

Deteksi Kecelakaan Lalu Lintas Menggunakan YOLO11 pada Citra Kamera CCTV untuk Klasifikasi Jenis Kendaraan Terlibat

^{1st} I Ketut Yasa Seniarta

¹Fakultas Informatika

¹Universitas Telkom

¹Bandung, Indonesia

¹yasaseniarta@student.telkomuniversity.ac.id

^{2nd} Ema Rachmawati

²Fakultas Informatika

²Universitas Telkom

²Bandung, Indonesia

²emarachmawati@telkomuniversity.ac.id

Abstrak — Kecelakaan lalu lintas merupakan masalah serius dengan lebih dari 1,19 juta kematian setiap tahun di dunia dan 148.307 kasus di Indonesia pada 2023. Sistem pengawasan yang ada sebagian besar masih manual sehingga belum mampu mendeteksi dan mengklasifikasikan kecelakaan secara otomatis. Topik ini penting karena keterlambatan dalam deteksi kecelakaan dapat memperburuk dampak bagi korban. Solusi otomatis dapat mempercepat respons, mengurangi kerugian, serta mendukung sistem transportasi cerdas. Saat ini terdapat kesenjangan antara kebutuhan sistem otomatis dengan keterbatasan pendekatan manual yang dominan digunakan. Penelitian ini mengembangkan model deteksi kecelakaan berbasis YOLO11 untuk mengidentifikasi insiden serta mengklasifikasikan jenis kendaraan yang terlibat, yaitu *car_accident*, *car_motorbike_accident*, dan *motorbike_accident*. Model dilatih menggunakan 1.483 data citra dari berbagai sumber, kemudian dievaluasi dengan metrik *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95*. Percobaan lain dilakukan melalui strategi *fine-tuning* seperti *freeze layer*, augmentasi data, dan penggunaan *optimizer AdamW*. Model *baseline* menghasilkan performa terbaik dengan *precision* 0,82, *recall* 0,79, dan *mAP@0.5* sebesar 0,84. Setelah *fine-tuning*, kinerja model menjadi lebih stabil dengan *precision* 0,81, *recall* 0,72, *mAP@0.5* sebesar 0,75, dan *mAP@0.5:0.95* yang lebih konsisten. Hasil ini menunjukkan performa yang lebih seimbang antar kelas dan berpotensi mendukung pengawasan lalu lintas yang lebih cerdas.

Kata kunci— deteksi kecelakaan, klasifikasi, lalu lintas, YOLO11, *fine-tuning*

I. PENDAHULUAN

Kecelakaan lalu lintas merupakan salah satu permasalahan serius dalam sektor transportasi global. Berdasarkan *Global Status Report on Road Safety 2023* dari WHO, lebih dari 1,19 juta jiwa meninggal akibat kecelakaan lalu lintas setiap tahunnya di seluruh dunia pada tahun 2021 [1]. Target global untuk menurunkan angka kematian dan cedera hingga 50% pada tahun 2030 pun masih jauh dari tercapai. Di Indonesia, data dari *Integrated Road Safety Management System (IRSMS)* mencatat sebanyak 148.307 kecelakaan sepanjang tahun 2023 [2]. Kondisi ini menunjukkan bahwa kecelakaan lalu lintas masih menjadi ancaman utama terhadap keselamatan publik serta menimbulkan kerugian sosial dan ekonomi yang signifikan.

Upaya mitigasi telah dilakukan melalui pembangunan jalur khusus kendaraan roda dua, implementasi sistem

pemantauan *real-time*, dan pemasangan CCTV di area rawan kecelakaan [3]. Namun, efektivitas deteksi kecelakaan secara cepat masih terbatas, karena sebagian besar sistem pengawasan masih mengandalkan pemantauan manual [4]. Keterlambatan mendeteksi insiden berpotensi memperlambat penanganan darurat dan memperburuk dampak kecelakaan bagi korban maupun kelancaran lalu lintas.

Seiring kemajuan teknologi kecerdasan buatan (AI) dan pengolahan citra, pendekatan berbasis computer vision menawarkan solusi untuk mendeteksi kecelakaan secara otomatis. Model deteksi objek seperti YOLO (*You Only Look Once*) terbukti mampu bekerja *real-time* dengan akurasi tinggi [5], [6]. YOLOv8, telah digunakan untuk identifikasi kecelakaan dari kamera pengawas dalam kondisi pencahayaan beragam [7]. Namun, sebagian besar penelitian sebelumnya masih berfokus pada deteksi insiden tanpa memperhatikan klasifikasi jenis kendaraan yang terlibat. Padahal, informasi ini penting untuk mendukung kebijakan berbasis data, seperti pembatasan jalur kendaraan berat, desain infrastruktur, serta strategi mitigasi kecelakaan [8].

YOLO11 sebagai pengembangan terbaru menghadirkan peningkatan dalam efisiensi parameter, akurasi deteksi, dan kemampuan generalisasi [9], [10]. Dengan memanfaatkan keunggulan tersebut, penelitian ini bertujuan mengembangkan model deteksi kecelakaan lalu lintas berbasis YOLO11 yang tidak hanya mendeteksi insiden secara otomatis, tetapi juga mengklasifikasikan jenis kendaraan yang terlibat. Pendekatan ini diharapkan dapat menghasilkan model deteksi bermanfaat untuk mendukung keselamatan transportasi modern.

II. KAJIAN TEORI

A. Kecelakaan Lalu Lintas

Kecelakaan lalu lintas merupakan salah satu penyebab utama kematian di dunia. Laporan WHO dalam *Global Status Report on Road Safety 2023* melaporkan lebih dari 1,19 juta jiwa meninggal akibat kecelakaan lalu lintas pada tahun 2021 [1]. Di Indonesia, sistem IRSMS mencatat sebanyak 148.307 kecelakaan pada tahun 2023 [2], dengan sepeda motor terlibat pada sekitar 74% kasus. Tingginya angka kecelakaan, khususnya pada pengendara sepeda motor, menuntut adanya sistem pengawasan yang lebih efektif untuk mendukung respons darurat dan kebijakan keselamatan transportasi.

B. Computer Vision dalam Deteksi Kecelakaan

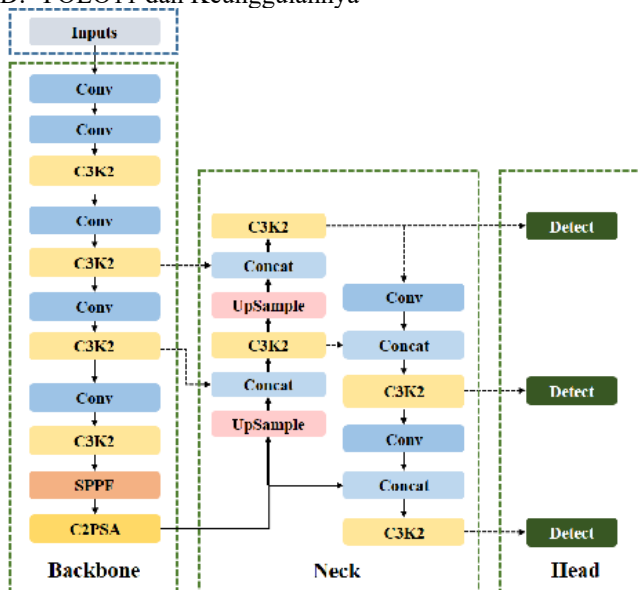
Kemajuan teknologi *computer vision* dan *deep learning* memungkinkan pengembangan sistem deteksi kecelakaan otomatis berbasis kamera pengawas (CCTV). Integrasi model deteksi objek dengan data video lalu lintas terbukti mampu mendeteksi insiden secara *real-time* dan akurat [5], [7]. Beberapa studi, seperti yang dilakukan oleh Adewopo dan Elsayed [10], menunjukkan bahwa pendekatan *ensemble deep learning* dapat mendeteksi kecelakaan dari rekaman CCTV dan dashcam pada skenario kota pintar. Penelitian lain oleh Farhan *et al.* [11] memanfaatkan drone dengan model YOLO11 untuk mengidentifikasi insiden lalu lintas.

Selain deteksi kecelakaan, klasifikasi jenis kendaraan juga penting. Sepeda motor, misalnya, sangat rentan akibat ukuran kecil dan minim perlindungan, sehingga menyumbang proporsi kematian tertinggi pada kecelakaan di Indonesia [12]. Riyadina dan Subik [13] mencatat cedera parah pada pengendara motor umumnya berupa luka terbuka dan cedera kepala. Jenis kendaraan juga memengaruhi tingkat keparahan, misalnya tabrakan mobil-truk lebih fatal dibanding antar mobil [14], [15]. Penelitian lain menekankan bahwa identifikasi jenis kendaraan menjadi faktor kunci dalam analisis kecelakaan untuk mendukung strategi pencegahan dan penanganan yang lebih efektif [16], [17].

C. YOLO sebagai Model Deteksi Objek

YOLO (*You Only Look Once*) merupakan model deteksi objek yang populer karena kemampuannya dalam menghasilkan deteksi *real-time* dengan akurasi tinggi [5], [6]. Model ini membagi citra ke dalam *grid* dan melakukan prediksi *bounding box* serta label klasifikasi pada satu kali proses inferensi. Versi YOLOv8, misalnya, telah digunakan untuk mendeteksi kecelakaan insiden dari kamera pengawas dengan hasil yang cukup baik [7]. Namun, sebagian besar penelitian masih terbatas pada identifikasi insiden tanpa memperhatikan klasifikasi jenis kendaraan yang terlibat.

D. YOLO11 dan Keunggulannya



GAMBAR 1.
(ARSITEKTUR YOLO11 [18])

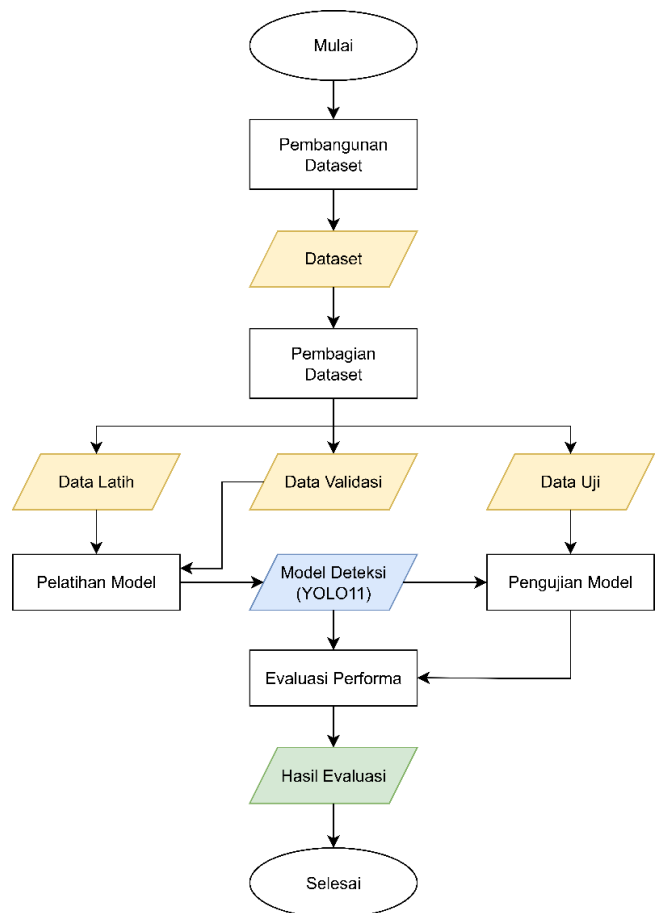
YOLO11 hadir sebagai versi terbaru dengan sejumlah peningkatan dibanding YOLOv8 dan YOLOv10, meliputi efisiensi parameter, akurasi deteksi lebih tinggi, serta

generalisasi yang lebih baik terhadap kondisi visual beragam [9], [11], [18], [19]. Arsitektur YOLO11 terdiri dari tiga komponen utama: *backbone*, *neck*, dan *head*, dengan modul seperti C3k2, SPPF, dan C2PSA yang mendukung ekstraksi fitur lebih efisien [9]. Dengan keunggulan tersebut, YOLO11 dianggap tepat untuk penelitian ini dalam membangun model deteksi kecelakaan yang tidak hanya mengenali insiden tetapi juga mengklasifikasikan jenis kendaraan yang terlibat.

III. METODE

Penelitian ini dirancang untuk membangun dan mengevaluasi model deteksi kecelakaan lalu lintas berbasis *You Only Look Once version 11* (YOLO11). Proses penelitian meliputi pengumpulan data, persiapan dataset, pelatihan model, validasi, pengujian, serta analisis hasil.

A. Rancangan Penelitian



GAMBAR 2.
(ALUR SISTEM YANG DIBANGUN)

Gambar 2 menunjukkan alur penelitian, dimulai dari pengumpulan *dataset* citra kecelakaan lalu lintas menggunakan kamera pengawas (*Closed Circuit Television – CCTV*) yang kemudian dianotasi dengan *bounding box* sesuai format YOLO11. Data yang telah disiapkan dibagi menjadi *subset* pelatihan, validasi, dan pengujian. Selanjutnya dilakukan proses pelatihan menggunakan *pretrained model* YOLO11, validasi kinerja model pada setiap *epoch*, dan pengujian akhir untuk mengukur kemampuan generalisasi.

B. Waktu dan Tempat Penelitian

Seluruh eksperimen dilakukan pada *platform* Kaggle Notebook menggunakan GPU NVIDIA Tesla P100 dengan kapasitas 16 GB. Bahasa pemrograman yang digunakan adalah Python 3.11 dengan *framework* PyTorch 2.6.0. Penelitian ini dilaksanakan pada semester genap tahun akademik 2024/2025.

C. Dataset

Dataset yang digunakan merupakan gabungan dari Roboflow, Kaggle, Google Images, dan tangkapan layar video YouTube [20], [21], [22]. Total terdapat 1.483 citra, yang terdiri atas tiga kelas kecelakaan yaitu *car_accident*, *motorbike_accident*, dan *car_motorbike_accident*, serta citra latar belakang (*background*). Dataset dibagi ke dalam tiga subset, yaitu 10% untuk pengujian, 90% sisanya untuk pelatihan dan validasi dengan perbandingan 80:20. Pembagian 80:20 antara data pelatihan dan validasi merupakan praktik umum dalam pelatihan model *machine learning*, meskipun tidak ada ketentuan baku yang mengatur proporsi tersebut [23], [24]

D. Metode Penelitian

Metode yang digunakan adalah pendekatan *deep learning* berbasis YOLO11. Data pelatihan melalui proses augmentasi berupa perubahan warna, *horizontal flipping*, *random erasing*, dan *mosaic augmentation* untuk meningkatkan variasi visual. Model dilatih dengan *optimizer* AdamW, *learning rate* awal 0,001, serta total 150 *epoch*. Untuk menghindari *overfitting*, lapisan awal model dibekukan (*freeze layer*) sebanyak 10.

Pelatihan model menggunakan 150 *epoch*. Penggunaan 150 *epoch* dipilih sebagai kompromi antara akurasi model dan keterbatasan sumber daya. Jumlah *epoch* dalam pelatihan *deep learning* tidak memiliki standar baku, melainkan bergantung pada karakteristik data, arsitektur model, dan kapasitas komputasi [25]. Dengan keterbatasan *platform* Kaggle versi gratis yang hanya menyediakan 30 jam per minggu, 150 *epoch* dinilai realistis serta masih sejalan dengan praktik umum penggunaan 100–200 *epoch* pada berbagai studi [25], [26].

Evaluasi dilakukan dengan menggunakan empat metrik utama, yaitu *Precision* (P), *Recall* (R), *Mean Average Precision at Intersection over Union* (IoU) *threshold* 0.5 atau *mAP@0.5*, dan *Mean Average Precision at multiple IoU thresholds* (0.5–0.95) atau *mAP@0.5:0.95*. Tahap ini penting untuk mengukur sejauh mana sistem deteksi kecelakaan berbasis YOLO11 mampu mengenali insiden secara akurat, konsisten, dan dapat digeneralisasikan pada data baru. Definisi metrik disajikan sebagai berikut:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$mAP@50 = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3)$$

$$mAP@50:95 = \frac{1}{10} \sum_{t=0,5}^{0,95} mAP@t \quad (4)$$

dimana:

TP = True Positive

FP = False Positif

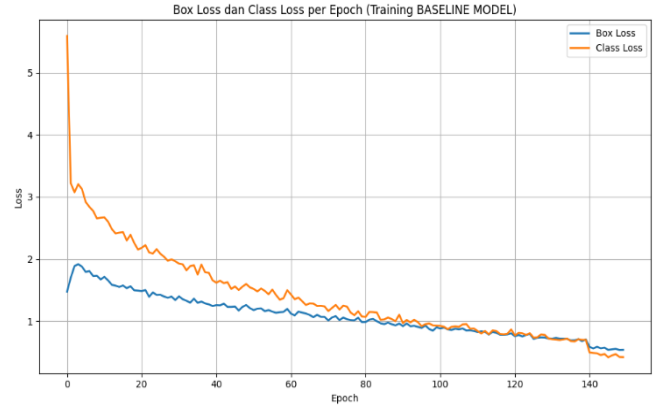
FN = False Negatif

AP = Average Precision

Metrik *precision* mengukur akurasi prediksi positif, *recall* menilai seberapa banyak objek terdeteksi, sedangkan *mAP* memberikan gambaran rata-rata akurasi klasifikasi seluruh kelas kecelakaan.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pelatihan Model



GAMBAR 3.

GRAFIK LOSS PADA PELATIHAN MODEL BASELINE

Proses pelatihan YOLO11 dijalankan sebanyak 150 *epoch*. Selama pelatihan, nilai *training loss* menurun secara konsisten hingga mencapai titik stabil pada sekitar *epoch* ke-120. Hal ini menunjukkan bahwa model telah mencapai kondisi konvergen.

B. Evaluasi Model

Kinerja model dievaluasi menggunakan data uji. Hasil evaluasi ditunjukkan pada Tabel 1, yang memuat metrik *precision*, *recall*, dan *mAP*.

TABEL 1.
METRIK GLOBAL EVALUASI BASELINE DAN FINE-TUNING

Metrik Evaluasi	Model Baseline			Model Fine-Tuning		
	Train	Valid	Test	Train	Valid	Test
Precision (P)	1,00	0,70	0,82	1,00	0,64	0,81
Recall (R)	0,99	0,69	0,79	0,99	0,71	0,72
mAP@0.5	0,99	0,74	0,84	0,98	0,72	0,75
mAP@0.5:0.95	0,90	0,48	0,54	0,89	0,47	0,51

Berdasarkan metrik global pelatihan pada Tabel 1, baik *baseline* maupun *fine-tuning* mencapai *precision* dan *recall* hampir sempurna ($\geq 0,99$) dengan *mAP@0.5* di atas 0,98. Namun, saat diuji pada data validasi dan pengujian, performa menurun signifikan, terutama pada *mAP@0.5:0.95* yang hanya mencapai 0,47–0,54. Hal ini menunjukkan model mengalami gap generalisasi terhadap data baru.

Confusion Matrix (PELATIHAN BASELINE)					
Actual	car_accident	247	4	0	13
	car_motorbike_accident	0	89	1	12
	motorbike_accident	0	1	189	19
	background	0	0	1	0
		car_accident	car_motorbike_accident	motorbike_accident	background
		Predicted			

GAMBAR 4.

CONFUSION MATRIX MODEL BASELINE PADA DATA LATIH

Gambar 4 menampilkan *confusion matrix baseline* pada data latih. Secara umum, model mampu mengenali kelas

car_accident dan *motorbike_accident* dengan tingkat akurasi yang cukup baik, masing-masing menghasilkan 247 dan 189 prediksi benar. Namun, performa model masih terbatas pada kelas *car_motorbike_accident*, di mana sebagian *instance* salah diklasifikasikan sebagai *background*, yaitu *car_accident* atau *motorbike_accident*.

Confusion Matrix (PELATIHAN FINE-TUNING)				
Actual	car_accident	car_motorbike_accident	motorbike_accident	background
	247	0	0	31
	0	93	0	16
	0	1	190	20
	0	0	1	0
Predicted				

GAMBAR 5.

CONFUSION MATRIX MODEL FINE-TUNING PADA DATA LATIH

Pada Gambar 5, terlihat bahwa model *fine-tuning* mengklasifikasikan *car_accident* dengan benar pada 247 *instance* dan 31 salah ke *background*. Kelas *car_motorbike_accident* terdeteksi benar pada 93 *instance* dengan 16 salah ke *background*, sedangkan *motorbike_accident* terdeteksi benar pada 190 benar, 1 salah ke *car_motorbike_accident*, dan 20 salah ke *background*. Selain itu, terdapat 1 *instance background* yang salah diprediksi sebagai *motorbike_accident*.

Confusion Matrix (VALIDASI BASELINE)				
Actual	car_accident	car_motorbike_accident	motorbike_accident	background
	42	1	0	13
	3	19	4	3
	3	0	37	16
	19	8	12	0
Predicted				

GAMBAR 6.

CONFUSION MATRIX MODEL BASELINE PADA DATA VALIDASI

Confusion matrix pada Gambar 6 menunjukkan model *baseline* paling akurat dalam mendeteksi *motorbike_accident* (37 benar) dan *car_accident* (42 benar). Namun, masih terjadi kesalahan signifikan pada kelas *background*, terutama ketika diprediksi sebagai *car_accident* (19 kasus) maupun sebaliknya. Selain itu, terdapat kebingungan pada kelas *car_motorbike_accident*, yang kerap salah terdeteksi sebagai *motorbike_accident* atau *car_accident*.

Confusion Matrix (VALIDASI FINE-TUNING)				
Actual	car_accident	car_motorbike_accident	motorbike_accident	background
	44	3	1	23
	3	21	3	12
	4	0	38	24
	16	4	11	0
Predicted				

GAMBAR 7.

CONFUSION MATRIX MODEL FINE-TUNING PADA DATA VALIDASI

Dari Gambar 7, *confusion matrix* menunjukkan model berhasil mengklasifikasikan *car_accident* (44 benar) dan *motorbike_accident* (38 benar) dengan akurasi relatif tinggi, sementara *car_motorbike_accident* hanya 21 benar dan masih sering salah menjadi kelas lain maupun *background*. Selain itu, masih terjadi prediksi keliru ke *background*, yaitu 16 pada *car_accident*, 4 pada *car_motorbike_accident*, dan 11 pada *motorbike_accident*, yang menggambarkan distribusi kesalahan model pada data validasi pasca *fine-tuning*.

Confusion Matrix (PENGUJIAN BASELINE)				
Actual	car_accident	car_motorbike_accident	motorbike_accident	background
	24	2	0	5
	0	13	0	3
	0	0	21	8
	9	1	8	0
Predicted				

GAMBAR 8.

CONFUSION MATRIX MODEL BASELINE PADA DATA UJI

Gambar 8 menunjukkan model mendeteksi *motorbike_accident* dengan benar pada 21 *instance* dan *car_accident* pada 24 *instance*, meskipun lebih rendah dibandingkan validasi. Kelas *car_motorbike_accident* hanya terdeteksi benar pada 13 *instance*, dengan 3 *instance* salah menjadi *background*. Selain itu, masih terdapat prediksi keliru terhadap *background*, yaitu 9 pada *car_accident*, 1 pada *car_motorbike_accident*, dan 8 pada *motorbike_accident*, yang mengindikasikan distribusi deteksi belum sepenuhnya stabil pada data uji.

Confusion Matrix (PENGUJIAN FINE-TUNING)				
Actual	car_accident	car_motorbike_accident	motorbike_accident	background
	26	1	0	9
	1	14	2	8
	0	0	19	16
	6	1	8	0
Predicted				

GAMBAR 9.

CONFUSION MATRIX MODEL FINE-TUNING PADA DATA UJI

Confusion matrix pada Gambar 9 menunjukkan model mengklasifikasikan *car_accident* (26 benar) dan *motorbike_accident* (19 benar) dengan cukup baik, sedangkan *car_motorbike_accident* hanya 14 benar dengan kesalahan ke *car_accident* (1), *motorbike_accident* (2), dan *background* (8). Masih terdapat prediksi keliru ke *background*, yaitu 6 pada *car_accident*, 1 pada *car_motorbike_accident*, dan 8 pada *motorbike_accident*.

TABEL 2.
METRIK PER KELAS MODEL BASELINE DAN FINE-TUNING PADA DATA LATIH

Kelas	Model Baseline				Model Fine-Tuning			
	P	R	Map @0.5	Map @0.5:0.95	P	R	Map @0.5	Map @0.5:0.95
0	0,99	0,99	0,99	0,91	1,00	1,00	0,99	0,89
1	1,00	0,99	0,99	0,90	0,99	0,98	0,99	0,90
2	1,00	0,98	0,99	0,88	1,00	0,98	0,99	0,88

Keterangan:

Kelas 0 = *car_accident*Kelas 1 = *car_motorbike_accident*Kelas 2 = *motorbike_accident*

P = Precision

R = Recall

Pada data latih (Tabel 2), semua kelas memperoleh *precision* mendekati sempurna ($\geq 0,99$) dengan nilai *mAP@0.5* yang konsisten pada angka 0,99. Nilai *mAP@0.5:0.95* sedikit lebih rendah (0,88–0,91), tetapi tetap menunjukkan ketelitian deteksi yang tinggi. Hal ini mengindikasikan bahwa model mampu belajar dengan baik dari data pelatihan.

TABEL 3.
METRIK PER KELAS MODEL *BASELINE* DAN *FINE-TUNING* PADA DATA VALIDASI

Kelas	Model Baseline				Model Fine-Tuning			
	P	R	mAP@0.5	mAP@0.5:0.95	P	R	mAP@0.5	mAP@0.5:0.95
0	0,76	0,64	0,74	0,52	0,69	0,66	0,68	0,45
1	0,67	0,71	0,73	0,52	0,60	0,75	0,77	0,55
2	0,68	0,72	0,75	0,39	0,62	0,73	0,72	0,42

Pada tahap validasi (Tabel 3), performa menurun cukup signifikan. Kelas *car_accident* pada *baseline* mencatat *precision* tertinggi (0,76), tetapi *recall* lebih rendah (0,64). Sebaliknya, kelas *car_motorbike_accident* pada *fine-tuning* memiliki *recall* tertinggi (0,75) namun *precision* lebih rendah (0,60). Nilai *mAP@0.5* per kelas berada pada kisaran 0,68–0,77, sedangkan *mAP@0.5:0.95* masih rendah (0,39–0,55), yang menandakan inkonsistensi deteksi pada *threshold* IoU yang lebih ketat.

TABEL 4.
METRIK PER KELAS MODEL *BASELINE* DAN *FINE-TUNING* PADA DATA UJI

Kelas	Model Baseline				Model Fine-Tuning			
	P	R	mAP@0.5	mAP@0.5:0.95	P	R	mAP@0.5	mAP@0.5:0.95
0	0,80	0,72	0,80	0,55	0,89	0,71	0,80	0,55
1	0,94	0,93	0,94	0,74	0,88	0,88	0,87	0,69
2	0,73	0,73	0,76	0,33	0,67	0,59	0,59	0,28

Pada data uji (Tabel 4), kelas *car_motorbike_accident* tampil sebagai yang paling stabil, khususnya pada model *baseline* (*precision* 0,94, *recall* 0,93, *mAP@0.5* 0,94). Kelas *car_accident* menunjukkan peningkatan *precision* pada *fine-tuning* (0,89), namun *recall* tetap moderat (0,71). Sementara itu, kelas *motorbike_accident* menjadi yang paling lemah, terutama pada *fine-tuning*, dengan *recall* hanya 0,59 dan *mAP@0.5:0.95* sebesar 0,28.

Model *baseline* menunjukkan performa sangat baik pada data pelatihan dengan *precision* sempurna (1,00), *recall* tinggi (0,99), serta *mAP@0.5* dan *mAP@0.5:0.95* masing-masing 0,99 dan 0,90. Kinerja per kelas juga konsisten dengan nilai *precision* dan *recall* mendekati sempurna. Namun, pada data validasi performa menurun cukup signifikan (*precision* 0,70; *recall* 0,69; *mAP@0.5* 0,74; *mAP@0.5:0.95* 0,48), mengindikasikan *overfitting*. Pada data pengujian, performa relatif stabil dengan *precision* 0,82, *recall* 0,79, dan *mAP@0.5* sebesar 0,84. Kelas *car_motorbike_accident* mencatat performa terbaik (*precision* 0,94; *recall* 0,93), sementara kelas *motorbike_accident* masih memiliki keterbatasan terutama pada akurasi spasial.

Model *fine-tuning* meningkatkan kestabilan performa antara data validasi dan pengujian. Pada data pelatihan, hasil *precision* 1,00 dan *recall* 0,99 menunjukkan kemampuan deteksi hampir sempurna, dengan *mAP@0.5* sebesar 0,98 dan *mAP@0.5:0.95* sebesar 0,89. Pada data validasi, *precision* turun menjadi 0,64 dengan *recall* 0,71, sementara pada pengujian *precision* dan *recall* masing-masing 0,81 dan 0,72. Performa antar kelas relatif merata, terutama untuk *car_motorbike_accident* (*precision* 0,88; *recall* 0,88; *mAP@0.5* 0,87). Sebaliknya, *motorbike_accident*

menunjukkan kelemahan dengan *precision* dan *recall* lebih rendah serta akurasi spasial terbatas.

Meski performa *model* cukup tinggi, sejumlah kesalahan prediksi masih terjadi pada data pengujian, seperti ditunjukkan pada Gambar 10–12.



GAMBAR 10.
KESALAHAN DETEKSI PADA LALU LINTAS PADAT/MACET

Model mendeteksi kondisi kemacetan sebagai kecelakaan mobil. Hal ini dipengaruhi oleh kemiripan visual antara mobil yang berdekatan pada kondisi macet dengan citra kecelakaan. Situasi ini menunjukkan adanya tantangan bagi model dalam membedakan pola spasial antara kemacetan dan tabrakan nyata.



GAMBAR 11.
KESALAHAN DETEKSI PADA OBJEK ORANG

Model salah mengklasifikasikan orang tergeletak sebagai kecelakaan motor. Kesalahan ini terjadi karena objek manusia memiliki pola visual mirip dengan sepeda motor pada sudut tertentu, sementara dataset pelatihan tidak mencakup anotasi untuk objek manusia. Kondisi ini menegaskan perlunya penambahan kelas negatif yang merepresentasikan keberadaan manusia agar model mampu membedakan dengan lebih akurat.

melalui integrasi ke sistem CCTV untuk menilai robustnes dan kontribusinya terhadap keselamatan transportasi.



GAMBAR 12.
KECELAKAAN MOBIL TIDAK TERDETEKSI OLEH MODEL

Kecelakaan yang terjadi di luar titik tabrak yang jelas pada kamera terabaikan oleh model. Kesalahan seperti ini terjadi karena tampilannya menyerupai mobil yang sedang parkir atau berdekatan di jalan.

Secara keseluruhan, *baseline* lebih unggul dibanding *fine-tuning* terutama pada data pengujian, yang menjadi indikator performa dalam kondisi nyata. *Fine-tuning* memang memperlihatkan kestabilan antar subset data, namun *baseline* mencatat nilai absolut metrik yang lebih tinggi, khususnya pada kelas *car_motorbike_accident*. Dengan demikian, *baseline* dipandang lebih andal sebagai model deteksi kecelakaan berbasis citra. Perbedaan distribusi data antar kelas, khususnya jumlah kasus kecelakaan mobil yang lebih dominan, menjadi salah satu faktor yang memengaruhi hasil ini.

V. KESIMPULAN

Penelitian ini berhasil membangun dan mengevaluasi model deteksi kecelakaan lalu lintas berbasis YOLO11 untuk mengidentifikasi kelas *car_accident*, *motorbike_accident*, dan *car_motorbike_accident*. Berdasarkan pengujian pada data pelatihan, validasi, dan pengujian, model *baseline* menunjukkan performa akurasi tertinggi dengan *Precision* 0,82, *Recall* 0,79, *mAP@0.5* sebesar 0,84, dan *mAP@0.5:0.95* sebesar 0,54. Model *fine-tuning* memberikan hasil yang lebih stabil dan seimbang antar kelas, namun nilai metrik absolutnya sedikit lebih rendah dibanding *baseline*. *Fine-tuning* efektif dalam mengurangi kesalahan prediksi terhadap latar belakang, sementara keterbatasan utama terlihat pada deteksi kelas *motorbike_accident*, yang akurasi spasialnya rendah akibat kemiripan visual dengan kondisi lalu lintas normal atau objek lain.

Penelitian selanjutnya dapat difokuskan pada peningkatan kualitas dan variasi dataset, khususnya kelas minoritas dan kelas dengan kemiripan fitur, serta penambahan kendaraan seperti truk dan sepeda. Eksplorasi YOLO11 yang lebih besar (YOLO11m/YOLO11l) dapat meningkatkan akurasi meski membutuhkan komputasi lebih besar. Penanganan latar belakang perlu dirancang lebih eksplisit dengan kelas negatif terstruktur, dan model sebaiknya diuji pada skenario nyata

REFERENSI

- [1] World Health Organization, "Global status report on road safety 2023," Geneva, 2023.
- [2] D. A. Muhammad and A. Maulana, "Sepanjang 2023 Ada 148.307 Kasus Kecelakaan di Seluruh Indonesia, Jawa Timur Mendominasi," *Kompas.com*, Jakarta, Jan. 16, 2024. Accessed: Aug. 11, 2025. [Online]. Available: <https://otomotif.kompas.com/read/2024/01/16/181200215/sepanjang-2023-ada-148.307-kasus-kecelakaan-di-seluruh-indonesia-jawa-timur>
- [3] Z. Xi, X. Liu, Y. Liu, Y. Cai, and Y. Zheng, "Integrating Generative Adversarial Networks and Convolutional Neural Networks for Enhanced Traffic Accidents Detection and Analysis," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2025, doi: 10.1109/TITS.2025.3573172.
- [4] M. C. Cruz and N. N. Ferenchak, "Emergency Response Times for Fatal Motor Vehicle Crashes, 1975–2017," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 8, pp. 504–510, Aug. 2020, doi: 10.1177/0361198120927698.
- [5] M. S. Pillai, G. Chaudhary, M. Khari, and R. G. Crespo, "Real-time image enhancement for an automatic automobile accident detection through CCTV using deep learning," *Soft comput*, vol. 25, no. 18, pp. 11929–11940, Sep. 2021, doi: 10.1007/s00500-021-05576-w.
- [6] M. I. Basheer Ahmed et al., "A Real-Time Computer Vision Based Approach to Detection and Classification of Traffic Incidents," *Big Data and Cognitive Computing*, vol. 7, no. 1, p. 22, Jan. 2023, doi: 10.3390/bdcc7010022.
- [7] D. T. Mane, S. Sangve, S. Kandhare, S. Mohole, S. Sonar, and S. Tupare, "Real-Time Vehicle Accident Recognition from Traffic Video Surveillance using YOLOV8 and OpenCV," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 5s, pp. 250–258, May 2023, doi: 10.17762/ijritcc.v11i5s.6651.
- [8] M. D. Fontaine, C. S. Bhamidipati, and L. E. Dougald, "Safety Impact of Truck Lane Restrictions on Multilane Freeways," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2096, no. 1, pp. 25–32, Jan. 2009, doi: 10.3141/2096-04.
- [9] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," *ArXiv*, Oct. 2024.
- [10] V. A. Adewopo and N. Elsayed, "Smart City Transportation: Deep Learning Ensemble Approach for Traffic Accident Detection," *IEEE Access*, vol. 12, pp. 59134–59147, 2024, doi: 10.1109/ACCESS.2024.3387972.

- [11] M. Farhan, H. Eesaar, A. Ahmed, K. T. Chong, and H. Tayara, "Transforming Highway Safety With Autonomous Drones and AI: A Framework for Incident Detection and Emergency Response," *IEEE Open Journal of Vehicular Technology*, vol. 6, pp. 829–845, 2025, doi: 10.1109/OJVT.2025.3549387.
- [12] A. Buntara, "Cedera Akibat Kecelakaan Lalu Lintas di Indonesia: Faktor Risiko serta Strategi Pencegahan dan Intervensi," *Jurnal Ilmiah Kesehatan Masyarakat*, vol. 11, no. 3, 2019.
- [13] W. Riyadina and I. P. Subik, "Profil keparahan cedera pada korban kecelakaan sepeda motor di Instalasi Gawat Darurat RSUP Fatmawati," *Universa Medicina*, vol. 28, no. 2, pp. 64–72, 2007.
- [14] Q. Wu *et al.*, "Analysis of Injury Severity of Drivers Involved Different Types of Two-Vehicle Crashes Using Random-Parameters Logit Models with Heterogeneity in Means and Variances," *J Adv Transp*, vol. 2023, pp. 1–15, Oct. 2023, doi: 10.1155/2023/3399631.
- [15] S. Conrady and L. Jouffe, "Vehicle Size, Weight, and Injury Risk: High-Dimensional Modeling and Causal Inference with Bayesian Networks," Jun. 2013.
- [16] I. Adnan, T. Umer, A. Arsalan, M. M. Al Dabel, A. K. Bashir, and A. Ansif, "Data driven vehicular heterogeneity based intelligent collision avoidance system for Internet of Vehicles (IoV)," *Digital Communications and Networks*, Apr. 2025, doi: 10.1016/j.dcan.2025.03.010.
- [17] C. Carrodano, "Data-driven risk analysis of nonlinear factor interactions in road safety using Bayesian networks," *Sci Rep*, vol. 14, no. 1, p. 18948, Aug. 2024, doi: 10.1038/s41598-024-69740-6.
- [18] Z. Zhou, "Traffic accident detection based on YOLOv11," in *2024 IEEE 2nd International Conference on Electrical, Automation and Computer Engineering (ICEACE)*, IEEE, Dec. 2024, pp. 363–369. doi: 10.1109/ICEACE63551.2024.10898397.
- [19] M. A. R. Alif, "YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems," Oct. 2024.
- [20] C. C. Kumar, "Accident Detection From CCTV Footage," Kaggle. Accessed: Aug. 11, 2025. [Online]. Available: <https://doi.org/10.34740/KAGGLE/DSV/1379553>
- [21] Tammy, "bike_accident Dataset," Roboflow Universe. Accessed: Aug. 11, 2025. [Online]. Available: https://universe.roboflow.com/tammy/bike_accident
- [22] DatasetV5p1, "training-4 Dataset," Roboflow Universe. Accessed: Aug. 11, 2025. [Online]. Available: <https://universe.roboflow.com/datasetv5p1/training-4>
- [23] F. Shahidi, S. Mohd Daud, H. Abas, N. A. Ahmad, and N. Maarop, "Breast Cancer Classification Using Deep Learning Approaches and Histopathology Image: A Comparison Study," *IEEE Access*, vol. 8, pp. 187531–187552, 2020, doi: 10.1109/ACCESS.2020.3029881.
- [24] V. R. Joseph, "Optimal ratio for data splitting," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 15, no. 4, pp. 531–538, Aug. 2022, doi: 10.1002/sam.11583.
- [25] M. Jeri and Z. S. Hidayat, "Penentuan ukuran batch optimal untuk pelatihan YOLOv8 dalam pendeteksian objek pada kendaraan otonom," *Jurnal Ilmiah NERO*, 2024.
- [26] N. Das and S. Das, "Epoch and accuracy based empirical study for cardiac MRI segmentation using deep learning technique," *PeerJ*, vol. 11, p. e14939, Mar. 2023, doi: 10.7717/peerj.14939.