

Penerapan Algoritma YOLOv11 dalam Sistem Deteksi Kecelakaan pada Lalu Lintas dengan Integrasi Notifikasi Telegram

¹Alif Firman Hakim, ²Adinda Febryanty Tambunan, ³Muhammad Dhafa Mahardhika, ⁴Rifqi Abdillah Rosyad

^{1,2,3,4} Program Studi Informatika, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika

Alamat Surat

Email: ¹firmanhakimalif@gmail.com*, ²adindatambunan3@gmail.com,
³mdhafam3004@gmail.com, ⁴rifqiabdillahrosyad@gmail.com

Article History:

Diajukan: 6 Juli 2025; Direvisi: 4 Agustus 2025; Accepted: 4 April 2026

ABSTRAK

Kecelakaan lalu lintas masih menjadi penyebab utama kerugian jiwa dan ekonomi di Indonesia, sementara pemantauan konvensional CCTV belum mampu memberikan peringatan cepat kepada petugas. Penelitian ini bertujuan merancang sistem deteksi kecelakaan otomatis berbasis algoritma YOLOv11 yang terintegrasi dengan notifikasi Telegram untuk mempercepat respons darurat. Metodologi mengikuti model waterfall, mulai dari pengumpulan video kecelakaan publik, ekstraksi 722 frame, pelabelan pada Roboflow, augmentasi hingga 1523 citra, pelatihan model 50 epoch pada resolusi 640×640 , serta pengujian real-time pada tiga skenario video. Hasil pelatihan menunjukkan mAP50 0,90 dan mAP50-95 0,643. Uji validasi menghasilkan akurasi klasifikasi 91,4% dengan waktu inferensi 84,2–157,4 ms per frame dan keberhasilan pengiriman notifikasi 100%. Sistem mampu membedakan situasi normal dan kecelakaan, serta mengirim tangkapan layar, koordinat, dan waktu secara real-time. YOLOv11 efektif mendeteksi kecelakaan pada CCTV dan mengirim notifikasi secara cepat, berpotensi meningkatkan efisiensi penanganan insiden di lapangan. Keterbatasan utama terletak pada ukuran dan keragaman dataset, perluasan data dan eksplorasi transfer learning direkomendasikan agar performa tetap stabil pada lingkungan nyata yang lebih heterogen.

Kata kunci: Yolo, YOLOv11, Pembelajaran Mesin, Deteksi Kecelakaan

ABSTRACT

Traffic accidents remain a leading source of human and economic loss in Indonesia, while conventional CCTV monitoring still fails to deliver timely warnings to first responders. This study designs an automated accident-detection system built on the YOLOv11 algorithm and integrated with Telegram notifications to accelerate emergency response. The methodology adopts a waterfall model encompassing the acquisition of public accident videos, extraction of 722 frames, annotation in Roboflow, augmentation to 1,523 images, training for 50 epochs at a resolution of 640×640 , and real-time evaluation across three video scenarios. Training produced a mAP50 of 0.90 and a mAP50-95 of 0.643. Validation testing achieved a classification accuracy of 91.4%, an inference time of 84.2–157.4 ms per frame, and a 100% success rate for notification delivery. The system reliably differentiates normal traffic from crash events and transmits screenshots, coordinates, and timestamps in real time. YOLOv11 thus proves effective for accident detection on CCTV and rapid alert dissemination, with the potential to enhance on-site incident management efficiency. The principal limitation concerns dataset size and diversity, expanding the data and exploring transfer-learning approaches are advised to sustain performance in more heterogeneous real-world settings.

Keywords: Yolo, YOLOv11, Pembelajaran Mesin, Deteksi Kecelakaan

1. PENDAHULUAN

Kecelakaan di jalan raya merupakan masalah yang krusial yang kerap muncul di berbagai tempat di dunia, termasuk juga di Indonesia. Dampak dari kecelakaan ini menyebabkan berbagai kerugian bagi para korban, baik yang bersifat materiil maupun non-materiil (Kaawoan, 2023). Metode tradisional untuk menangani kecelakaan, yaitu dengan menelepon nomor darurat, dapat menghadapi berbagai kendala seperti keterlambatan dan sulitnya diterapkan jika lokasi kejadian terjadi di area yang terpencil (Rezky et al., 2023). Namun demikian, apabila sarana dan prasarana lalu lintas yang tersedia tidak memadai serta kurang dikelola secara efektif, maka hal ini berpotensi meningkatkan frekuensi kecelakaan lalu lintas secara substansial.

Menurut data statistik kejadian kecelakaan lalu lintas di Indonesia Pada tahun 2024, terdapat sekitar 149.263 insiden laka lantasi (Bareskrim Polri, 2024), mengalami kenaikan dari tahun sebelumnya yang mencapai 150.331 insiden. Informasi dari Korps Lalu Lintas Polri juga memperlihatkan kenaikan dalam jumlah korban dengan luka ringan, yakni dari 182.886 insiden pada tahun 2023 menjadi 187.162 insiden pada tahun 2024. Sementara itu, jumlah korban dengan cedera serius juga menurun sebanyak 14%, yang berkurang dari 12.475 pada tahun 2019 menjadi 10.751 pada tahun 2020. Total kematian akibat kecelakaan lalu lintas pada tahun 2020 tercatat sebanyak 23.529, berkurang sekitar 8% dibandingkan tahun sebelumnya yang mencapai 25.671 korban (Dyah Pangestu Suhartati et al., 2022).

Untuk meningkatkan keamanan di jalan raya, pengembangan sistem yang mampu mendeteksi kecelakaan adalah hal yang sangat krusial. Sistem ini memungkinkan perekaman dan analisis mendalam atas faktor-faktor penyebab kecelakaan, sehingga tindakan yang tepat dan segera dapat dilaksanakan untuk menghindari terulangnya insiden serupa di waktu yang akan datang.

Seiring berkembangnya teknologi komputer dan kecerdasan buatan, penerapan kecerdasan buatan terus meningkat, terutama dalam sektor computer vision yang memiliki fungsi serupa dengan pemanfaatan komputer untuk memahami kecerdasan manusia (Hayati et al., 2023), berbagai metode juga telah dikembangkan untuk mendeteksi kecelakaan secara otomatis. Salah satu pendekatan yang menjanjikan adalah penggunaan teknologi object detection yang mampu mengenali dan melacak peristiwa visual dalam rekaman video atau citra secara real-time. Di antara berbagai algoritma deteksi objek yang ada, Metode You Only Look Once (YOLO) adalah teknik terkini dalam deteksi objek yang menawarkan akurasi luar biasa dan mampu mendukung implementasi program deteksi kecelakaan ini.

Penelitian yang dilakukan Triana dan Enri pada tahun 2022 mengadakan studi untuk mendeteksi insiden di jalan dengan menggunakan metode CNN dan berhasil meraih akurasi validasi hingga 90% (Triana & Enri, 2022). Namun, studi tersebut belum melaksanakan pengujian dan belum mempertimbangkan durasi pengambilan kesimpulan. Selanjutnya, pada tahun 2022 penelitian dilakukan oleh Dio dkk. dimaksudkan untuk menciptakan sistem identifikasi kecelakaan pada mobil dengan memanfaatkan algoritma YOLOv5, Penelitian ini menggabungkan teknologi computer vision dengan data video dari CCTV untuk mendeteksi kejadian kecelakaan secara otomatis. Hasilnya menunjukkan bahwa YOLOv5 memiliki tingkat akurasi yang tinggi sebesar 84% dalam mendeteksi kecelakaan, sehingga menjadi salah satu acuan penting dalam bidang ini (Dio Riza Pratama et al., 2022). Kemudian, Alaydrus & Banu Santoso pada tahun 2024 membangun sistem pendeteksian kecelakaan yang otomatis dengan memanfaatkan pembelajaran mesin terawasi melalui penerapan algoritma YOLOv8, guna meningkatkan efisiensi serta akurasi dalam mendeteksi kejadian kecelakaan secara real-time dan mendukung sistem respons darurat yang lebih cepat dan tepat sasaran. Sehingga memungkinkan respons cepat dari pihak terkait dalam menangani situasi darurat di lapangan (Farhan & Setiaji, 2023). Terakhir, Ghulam dkk. pada tahun 2022 juga menerapkan algoritma YOLOv7 dengan memanfaatkan dataset "Accident Detection 1" dari Roboflow. Pelatihan ini dilakukan dengan batch size 1, epoch 40, dan optimizer SGD, hasil presisi 65,1% (Ghulam et al., 2023).

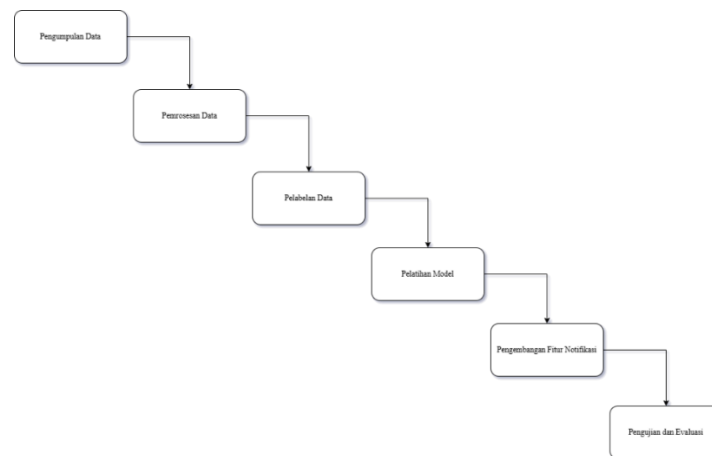
Mengacu pada diskusi di atas, studi yang telah dilaksanakan melibatkan pengembangan suatu model klasifikasi untuk mengidentifikasi kecelakaan di jalan menggunakan algoritma YOLOv11.

You Only Look Once (YOLO) adalah sebuah metode yang dirancang untuk mengenali objek secara langsung dalam waktu nyata (Hidayatulloh, 2021). YOLOv11, sebagai versi terbaru dari algoritma ini, menawarkan peningkatan terutama dalam hal kecepatan dan akurasi deteksi, seperti kendaraan dalam kondisi kecelakaan (Ismail & Rahmadewi, 2025). Dengan memanfaatkan YOLOv11, sistem deteksi kecelakaan dapat diimplementasikan secara lebih efisien dan andal, baik dalam lingkungan pengawasan lalu lintas maupun dalam sistem kendaraan pintar.

Dokumen ini memiliki tujuan untuk merancang dan menerapkan sistem deteksi kecelakaan otomatis yang menggunakan YOLOv11, serta mengevaluasi kinerjanya dalam mendeteksi insiden kecelakaan dari data visual secara real-time. Diharapkan penelitian ini bisa memberikan sumbangan untuk kemajuan teknologi pengamanan transportasi yang lebih adaptif dan responsif, serta mampu menyesuaikan diri dengan berbagai keadaan dan tuntutan di lapangan, yang pada gilirannya dapat meningkatkan keselamatan para pengguna secara menyeluruh.

2. METODE

Langkah pertama yang diambil oleh penyusun untuk merancang sistem adalah dengan memilih jenis model informasi yang akan digunakan. Dalam konteks ini, penyusun memilih atau menentukan model sistem informasi, di mana salah satu pilihannya adalah menerapkan model sistem waterfall (Fachri & Surbakti, 2021). Alur dari penelitian ini mencakup sejumlah langkah yang terstruktur dalam rangka merancang dan menerapkan sistem yang mendeteksi kecelakaan lalu lintas dengan menggunakan YOLOv11. Rangkaian prosedur yang dilaksanakan terdiri dari beberapa tahap yang diilustrasikan dalam diagram alir pada Gambar 1.



Sumber : Hasil penelitian (2025)

Gambar 1. Diagram Alir Penelitian

2.1 Pengumpulan Data

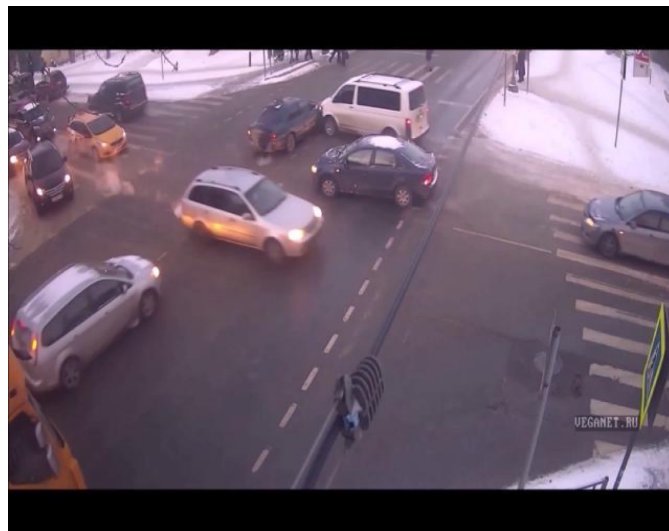
Pengumpulan informasi dilakukan melalui pemilihan contoh video seperti yang tertera pada Tabel 1 yang berperan sebagai referensi untuk pelatihan dan penilaian model YOLO v11. Berikut adalah rangkaian tahapan yang dilakukan dalam proses pemilihan contoh video yang ditunjukkan pada Gambar 2, Gambar 3, dan Gambar 4 :

- 1) Menemukan sumber data yang mungkin berguna.
- 2) Mengambil atau merekam video dari beragam tempat dan situasi.
- 3) Menyimpan video dengan format yang sesuai untuk tahap pengolahan berikutnya.

Tabel 1. Dataset yang digunakan

No	Video	Sumber	Ukuran	Format
1	Video A	Youtube	18,8 MB	MP4
2	Video B	Youtube	5,86 MB	MP4
3	Video C	Youtube	14,4 MB	MP4

Sumber : Hasil penelitian (2025)



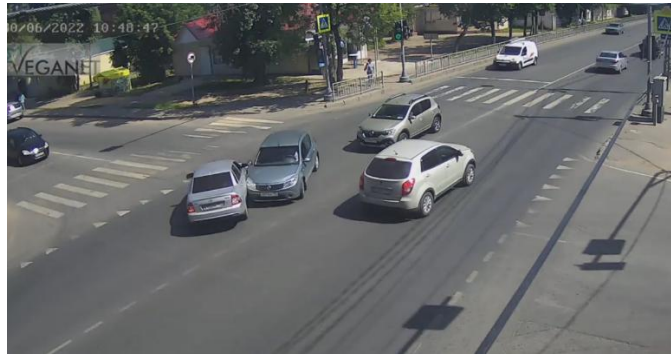
Sumber : https://www.youtube.com/watch?v=_yJyGv91ITs

Gambar 2. Video kecelakaan A di CCTV



Sumber : <https://www.youtube.com/watch?v=oK7PZ3ybl18>

Gambar 3. Video kecelakaan B di CCTV



Sumber : <https://www.youtube.com/watch?v=om6go9I5SO0>

Gambar 4. Video kecelakaan C di CCTV

2.2 Pemrosesan Data

Dalam tahap ini, penulis melakukan pecahan video menjadi gambar-gambar atau frame yang dapat diberi tanda Bertujuan untuk mempersiapkan data gambar tunggal dari video dalam rangka proses penandaan sesuai yang dicantumkan pada Tabel 2 dengan tahap-tahap yang meliputi :

- 1) Memanfaatkan pustaka OpenCV dan diolah menggunakan Python untuk mengubah video menjadi gambar-gambar.
- 2) File-file gambar tersebut disimpan di folder yang sudah ditentukan.

Tabel 2. Hasil Konversi Video menjadi Gambar

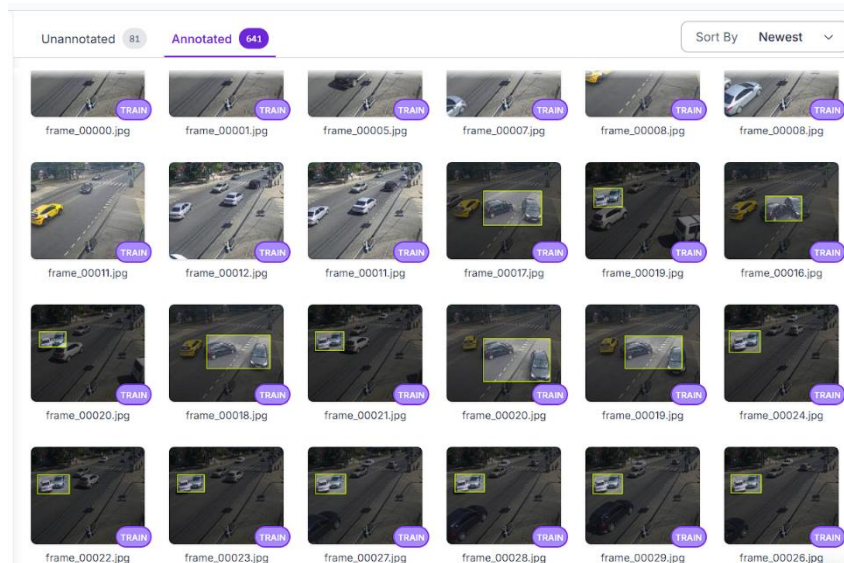
No	Video	Jumlah Gambar	Frame Per Second (FPS)	Frame Size
1	Video A	395	25	1280x720
2	Video B	136	25	1280x720
3	Video C	191	25	1280x720
TOTAL		722		

Sumber : Hasil penelitian (2025)

2.3 Pelabelan Gambar

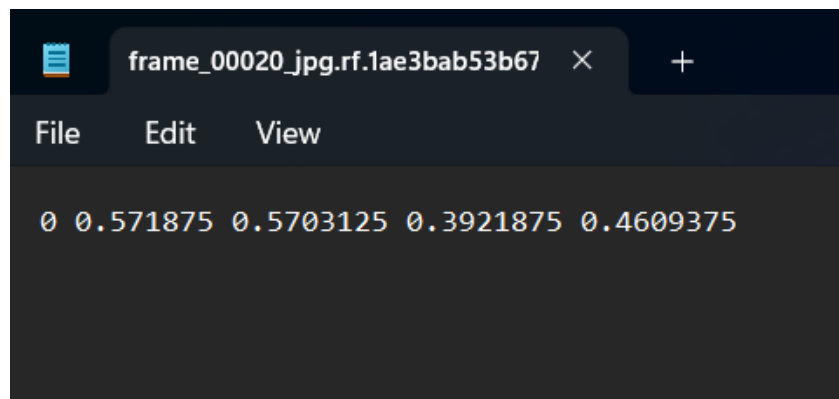
Selanjutnya, pelabelan data merupakan aktifitas pelabelan pada setiap gambar dengan membuat bounding box dan memberi nama kelas (Subekti & Putra, 2024). Dalam tahapan ini, Peneliti memberikan label pada tiap gambar menggunakan website Roboflow dengan maksud untuk menghasilkan dataset berkualitas tinggi agar penandaan objek sesuai untuk mengembangkan model YOLOv11 sehingga menghasilkan penandaan seperti yang terlihat pada Gambar 5 dan Gambar 6. Dengan tahapan sebagai berikut :

- 1) Frame dibuka dalam bentuk gambar menggunakan website Roboflow.
- 2) Memberikan pelabelan “kecelakaan” pada objek kendaraan yang mengalami kecelakaan.
- 3) Label disimpan dalam bentuk yang sesuai dengan YOLO v11.



Sumber : Hasil penelitian (2025)

Gambar 5. Hasil Pelabelan Menggunakan Roboflow



Sumber : Hasil penelitian (2025)

Gambar 6. Hasil Pelabelan menggunakan format YOLOv11

2.4 Pelatihan Model

Setelah itu, dilakukan pengembangan model YOLO dengan memanfaatkan kumpulan data yang sudah disiapkan (Ramadhani et al., 2024). Pada titik ini, Penulis melaksanakan pelatihan untuk model dengan melalui Sekumpulan langkah yang dirancang untuk melatih model YOLO v11 supaya mampu mengidentifikasi objek seperti mobil dan situasi kecelakaan, sebagai berikut:

- 1) Dataset dipersiapkan.
- 2) Menyiapkan lingkungan pelatihan dengan Python.
- 3) Pelatihan dijalankan.

2.5 Pengembangan Fitur Notifikasi

Pengembangan fitur pemberitahuan penulis dilakukan melalui pengiriman pesan serta tangkapan layar ke Telegram. Jika teridentifikasi adanya insiden kecelakaan, maka notifikasi akan muncul secara langsung kepada pihak yang berkepentingan. Dengan langkah-langkah sebagai berikut:

- 1) Memanfaatkan API pada Telegram untuk mengirim notifikasi teks dan gambar.
- 2) Mengatur bot Telegram agar dapat menerima dan menampilkan sebuah notifikasi.
- 3) Python mengintegrasikan sebuah notifikasi telegram dengan bentuk script bahwa terdeteksi kecelakaan.

2.6 Pengujian

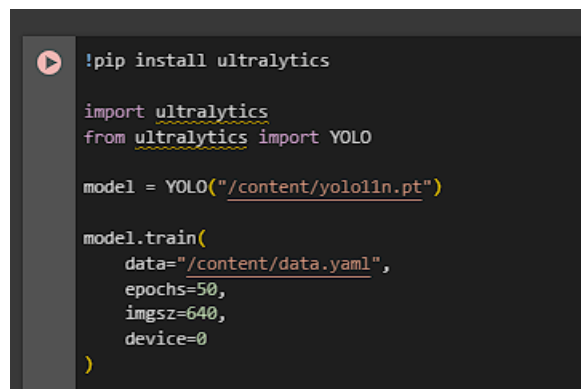
Pengujian sistem dilakukan dengan cara mendeteksi kecelakaan yang ditangkap oleh layar CCTV kemudian hasilnya dikirimkan ke program Python guna olah data kemudian hasilnya dalam bentuk notifikasi di Telegram Bot. Adapun tahapannya sebagai berikut:

- 1) Mula-mula peneliti mempersiapkan skenario pengujian dengan video berbagai situasi kecelakaan untuk diuji oleh Python.
- 2) Python mengevaluasi akurasi data visual lalu menghasilkan model pretrain.
- 3) Model pretrain kemudian digunakan pada program Python guna melakukan proses pendeteksian pada CCTV.
- 4) Setelah program mendeteksi adanya unsur kecelakaan maka notifikasi disampaikan ke Bot Telegram secara realtime.

3. HASIL DAN PEMBAHASAN

3.1 Melatih Model dengan YOLOv11

Peneliti menggunakan model YOLOv11 untuk menguji sebanyak 50 epoch (putaran pelatihan) berupa gambar berukuran 640x640 piksel agar menghasilkan plot hasil pelatihan, penulisan program Python seperti pada Gambar 7.



```
!pip install ultralytics

import ultralytics
from ultralytics import YOLO

model = YOLO("/content/yolov11n.pt")

model.train(
    data="/content/data.yaml",
    epochs=50,
    imgsz=640,
    device=0
)
```

Sumber : Hasil penelitian (2025)

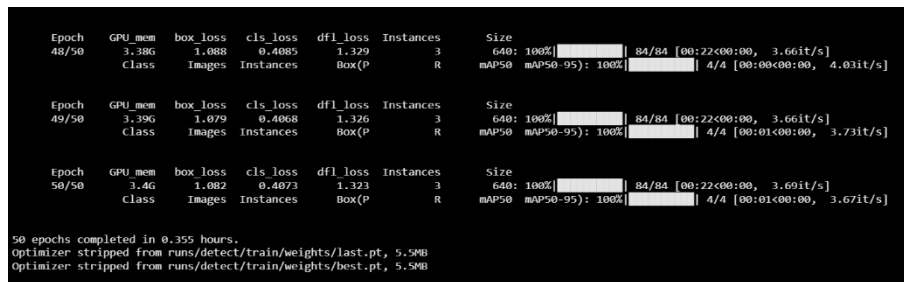
Gambar 7. Perintah untuk melakukan Pelatihan Model

Pada Gambar 7, kode python yang dipergunakan menunjukkan proses melatih model YOLO (You Only Look Once), python juga merupakan scripting language yang berorientasi objek (Dodi Aleksander Manalu, 2022). Sebuah arsitektur yang umum digunakan dalam deteksi objek secara real-time. Perintah pertama yaitu `!pip install ultralytics` adalah perintah untuk menginstal pustaka `ultralytics`, yang menyediakan implementasi dari model YOLO yang telah dioptimalkan.

Selanjutnya, pustaka tersebut diimpor melalui perintah `import ultralytics` dan `from ultralytics import YOLO`. Ini memungkinkan penggunaan fungsi dan kelas yang telah tersedia dalam pustaka `ultralytics`. Setelah itu, model YOLO dengan bobot awal yang telah dilatih sebelumnya (`yolov11n.pt`) dimuat melalui perintah `model = YOLO("/content/yolov11n.pt")`.

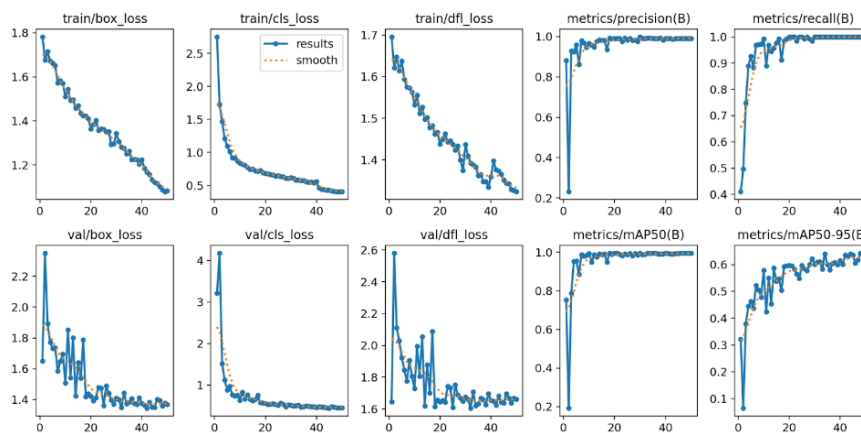
- 1) Perintah selanjutnya adalah perintah utama untuk melatih model: `model.train(data="/content/data.yaml", epochs=50, imgsz=640, device=0)`. Dalam perintah ini:
- 2) `data="/content/data.yaml"` menunjukkan lokasi file konfigurasi dataset, yang berisi informasi seperti jalur gambar pelatihan dan validasi, serta kelas-kelas objek yang akan dideteksi.
- 3) `epochs=50` menunjukkan bahwa pelatihan akan dilakukan sebanyak 50 epoch.
- 4) `imgsz=640` menunjukkan ukuran gambar input yang digunakan dalam proses pelatihan. Ukuran 640x640 merupakan resolusi umum dalam pelatihan model YOLO untuk menyeimbangkan antara kecepatan dan akurasi.
- 5) `device=0` menunjukkan bahwa proses pelatihan dilakukan pada perangkat GPU pertama yang tersedia, yang mempercepat proses pelatihan secara signifikan dibandingkan penggunaan CPU.

Selama proses pelatihan, performa model dievaluasi secara berkala menggunakan data validasi untuk memonitor perkembangan pelatihan dan mencegah overfitting. Hasil dari proses pelatihan ini, seperti akurasi, loss, dan performa lainnya serta hasil bobot (weight) yaitu `best.pt` yang akan digunakan untuk custom model pengujian ditampilkan secara visual maupun numerik, seperti yang terlihat pada Gambar 8.



Sumber : Hasil penelitian (2025)

Gambar 8. Hasil pelatihan menggunakan YOLOv11



Sumber : Hasil penelitian (2025)

Gambar 9. Grafik hasil pelatihan model

Gambar 8 adalah gambar visual hasil dari pelatihan YOLOv11 terhadap 50 epochs yang berjalan sekitar 0,355 hour. Bisa dilihat pada Gambar 9, mAP50 Hasil dari pelatihan ini menunjukkan performa yang sangat impresif dengan nilai rata-rata mAP mencapai 0.9 untuk setiap kategori.

Selanjutnya, peneliti melakukan analisis pada informasi yang diperoleh dari hasil pengujian data yang telah dilatih.

```
Validating runs/detect/train/weights/best.pt...
Ultralytics 8.3.127 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
YOLO11n summary (fused): 100 layers, 2,582,347 parameters, 0 gradients, 6.3 GFLOPs
Class      Images Instances  Box(P  R      mAP50  mAP50-95);
all        128         127    0.992  1      0.994   0.643
Speed: 0.4ms preprocess, 2.6ms inference, 0.0ms loss, 5.5ms postprocess per image
```

Sumber : Hasil penelitian (2025)

Gambar 10. Nilai mAP dari pengujian model

Dari hasil pengujian diperoleh Nilai mAP yang dapat diperhatikan pada Gambar 10. Nilai mAP untuk semua kategori dengan IoU 0.5 adalah 0.994 sementara untuk IoU 0.5 hingga 0.95, nilai mAP tercatat 0.643 yang menunjukkan bahwa model ini memiliki kemampuan dalam mendeteksi objek dengan tingkat ketepatan yang cukup tinggi yaitu hampir menyentuh angka 1.

Langkah terakhir dari proses ini adalah pemberitahuan notifikasi kepada Bot Telegram berupa screenshot dan pesan pada saat objek terdeteksi sedang terjadi kecelakaan secara realtime.

Sebelum memanfaatkan notifikasi Telegram, langkah awal yang harus diambil adalah mendirikan bot di BotFather dengan nama Traffic_Accident_BOT dan memperoleh token bot serta group chat ID melalui Get ID Bot. Token bot berfungsi untuk mengautentikasi bot dalam proses pengiriman pesan, sedangkan group chat ID akan dipakai untuk menentukan siapa yang menerima pesan tersebut.

```
1 import cv2
2 from ultralytics import YOLO
3 import os
4 import random
5 import requests
6 from datetime import datetime
7 from collections import deque
8
9 # ===== SETTING TELE BOT =====
10 TELEGRAM_BOT_TOKEN = '7129737708:AAH1cyf3hX40utpKTEpWiczzy7b0LXtBLUM'
11 TELEGRAM_CHAT_ID = '6328841191'
12
13 # ===== FUNGSI =====
14 def send_telegram_photo(photo_path, caption=None):
15     url = f'https://api.telegram.org/bot{TELEGRAM_BOT_TOKEN}/sendPhoto'
16     with open(photo_path, 'rb') as photo:
17         files = {"photo": photo}
18         data = {"chat_id": TELEGRAM_CHAT_ID}
19         if caption:
20             data["caption"] = caption
21         requests.post(url, files=files, data=data)
22
23 def send_telegram_message(message):
24     url = f'https://api.telegram.org/bot{TELEGRAM_BOT_TOKEN}/sendMessage'
25     payload = {"chat_id": TELEGRAM_CHAT_ID, "text": message}
26     requests.post(url, data=payload)
27
28 def generate_random_location():
29     lat = round(random.uniform(-11.0, 6.1), 6)
30     lon = round(random.uniform(95.0, 141.0), 6)
31     return lat, lon
32
33 # ===== Membuat Folder Screenshot & Klip Video =====
34 if not os.path.exists('screenshots'):
35     os.makedirs('screenshots')
36 if not os.path.exists('clips'):
37     os.makedirs('clips')
38
39 # ===== Load Model YOLO =====
40 model = YOLO('50epoch.pt')
```

Sumber : Hasil penelitian (2025)

Gambar 11. Script untuk mengirim notifikasi teks, log text dan screenshot

Kode python yang terlihat pada Gambar 11 dapat memberikan kesempatan bagi sistem untuk mengirimkan pemberitahuan dalam bentuk notif telegram, tempat kejadian perkara, serta tangkapan layar ke chat Telegram yang telah ditentukan, kemudian juga membuat folder screenshot dan klip video ketika kecelakaan terdeteksi.

Perlu diketahui, untuk penentuan lokasinya peneliti menggunakan generate random Latitude (Garis Lintang) dan Longitude (Garis Bujur) dari negara Indonesia yang bersumber dari website <https://latitudelongitude.org/id/> . Lokasi ditentukan secara random karena penelitian kali ini masih bersifat prototipe dengan harapan jika kelak digunakan pada dunia nyata maka penentuan lokasi bisa disesuaikan dengan titik koordinat CCTV yang dipasang pada jalan.

Selanjutnya, agar notifikasi dapat dikirim secara langsung saat terdeteksi adanya kecelakaan, perlu disisipkan instruksi yang berupa logika seperti yang terlihat pada Gambar 12.

```

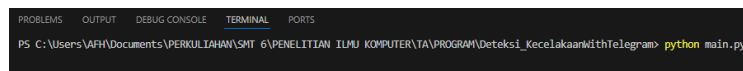
1 # ===== DETEKSI KECELAKAAN =====
2 while True:
3     ret, frame = cap.read()
4     if not ret:
5         break
6
7     frame buffer.append(frame.copy())
8
9     # ===== BENTUK BLOK KECELAKAAN =====
10    results = model.predict(source=frame, source_size=(conf['h'], source_size), device='cpu')
11    found_obj_frame = False
12
13    for result in results:
14        names = [model.names[c] for c in result.names['cls']]
15        confs = result.bboxes.conf
16        xyxy = result.bboxes.xyxy
17
18        for name, conf, box in zip(names, confs, xyxy):
19            if name.lower() not in 'kecelakaan':
20                continue
21            found_obj_frame = True
22            if conf > 0.5:
23                best_conf = float(conf)
24                best_frame = frame.copy()
25                best_box = box
26
27    if detected_vehicle and not found_obj_frame and best_frame is not None:
28        # ===== DETAIL KECELAKAAN =====
29        timestamp = datetime.now()
30        timestamp_str = timestamp.strftime("%Y-%m-%d %H:%M:%S")
31        file_name = timestamp.strftime("%Y-%m-%d_%H-%M-%S")
32        screenshot_path = f'screenshots/{timestamp_str}.jpg'
33
34        # ===== BENTUK BLOK KECELAKAAN =====
35        x1, y1, x2, y2 = best_box.tolist()
36        cv2.imwrite(screenshot_path, frame[y1:y2, x1:x2, :])
37        cv2.imwrite(video_path, best_frame[y1:y2, x1:x2, :])
38
39        # ===== DETAIL KECELAKAAN =====
40        url = 'https://www.google.com/maps/@1,1,15m'
41        map_link = f'https://www.google.com/maps/@1,1,15m'
42        pesan = f'===== KECELAKAAN =====\n'
43        pesan += f'📍 Lokasi: {map_link}\n'
44        pesan += f'📸 Foto & Video: {screenshot_path}\n'
45        pesan += f'📺 Video: {video_path}\n'
46
47        # ===== DETAIL KECELAKAAN =====
48        cv2.imwrite(screenshot_path, best_frame)
49        save_screenshot_path(screenshot_path, pesan)
50
51        # ===== DETAIL KECELAKAAN =====
52        with open('log.txt', 'a') as log:
53            log.write(f'===== KECELAKAAN =====\n')
54            log.write(f'📍 Lokasi: {map_link}\n')
55            log.write(f'📸 Foto & Video: {screenshot_path}\n')
56            log.write(f'📺 Video: {video_path}\n')
57
58        # ===== DETAIL KECELAKAAN =====
59        cv2.imwrite(video_path, best_frame)
60        save_video_path(video_path, pesan)
61
62        # ===== DETAIL KECELAKAAN =====
63        frame_height, frame_width = best_frame.shape[:2]
64        out = cv2.VideoWriter(video_path, cv2.VideoWriter_4CC_P_N, 10, 1, frame_width, frame_height)
65

```

Sumber : Hasil penelitian (2025)

Gambar 12. Script python untuk mengirimkan notifikasi saat kecelakaan terjadi

Setelah perintah selesai disusun, langkah selanjutnya adalah melakukan pengujian dengan mengeksekusi kode sumber utama pada Gambar 13 untuk memastikan bahwa perintah yang diterapkan berfungsi dengan baik dan benar.



Sumber : Hasil penelitian (2025)

Gambar 13. Menjalankan program utama

3.2 Pengujian

Pada penelitian kali ini, data yang digunakan untuk pengujian adalah video yang sudah disiapkan untuk melakukan pengujian, yaitu Video A, Video B, dan Video C. Pengujian tahap pertama dilaksanakan pada Tabel 3.

Tabel 3. Pengujian Notifikasi di Video A

No	Gambar	Keterangan
1		Hasil dari proses pendeteksian dan kecepatan mengirimkan notifikasi berupa gambar dan teks ke telegram.
2		Tampilan Frame video yang ditampilkan saat dijalankan program deteksi kecelakaan pada kondisi dimana sedang terjadinya kecelakaan.
3		Notifikasi berupa screenshot dan teks peringatan saat terjadinya kecelakaan terkirim ke telegram bot yang sebelumnya dibuat.

Sumber : Hasil penelitian (2025)

Dari Tabel 3, terlihat bahwa kecepatan Preprocess (2.3ms), dalam waktu Inference (80.2ms), dan Postprocess (1.6ms), sehingga total kecepatan keseluruhan didapatkan sebesar 84.2ms.

Tabel 4. Pengujian Notifikasi di Video B

No	Gambar	Keterangan
1		Hasil dari proses pendeteksian dan kecepatan mengirimkan notifikasi berupa gambar dan teks ke telegram.
2		Tampilan Frame video yang ditampilkan saat dijalankan program deteksi kecelakaan pada kondisi dimana sedang terjadinya kecelakaan.
3		Notifikasi berupa screenshot dan teks peringatan saat terjadinya kecelakaan terkirim ke telegram bot yang sebelumnya dibuat.

Sumber : Hasil penelitian (2025)

Dari Tabel 4, dapat diamati bahwa waktu yang dicapai adalah Preprocess (13.3ms), Inference (142.3ms), dan Postprocess (1.8ms) sehingga total keseluruhan adalah 157.4ms.

Tabel 5. Pengujian Notifikasi di Video C

No	Gambar	Keterangan
1		Hasil dari proses pendeteksian dan kecepatan mengirimkan notifikasi berupa gambar dan teks ke telegram.
2		Tampilan Frame video yang ditampilkan saat dijalankan program deteksi kecelakaan pada kondisi dimana sedang terjadinya kecelakaan.
3		Notifikasi berupa screenshot dan teks peringatan saat terjadinya kecelakaan terkirim ke telegram bot yang sebelumnya dibuat.

Sumber : Hasil penelitian (2025)

Selanjutnya pada Tabel 5, terlihat bahwa, kecepatan yang didapat yaitu Preprocess (5.1ms), Inference (132.6ms) , dan Postprocess (1.6ms) dan total kecepatan yang diperoleh sebesar 139.3ms.

4. SIMPULAN DAN SARAN

Dari penelitian ini terlihat bahwa sistem berbasis YOLOv11 mampu mengenali dan mengklasifikasikan insiden kecelakaan kendaraan dengan akurasi mencapai 0,914 serta mengirimkan notifikasi real-time melalui Bot Telegram dalam rentang waktu respons masing-masing sebesar 84,2 ms hingga 157,4 ms, sehingga dapat mendukung penanganan kecelakaan lalu lintas secara cepat dan efektif, meski perlu perbaikan lebih lanjut pada keanekaragaman data untuk meningkatkan performa deteksi.

5. DAFTAR PUSTAKA

- Bareskrim Polri. (2024). *Statistik Laka Lantas*. https://pusiknas.polri.go.id/laka_lantas
- Dio Riza Pratama, M., Priyatna, B., Hilabi, S. S., & Hananto, A. L. (2022). Deteksi Objek Kecelakaan Pada Kendaraan Roda Empat Menggunakan Algoritma YOLOv5. *Jurnal Ilmiah Sistem Informas*, 12(2), 15–26.
- Dodi Aleksander Manalu, G. G. (2022). IMPLEMENTASI METODE DATA MINING K-MEANS CLUSTERING TERHADAP DATA PEMBAYARAN TRANSAKSI MENGGUNAKAN BAHASA PEMROGRAMAN PYTHON PADA CV DIGITAL DIMENSI. *INFOTECH: JOURNAL OF TECHNOLOGY INFORMATION*, file:///C:/, 1–10. <https://doi.org/https://doi.org/10.37365/jti.v8i1.131>
- Dyah Pangestu Suhartati, Tri Susila Hidayati, Suprpto Hadi, & Dessi Rosdiana. (2022). Penelitian Efektivitas Penggunaan Media Puzzle Penanda Dalam Meningkatkan Pengetahuan Keselamatan Lalu Lintas Pada Siswa Sekolah Dasar Di Kabupaten Batang. *Jurnal Ilmiah Teknik Mesin, Elektro Dan Komputer*, 2(3), 236–240. <https://doi.org/10.51903/juritek.v2i3.2154>
- Fachri, B., & Surbakti, R. W. (2021). Perancangan Sistem Dan Desain Undangan Digital Menggunakan Metode Waterfall Berbasis Website (Studi Kasus: Asco Jaya). *Journal of Science and Social Research*, 4(3), 263. <https://doi.org/10.54314/jssr.v4i3.692>

- Farhan, N. M., & Setiaji, B. (2023). Indonesian Journal of Computer Science. *Indonesian Journal of Computer Science*, 12(2), 284–301.
<http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135>
- Ghulam, M., Binuri, A., Amirul, M., Haryanti, T., & Peningkatan, A. (2023). Penerapan Algoritma Yolo V7 Sebagai Dekteksi Kecelakaan Pada Lalu Lintas. *Prosiding-Seminar Nasional Teknologi Informasi & Ilmu Komputer (SEMASTER)*, 2(1), 35–41.
<https://journal.unilak.ac.id/index.php/Semaster/article/view/18402/5832>
- Hayati, N. J., Singasatia, D., Muttaqin, M. R., Informatika, T., Tinggi, S., & Wastukencana, T. (2023). OBJECT TRACKING MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE (YOLO)v8 UNTUK MENGHITUNG KENDARAAN. *KOMPUTA : Jurnal Ilmiah Komputer Dan Informatika*, 12(2). <https://universe.roboflow.com/>
- Hidayatulloh, M. S. (2021). Sistem Pengenalan Wajah Menggunakan Metode Yolo (You Only Look Once). *Universitas Medika*, i–43.
- Ismail, D. R., & Rahmadewi, R. (2025). *SISTEM DETEKSI JALAN BERLUBANG SECARA REAL-TIME MENGGUNAKAN YOLOV11 : INTEGRASI DATA DAN LOKASI MELALUI WEBSITE (STUDI KASUS : DAERAH KARAWANG)*. 9(3), 3953–3961.
- Kaawoan, Y. J. W. (2023). Ganti Kerugian Oleh Pihak Yang Bertanggung Jawab Atas Terjadinya Kecelakaan Lalu Lintas. *Jurnal Lex Privatum*, 11(3), 9.
- Ramadhani, F., Satria, A., & Dewi, S. (2024). Identifikasi Kendaraan Bermotor pada Dashcam Mobil Menggunakan Algoritma YOLO. *Hello World Jurnal Ilmu Komputer*, 2(4), 199–206.
<https://doi.org/10.56211/helloworld.v2i4.466>
- Rezky, A., Bagir, A., Pamerean, D., & Makhrus, F. (2023). Deteksi Kecelakaan Lalu Lintas Otomatis Pada Rekaman CCTV Indonesia Menggunakan Deep Learning. *Buletin Pagelaran Mahasiswa Nasional Bidang Teknologi Informasi Dan Komunikasi*, 1(1), 1–5.
- Subekti, J. N., & Putra, A. D. (2024). *DETEKSI TRAFIC PADA JUMLAH KENDARAAN YANG LEWAT DI YOUSEE INDONESIA MENGGUNAKAN METODE YOLOV8*. 7(2), 78–86.
- Triana, H., & Enri, U. (2022). Penerapan Deep Learning Pada Kamera Pengawas Jalan Raya Dalam Mendeteksi Kecelakaan. *Syntax : Jurnal Informatika*, 11(02), 46–55.
<https://doi.org/10.35706/syji.v11i02.6356>