsi kontribusinya terhadap kecelakaan dan potensi deteksinya melalui pendekatan visual, kantuk dan distraksi dipilih sebagai fokus utama dalam penelitian ini. Keduanya mewakili bentuk kegagalan persepsi dan atensi pengemudi yang dapat diamati secara langsung oleh kamera.

2.2.1 Pengaruh Kantuk terhadap Pengemudi

Menurut data Badan Pusat Statistik (2022), lebih dari 25% kecelakaan lalu lintas di Indonesia disebabkan oleh kelelahan yang menyebabkan pengemudi tertidur saat mengemudi. Kantuk merupakan kondisi fisiologis yang ditandai dengan penurunan kewaspadaan, refleks tubuh, dan dorongan kuat untuk tidur. Dalam konteks berkendara, kondisi ini berbahaya karena dapat menyebabkan hilangnya kesadaran secara tiba-tiba dan menurunkan kemampuan pengemudi dalam merespons situasi lalu lintas (Kurniawan et al., 2023).

Apabila tidak segera dikenali dan ditangani, kantuk dapat berkembang menjadi *Microsleep* yaitu kondisi tidur pendek berkisar antara 1 sampai 30 detik, dimana orang yang mengalami kondisi ini gagal merespon sensor motoric dan menjadi tidak sadarkan diri. Microsleep dapat menciptakan situasi berbahaya, misalnya ketika seorang pengguna mengendarai mobil dapat mengakibatkan situasi yang tidak aman atau bahkan kematian (Sugiharto et al., 2019).

Penutupan mata selama dua detik dijadikan indikator bahwa pengemudi mengalami kondisi mengantuk, karena kedipan mata normal umumnya hanya berlangsung 0,1 hingga 0,3 detik, sehingga durasi dua detik menunjukkan adanya kantuk serius (Kongcharoen et al., 2020). Selain itu, ekspresi menguap juga merupakan indikator visual yang relevan dalam mendeteksi kantuk. Menguap ditandai dengan pembukaan mulut secara lebar selama 1 hingga 3 detik dan dapat dikenali melalui analisis citra wajah (Majeed et al., 2023).

2.2.2 Pengaruh Distraksi terhadap Pengemudi

Driving distraction merupakan gangguan yang menyebabkan perhatian pengemudi teralihkan dari kegiatan utama mengemudi menjadi aktivitas lain di luar tugas tersebut. Ketika pengemudi mengalami gangguan semacam ini, fokusnya terbagi menjadi *primary task*, yaitu mengemudi, dan *secondary task* berupa aktivitas lain yang tidak berkaitan dengan mengemudi. Distraksi ini dapat terjadi dalam bentuk penggunaan *handphone* maupun makan dan minum saat mengemudi, yang keduanya memiliki probabilitas tertinggi sebagai penyebab kecelakaan lalu lintas (Ismail, 2019).

Penggunaan *smartphone* saat berkendara merupakan salah satu bentuk *Driving distraction* yang sangat berbahaya karena secara signifikan menurunkan fokus pengemudi dari aktivitas utama ke aktivitas sekunder. Penggunaan *smartphone* selama 45 detik berisiko terjadinya kecelakaan mencapai 100%, untuk jalan dua lajur dengan *traffic peak hour*. Penggunaan *smartphone* saat berkendara lebih dari 10 detik pada jalan 2 atau 4 lajur dengan kondisi lalu lintas sepi maupun sibuk menunjukkan risiko kecelakaan 65-100% (Asyari et al., 2022). Penelitian ini difokuskan pada deteksi distraksi visual akibat penggunaan ponsel karena

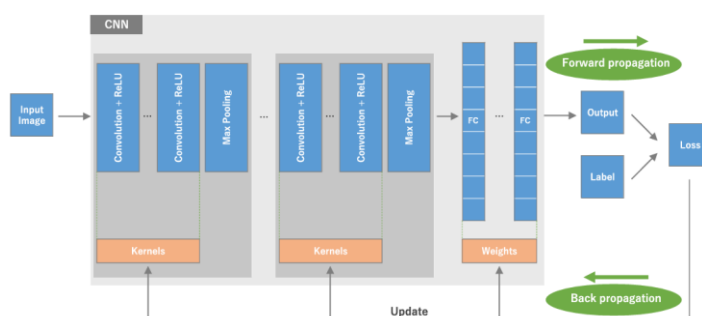
merupakan jenis gangguan paling umum, memiliki indikator visual yang mudah diamati.

2.3 Deteksi Objek Visual Menggunakan *Deep Learning*

Deteksi objek berbasis citra visual penting dalam sistem pemantauan pengemudi untuk mengenali kondisi seperti kantuk dan distraksi secara real-time. Pendekatan *deep learning*, khususnya *Convolutional Neural Network* (CNN), memungkinkan ekstraksi fitur visual secara otomatis dan akurat tanpa perancangan manual. CNN efektif dalam mengenali objek kecil seperti mata dan ponsel. Berbagai metode deteksi objek telah dikembangkan, mulai dari pendekatan konvensional seperti Haar Cascade hingga metode modern berbasis CNN seperti SSD, Faster-RCNN, dan YOLO, yang memiliki karakteristik berbeda dalam hal akurasi, kecepatan, dan efisiensi.

Convolutional Neural Network (CNN) merupakan metode *deep learning* yang dirancang untuk mengenali pola visual dari data citra secara otomatis. CNN banyak digunakan dalam pengolahan citra karena tidak memerlukan rekayasa fitur manual, berbeda dengan metode tradisional seperti Gabor filter atau SIFT. Arsitektur CNN secara umum terdiri dari tiga proses utama yaitu *convolution* untuk mengekstraksi fitur visual, fungsi aktivasi *non-linear* seperti ReLU untuk mengenali pola kompleks, dan *pooling* yang mereduksi dimensi citra guna mengurangi beban komputasi tanpa kehilangan informasi penting. Setelah fitur-fitur ini diproses, lapisan *fully connected* menghasilkan output akhir berupa hasil klasifikasi.

Kemampuan CNN dalam mengenali fitur berlangsung secara bertahap dan hierarkis mulai dari pola sederhana seperti tepi dan sudut, hingga objek kompleks seperti wajah atau mata. Keunggulan ini menjadikan CNN sangat cocok diterapkan dalam sistem deteksi objek. Arsitektur Dasar CNN ditampilkan pada Gambar 2.1.



Gambar 2.1 Arsitektur Dasar Convolutional Neural Network (CNN)

Sumber: Yamashita et al. (2018)

Keunggulan CNN dalam mengenali fitur visual secara otomatis telah mendorong pengembangan berbagai metode deteksi objek dalam visi komputer. Deteksi objek merupakan bagian penting dalam visi komputer yang bertujuan mengenali dan menentukan lokasi objek dalam citra atau video, dengan aplikasi luas seperti pemantauan pengemudi dan kendaraan otonom. Pendekatan awal menggunakan metode konvensional seperti Haar Cascade dan HOG-SVM, namun terbatas karena bergantung pada rekayasa fitur manual dan kurang tangguh terhadap variasi pencahayaan maupun latar belakang. Perkembangan deep learning dengan arsitektur CNN memungkinkan ekstraksi fitur otomatis yang lebih akurat dan efisien. Algoritma seperti SSD dan Faster R-CNN meningkatkan performa deteksi, tetapi masing-masing masih memiliki kekurangan dari sisi akurasi terhadap objek kecil atau kecepatan inferensi. YOLO (You Only Look Once) hadir sebagai

solusi yang menyatukan proses klasifikasi dan regresi dalam satu tahap, sehingga mampu melakukan deteksi secara real-time dengan efisiensi tinggi. YOLO menjadi salah satu algoritma paling optimal karena menawarkan keseimbangan antara kecepatan, akurasi, dan kesederhanaan implementasi, serta sangat cocok diterapkan pada perangkat embedded seperti Raspberry Pi (Edozie et al., 2025).

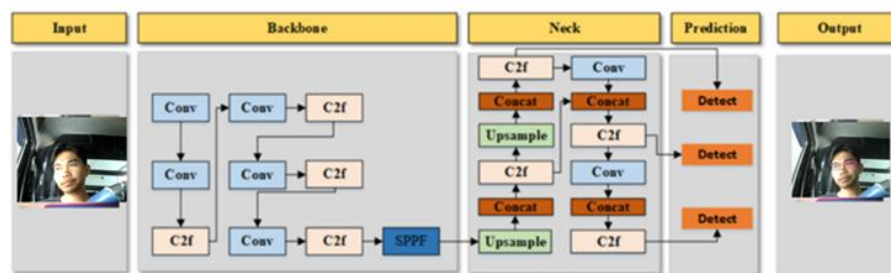
2.4 YOLO

YOLO (You Only Look Once) merupakan algoritma deteksi objek satu tahap yang terus dikembangkan dari versi YOLOv1 hingga YOLOv7, dengan fokus pada kecepatan dan akurasi. Namun, versi sebelumnya masih menggunakan pendekatan anchor-based yang kompleks. YOLOv8 hadir sebagai penyempurnaan dengan desain anchor-free, Unified Detection Head, serta dukungan mixed precision training, sehingga lebih efisien dan fleksibel untuk berbagai bentuk objek (Yaseen, 2024). Versi ringan YOLOv8n mencapai mAP sebesar 80,9% dengan waktu inferensi 24,2 ms, menjadikannya sesuai untuk aplikasi real-time dan perangkat terbatas (Gamani et al., 2024).

2.4.1 Arsitektur YOLOv8

YOLOv8 mengusung arsitektur modular yang terdiri dari tiga komponen utama yaitu backbone CSPNet, neck FPN+PAN, dan head anchor-free. CSPNet pada bagian backbone digunakan untuk meningkatkan efisiensi ekstraksi fitur dan mengurangi redundansi komputasi, sedangkan neck menggabungkan informasi dari berbagai level melalui versi yang disempurnakan dari PANet, memungkinkan deteksi multi-skala yang lebih baik. Inovasi utama terletak pada head anchor-free, yang menghilangkan kebutuhan anchor box manual dan menyederhanakan proses prediksi bounding box. Pendekatan ini meningkatkan fleksibilitas terhadap variasi

bentuk dan ukuran objek, serta mempercepat proses pelatihan tanpa menurunkan akurasi. Selain itu, YOLOv8 menerapkan struktur loss function yang lebih adaptif, terdiri dari *Focal Loss* untuk mengatasi ketidakseimbangan kelas, *IoU Loss* untuk meningkatkan presisi prediksi lokal, untuk mengarahkan model hanya pada area relevan dalam citra. Peningkatan ini menjadikan YOLOv8 unggul dalam mendeteksi objek kecil dan tumpang tindih secara lebih konsisten. Dengan optimalisasi struktur arsitektur ini, YOLOv8 mampu mempertahankan kecepatan inferensi tinggi dan akurasi yang kompetitif untuk aplikasi real-time (Yaseen 2024). Ilustrasi mengenai cara kerja YOLO sebagaimana ditunjukkan pada Gambar 2.2.



Gambar 2.2 Proses Deteksi Objek YOLO

Pemilihan parameter pelatihan berperan penting dalam menentukan performa dan efisiensi sistem deteksi objek secara real-time, terutama pada perangkat terbatas seperti Raspberry Pi. Dalam penelitian yang dilakukan oleh (Gamani et al., 2024), lima konfigurasi YOLOv8 yaitu YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, dan YOLOv8x dibandingkan untuk tugas segmentasi instance buah stroberi di lingkungan terbuka. Masing-masing model dilatih menggunakan resolusi input sebesar 640×640 piksel, batch size 16, dan learning rate 0.01 dengan algoritma optimisasi Stochastic Gradient Descent. Proses

pelatihan dilakukan maksimal selama 100 epoch dengan mekanisme early stopping apabila tidak terdapat peningkatan validasi dalam 30 epoch berturut-turut. Berdasarkan hasil evaluasi, YOLOv8n menunjukkan performa terbaik dengan nilai mAP sebesar 80,9% dan waktu inferensi tercepat sebesar 24,2 milidetik, sehingga direkomendasikan untuk implementasi sistem real-time pada perangkat terbatas seperti Raspberry Pi.

Berdasarkan hasil penelitian yang dilakukan oleh (Apeināns, 2024), Peningkatan jumlah data pelatihan dari 100 hingga 500 gambar menghasilkan peningkatan akurasi yang signifikan pada model YOLOv8, dengan rata-rata peningkatan mAP@0.5 mencapai 13–19%. Sebaliknya, penambahan data dari 500 hingga 1000 gambar hanya memberikan peningkatan akurasi sekitar 2–3%. Temuan ini menunjukkan bahwa jumlah data sebanyak 500 gambar merupakan pilihan yang efisien dan seimbang untuk pelatihan model YOLOv8, baik dari segi performa maupun beban pengumpulan data.

Sejalan dengan hal tersebut, Sejalan dengan hal tersebut, meskipun rasio pembagian data 80:20 dan 90:10 sering kali menghasilkan akurasi yang tinggi, penelitian oleh (Bichri et al., 2024) menunjukkan bahwa penggunaan 70% data untuk pelatihan tetap memberikan performa yang sangat baik dan stabil. Rasio ini dinilai optimal, terutama dalam kondisi jumlah data terbatas atau saat diperlukan alokasi data untuk proses validasi dan pengujian. Dengan kata lain, proporsi pelatihan sebesar 70% sudah memadai untuk membentuk model yang mampu melakukan generalisasi secara efektif terhadap data baru.

2.4.2 Evaluasi Model YOLO

Evaluasi performa model deteksi objek dilakukan dengan menggunakan sejumlah metrik standar, yaitu *accuracy*, *precision*, *recall*, *average precision (AP)*, *mean average precision (mAP)*, serta *frames per second (FPS)*. *mAP* merupakan metrik yang paling umum digunakan dalam evaluasi model deteksi objek. Nilai ini diperoleh dengan menghitung rata-rata dari nilai *AP* setiap kategori, dan meringkas keseimbangan antara *precision* dan *recall* dalam satu angka. Dua standar *mAP* yang sering digunakan adalah *mAP@0.5*, yaitu *mAP* pada ambang batas *Intersection over Union (IoU)* $\geq 0,5$, dan *mAP@0.5–0.95*, yaitu rata-rata *mAP* dari berbagai ambang batas *IoU* (mulai dari 0,5 hingga 0,95 dengan interval 0,05). *Precision* mengukur proporsi deteksi positif yang benar, sedangkan *recall* mengukur sejauh mana model berhasil menemukan semua objek yang seharusnya terdeteksi. *FPS (frames per second)* digunakan untuk menilai seberapa cepat model dapat memproses citra secara *real-time* (Zhou et al., 2024).

2.5 Edge Computing

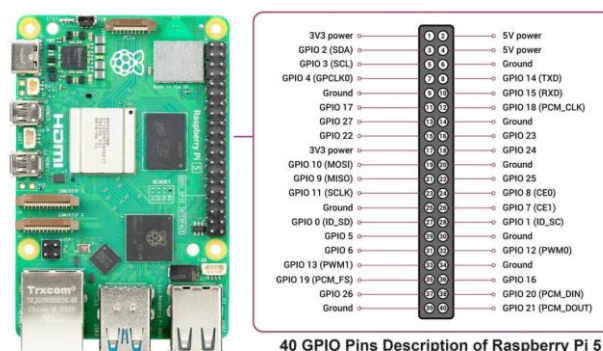
Edge Computing adalah komputasi di mana pengolahan data dilakukan di dekat sumber data, yaitu pada perangkat atau node yang terletak di jaringan, daripada mengirim semua data ke cloud untuk diproses. Dengan mendekatkan ini, Edge Computing dapat mengurangi latensi, menghemat bandwidth, dan meningkatkan efisiensi energi (Suryadi et al., 2024).

Pada sistem deteksi kondisi pengemudi, edge computing memungkinkan proses inferensi model YOLO dilakukan langsung secara lokal sehingga sistem dapat beroperasi secara mandiri tanpa ketergantungan penuh pada server pusat. Raspberry Pi digunakan sebagai perangkat komputasi tepi karena memiliki kinerja

yang memadai untuk menjalankan YOLO dengan latensi rendah, konsumsi daya yang efisien, serta kemudahan integrasi dengan kamera dan media penyimpanan (Ali Farooq et al., 2023).

2.6 Raspberry Pi

Raspberry Pi merupakan komputer papan tunggal (*single-board computer*) yang banyak digunakan dalam sistem pemrosesan data pada perangkat tepi (*edge computing*). Perangkat ini mampu menjalankan model deteksi objek, termasuk model berbasis YOLO, *secara real-time* dengan konsumsi daya rendah, meskipun digunakan dalam lingkungan dengan sumber daya terbatas (Silva et al., 2024). Raspberry Pi 5 dipilih dalam penelitian ini karena mampu menjalankan deteksi visual secara lokal tanpa bergantung pada layanan komputasi *cloud*, dengan dukungan prosesor ARM Cortex-A76, RAM 8 GB, serta konektivitas Gigabit Ethernet dan Wi-Fi. Perangkat ini juga mendukung akselerator eksternal seperti Hailo-8L dan memiliki efisiensi daya tinggi, sehingga menjadi platform yang sesuai untuk sistem deteksi kantuk dan distraksi berbasis pengolahan citra *secara real-time*. Diagram pinout GPIO Raspberry Pi ditampilkan pada Gambar 2.3.



Gambar 2.3 GPIO Raspberry Pi 5

2.7 Webcam

Webcam digunakan sebagai sensor utama dalam sistem ini untuk menangkap citra wajah pengemudi yang kemudian dianalisis menggunakan model YOLO guna mendeteksi tanda-tanda kantuk dan distraksi. Perangkat yang digunakan adalah Logitech C270 dengan resolusi 720p pada 30 frame per detik serta sudut pandang sekitar 60 derajat. Kamera ini dihubungkan ke Raspberry Pi 5 melalui port USB, dipilih karena lebih hemat biaya, mudah diintegrasikan, dan mampu mendukung proses pemantauan kondisi pengemudi secara real-time. Meskipun kamera inframerah dapat bekerja lebih baik pada pencahayaan rendah, keterbatasan anggaran dan pertimbangan implementasi praktis (Ezzouhri et al., 2021). Sebagaimana ditunjukkan pada Gambar 2.4.



Gambar 2.4 Webcam

(Sumber: www.logitech.com)

2.8 Pengiriman Data

Dalam sistem Internet of Things (IoT), pemilihan metode pengiriman data menjadi faktor krusial dalam menjamin keandalan sistem, khususnya pada lingkungan dengan koneksi jaringan yang tidak stabil seperti kendaraan bergerak.

Secara umum, terdapat empat pendekatan pengiriman data, yaitu pengiriman waktu nyata (*real-time*), pengiriman berkala (*batch*), simpan dan teruskan (*store-and-forward*), serta pendekatan hibrida.

Salah satu pendekatan yang banyak digunakan pada kondisi jaringan tidak stabil adalah Metode *store-and-forward*, *carry and forward* adalah teknik untuk mengirimkan informasi melalui perantara dengan cara menyimpan data terlebih dahulu, kemudian dikirimkan kembali beberapa saat berikutnya menuju tujuan akhir. Teknik ini umum digunakan dalam Delay Tolerant Network karena pengiriman langsung ke endhost sering terkendala oleh keterlambatan transmisi atau tidak tersedianya koneksi. Pada implementasinya, penyimpanan dilakukan dengan basis data SQLite yang memiliki kapasitas hingga sekitar 140 terabyte, sehingga data tetap tersimpan dengan baik sebelum diteruskan (Tobing et al., 2020).

2.9 Analisis Penyebab Kecelakaan

Penentuan penyebab kecelakaan umumnya dilakukan melalui dua pendekatan, yaitu kecerdasan buatan dan aturan logika. Pendekatan AI mengandalkan model pembelajaran mesin yang memerlukan data historis besar dan komputasi tinggi. Sebaliknya, rule-based menggunakan aturan eksplisit berbasis logika atau pengetahuan domain, sehingga lebih ringan, cepat, dan mudah dijelaskan. Dalam penelitian ini, pendekatan rule-based dipilih karena sistem menganalisis data status pengemudi yang telah terstruktur dalam jendela waktu 10 detik sebelum kecelakaan. Menurut (Darveau et al., 2020), pendekatan ini efektif ketika indikator keputusan dapat dirumuskan secara deterministik tanpa memerlukan generalisasi kompleks seperti pada AI. Adapun pemilihan durasi 10 detik merujuk pada temuan (Ahmad et al., 2023), yang menunjukkan bahwa

gangguan visual selama 10 detik meningkatkan probabilitas kecelakaan dari 16% menjadi 29%. Oleh karena itu, interval ini dianggap kritis untuk deteksi perilaku mengemudi berisiko secara real-time dan mendukung pengambilan keputusan yang tepat.

2.10 Penelitian Terdahulu dan Keterbaruan Penelitian

Berbagai penelitian telah dilakukan untuk mengembangkan sistem deteksi kondisi pengemudi, khususnya terkait kantuk dan distraksi saat berkendara. Salah satu pendekatan yang banyak digunakan adalah model deteksi berbasis YOLO (You Only Look Once), yang dikenal memiliki kecepatan dan akurasi tinggi untuk aplikasi real-time. Ringkasan beberapa penelitian yang relevan disajikan pada Tabel 2.1.

Tabel 2.1 Penelitian Terkait

No	Judul Jurnal	Nama Peneliti	Tempat Dan Tahun Penelitian	Pembahasan Jurnal
1	Utilization of YOLOv8 Algorithm for In-Vehicle Video-Based Driver Monitoring System	M. Rajendra Aria Satya, Odi Nurdiawan, Fadhil M. Basysyar, Rahmat Hidayat	STMIK IKMI Cirebon, Indonesia , 2025.	Penelitian ini Mendeteksi perilaku pengemudi berbasis video menggunakan YOLOv8 pada dataset DMS Driver Monitoring System. Model dilatih

				selama 100 epoch dengan hasil Precision 89,6%, Recall 87,2%, F1-score 88,0%, mAP@0.5 sebesar 92,0%, dan mAP@0.5–0.95 sebesar 69,1%. Dalam pengujian video real-time, model mendeteksi mata terbuka dengan confidence 85% dan sabuk pengaman dengan confidence 35%. Sistem menunjukkan performa tinggi untuk deteksi standar, tetapi perlu peningkatan untuk skenario kompleks
--	--	--	--	---

				seperti pencahaya- an rendah atau sudut kamera ekstrem.
2	Driver Distraction and Fatigue Detection in Images using ME-YOLOv8 Algorithm	Ali Debsi, Guo Ling, Mohammed Al-Mahbashi, Mohammed Al-Soswa, Abdulkareem Abdullah	Nanjing University of Science and Technology, Chang'an University (Tiongkok), University of Aleppo (Syria), Tahun 2024	Penelitian ini mengembangkan model ME- YOLOv8, yakni modifikasi dari YOLOv8 untuk meningkatkan akurasi deteksi perilaku berbahaya pengemudi secara real-time, seperti mengantuk, menggunakan ponsel. Model ini menunjukkan kinerja tinggi dengan mAP@0.5 sebesar 97,9% dan mAP@0.5–0.95 sebesar 75,3%,

				serta mampu mendeteksi gangguan secara akurat pada uji langsung di jalan, baik siang (92,3%) maupun malam hari (90,2%).
3	Real-Time and Deep Learning-Based Fatigue Detection for Drivers	Abdil Karakan	Afyon Kocatepe University, Türkiye, 2024	penelitian ini membahas pengembangan sistem deteksi kelelahan pengemudi secara <i>real-time</i> berbasis <i>deep learning</i> . Sistem ini memantau area mata dan mulut untuk mengklasifikasikan kondisi pengemudi ke dalam kategori normal, menguap,

				dan lelah. Data diperoleh dari 129 pengemudi dalam kondisi nyata, kemudian diproses menggunakan teknik <i>image augmentation</i> untuk meningkatkan akurasi deteksi. Studi ini membandingkan empat arsitektur YOLO, yaitu YOLOv5 dengan akurasi sebesar 94,15 %, YOLOv6 sebesar 95,24 %, YOLOv7 sebesar 96,48 %, dan YOLOv8 sebesar 97,50 %. Hasil terbaik dicapai oleh arsitektur YOLOv8.
--	--	--	--	--

4	Performa Model YOLOv8 untuk Deteksi Kondisi Mengantuk pada Pengendara Mobil	Edmund Ucok Armin, Husnul Amalia, Riska Asari, Febrian Dwi Cahyo	Universitas Singaperbangsa Karawang, Tahun 2023	Penelitian ini menggunakan YOLOv8 untuk mendeteksi kantuk pengemudi secara real-time melalui analisis citra wajah, khususnya area mata. Dari 3.708 citra berlabel "Awake" dan "Drowsy", model YOLOv8 diuji dan dibandingkan dengan SSD-Resnet, YOLOv3, dan YOLOv5. Hasilnya, YOLOv8 menunjukkan performa terbaik dengan precision 94,65%, recall 95,26%, F1-score 94,96%, dan mAP
---	--	---	--	--

				98,06%, menjadikannya model paling akurat untuk deteksi kelelahan pengemudi, meskipun FPS-nya sedikit di bawah SSD-Resnet.
5	Design of Emergency Alarm System for Drowsiness Detection Using YOLO Method Based on Raspberry Pi	Dinda Ayu Permatasari, Anindya Dwi Risdhayanti, Gillang Al Azhar, Arief Rahman Hidayat, Muhammad Rifqi Zharfan, Denda Dewatama	Politeknik Negeri Malang, Indonesia, 2024	Penelitian ini membahas sistem alarm darurat untuk deteksi kantuk pengemudi menggunakan YOLO dan Mini PC. Sistem ini mendeteksi kondisi kantuk berdasarkan deteksi wajah pengemudi dan memberi peringatan melalui buzzer, LCD, dan

				semprotan air. Sistem diuji dengan hasil akurasi 88% pada pencahayaan optimal dan 66,8% pada pencahayaan rendah.
--	--	--	--	--

Sebagian besar penelitian terdahulu hanya berfokus pada deteksi kondisi pengemudi seperti kantuk dan distraksi secara *real-time*, tanpa melanjutkan pada analisis penyebab kecelakaan. Deteksi yang dilakukan bersifat sesaat dan tidak merekam riwayat status pengemudi dalam rentang waktu tertentu. Belum ditemukan penelitian yang mengaitkan hasil deteksi tersebut dengan identifikasi penyebab kecelakaan secara sistematis.

Penelitian ini mengisi celah tersebut dengan mengembangkan sistem yang tidak hanya mendeteksi kondisi pengemudi, tetapi juga menganalisis distribusi status pengemudi sebelum kecelakaan. Dengan demikian, hasil deteksi digunakan untuk mengidentifikasi kemungkinan penyebab kecelakaan berdasarkan kondisi pengemudi dalam periode waktu tertentu. Pendekatan ini menjadi kontribusi baru yang belum ditemukan pada penelitian sebelumnya.