

Pengembangan Sistem Deteksi Kecelakaan di Jalan Raya Menggunakan Algoritma YOLOv8 dan Notifikasi Otomatis Melalui Telegram

Syarif Ahmad Hasny Al Mutsanna Alaydrus¹, Banu Santoso^{2*}

syarifhas123@students.amikom.ac.id¹, banu@amikom.ac.id^{2*}

^{1,2}Fakultas Ilmu Komputer, Teknik Komputer, Universitas Amikom Yogyakarta

Informasi Artikel

Diterima : 29 Jun 2024
Direview : 23 Jul 2024
Disetujui : 8 Agu 2024

Kata Kunci

Deteksi Kecelakaan,
YOLOv8, Deteksi Objek,
Notifikasi Telegram.

Abstrak

Penelitian ini mengembangkan sistem deteksi kecelakaan dengan metode supervised machine learning menggunakan YOLOv8 untuk deteksi objek. Tahapannya mencakup pengumpulan data, pelabelan, pelatihan model, dan implementasi sistem dengan notifikasi ke Telegram. Data diambil dari video sampel, diubah menjadi gambar per frame, dan dilabeli dengan LabelImg. YOLOv8 dilatih untuk mengenali lima kelas objek: mobil, kecelakaan, truk, orang, dan sepeda motor. Implementasinya menggunakan Python dengan OpenCV, ultralytics, dan cvzone. Sistem ini mengirimkan notifikasi real-time ke Telegram saat terjadi kecelakaan, dengan akurasi rata-rata 0,914 dan waktu notifikasi 287,2ms – 334,1ms. Sistem ini membantu pengawasan lalu lintas dan respons cepat terhadap kecelakaan, mengurangi dampak negatif kecelakaan lalu lintas.

Keywords

Accident Detection, YOLOv8,
Object Detection, Telegram
Notification.

Abstract

This research develops an accident detection system using supervised machine learning with YOLOv8 for object detection. The stages include data collection, labeling, model training, and system implementation with Telegram notifications. Data is taken from sample videos, converted into frame-by-frame images, and labeled with LabelImg. YOLOv8 is trained to recognize five object classes: car, accident, truck, person, and motorcycle. Implementation is done in Python with OpenCV, ultralytics, and cvzone. The system sends real-time notifications to Telegram upon an accident, achieving an average accuracy of 0.914 with notification times of 287.2ms – 334.1ms. This system aids traffic monitoring and quick response to accidents, reducing the negative impact of traffic accidents.

A. Pendahuluan

Kecelakaan lalu lintas merupakan salah satu masalah utama yang dihadapi di berbagai negara, termasuk Indonesia. Kecelakaan lalu lintas tidak hanya menyebabkan kerugian materi, tetapi juga mengakibatkan korban luka dan jiwa. Menurut data dari Kepolisian Negara Republik Indonesia (POLRI) jumlah kecelakaan lalu lintas di Indonesia terus meningkat setiap tahunnya [1]. Oleh karena itu, diperlukan sistem yang efektif untuk mendeteksi dan merespons kejadian kecelakaan secara cepat[2].

Teknologi deteksi objek telah berkembang pesat seiring dengan kemajuan dalam bidang kecerdasan buatan (Artificial Intelligence) dan pembelajaran mesin (Machine Learning). Salah satu pendekatan yang populer adalah penggunaan model deteksi objek berbasis Deep Learning, seperti YOLO (You Only Look Once)[3], SSD (Single Shot MultiBox Detector)[4], R-CNN (Region-based Convolutional Neural Network)[5], dari ketiga algoritma tersebut YOLO merupakan algoritma yang paling sering digunakan saat ini dan memiliki kecepatan dan akurasi yang lebih baik[6].

Pengembangan sistem ini melibatkan beberapa tahap utama, yaitu pengumpulan data, pelabelan data, pelatihan model, dan implementasi sistem deteksi serta menambahkan fitur notifikasi. Data yang digunakan berasal dari sampel video yang diolah menjadi gambar per frame dan dilabeli menggunakan alat LabelImg. Model YOLOv8 dilatih menggunakan dataset yang telah dilabeli untuk mengenali lima kelas objek: car, accident, truck, person, dan motorcycle. Setelah model dilatih, sistem deteksi diimplementasikan menggunakan bahasa pemrograman Python dengan bantuan pustaka OpenCV, ultralytics, dan cvzone.

Pada penelitian yang dilakukan oleh Heru Triana dan Ultach Enri (2022), membahas tentang Penerapan Deep Learning Pada Kamera Pengawas Jalan Raya Dalam Mendeteksi Kecelakaan. Algoritma yang digunakan adalah CNN (Convolutional Neural Network) untuk mendeteksi terjadinya kecelakaan dan mengirimkan peringatan atau notifikasi kepada rumah sakit, dari penelitian ini hasil yang didapatkan rata-rata nilai akurasi deteksi di angka 90.80%[7].

Pada penelitian yang dilakukan oleh Adam Rezky , Abdullah Bagir , David Pamerean dan Faizal Makhrus (2023), membahas tentang Deteksi Kecelakaan Lalu Lintas Otomatis Pada Rekaman CCTV Indonesia Menggunakan Deep Learning. Algoritma yang digunakan adalah YOLOv8 untuk mendeteksi adanya kecelakaan dan tidak adanya kecelakaan , dari penelitian ini hasil yang didapatkan rata-rata nilai akurasi deteksi di angka 89%[8].

Pada penelitian yang dilakukan oleh Luthfi Atikah , Diana Purwitasari dan Nanik Suciati (2022), membahas tentang Deteksi Kejadian Lalu Lintas pada Teks Twitter dengan Pendekatan Klasifikasi Multi-Label Berbasis Deep Learning. Algoritma yang digunakan adalah CNN (Convolutional Neural Network) dan LSTM (Long Short-Term Memory) untuk mendeteksi terjadinya kecelakaan berdasarkan informasi yang didapat dari platform digital Twitter, dari penelitian ini hasil yang didapatkan rata-rata nilai akurasi deteksi di angka 75%[9].

Pada penelitian yang dilakukan oleh Muhammad Fachry Ikhsal, Budi Arif Dermawan dan Riza Ibnu Adam (2023), membahas tentang Peningkatan Deteksi Kecelakaan di Jalan Raya Menggunakan Real-ESRGAN pada Citra CCTV Persimpangan Jalan. Penelitian ini menggunakan Real-ESRGAN (Real-Enhanced

Super-Resolution Generative Adversarial Network) untuk mendeteksi kecelakaan dan tidak kecelakaan, dari penelitian ini hasil yang didapatkan rata-rata nilai akurasi deteksi di angka 94%[10].

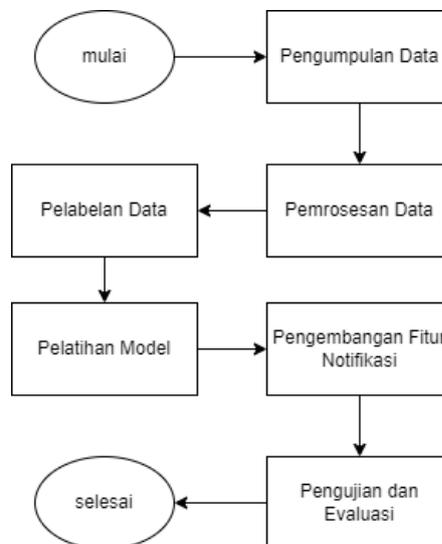
Pada penelitian yang dilakukan oleh Muhammad Dio Riza Pratama, Bayu Priyatna, Shofa Shofiah Hilabi dan April Lia Hananto (2022), membahas tentang Deteksi Objek Kecelakaan Pada Kendaraan Roda Empat Menggunakan Algoritma YOLOv5. Penelitian ini menggunakan algoritma YOLOv5 untuk mendeteksi kecelakaan pada kendaraan beroda 4, dari penelitian ini hasil yang didapatkan rata-rata nilai akurasi deteksi di angka 90.2%[11].

Pada penelitian yang dilakukan oleh Mochammad Ghulam Abrari Binuri, Muhammad Amirul dan Tining Haryanti (2023), membahas tentang Penerapan Algoritma Yolo V7 Sebagai Dekteksi Kecelakaan Pada Lalu Lintas. Penelitian ini menggunakan algoritma YOLOv7 untuk mendeteksi kecelakaan pada kendaraan di lalu lintas, dari penelitian ini hasil yang didapatkan rata-rata nilai akurasi deteksi di angka 26.4%[12].

Penelitian ini bertujuan untuk mengembangkan sistem deteksi kecelakaan yang memanfaatkan metode supervised machine learning dengan teknik klasifikasi menggunakan model YOLOv8. Sistem ini dirancang untuk mendeteksi berbagai objek dalam video, termasuk kendaraan dan kecelakaan, serta memberikan notifikasi secara real-time melalui aplikasi Telegram. Dengan adanya sistem ini, diharapkan dapat membantu dalam pengawasan lalu lintas dan memberikan respons cepat terhadap terjadinya kecelakaan, sehingga dapat mengurangi dampak negatif dari kecelakaan lalu lintas.

B. Metode Penelitian

Alir penelitian ini terdiri dari beberapa tahap yang sistematis untuk mengembangkan dan mengimplementasikan sistem deteksi kecelakaan lalu lintas berbasis YOLOv8. Serangkaian proses yang dilakukan meliputi berbagai tahapan yang ditampilkan dalam diagram alir pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

1. Pengumpulan Data

Pengumpulan data dilakukan dengan mengambil sampel video seperti pada Tabel 1 yang digunakan sebagai sumber data untuk pelatihan dan pengujian model YOLOv8. Berikut adalah langkah-langkah yang diambil dalam proses pengambilan sampel video pada Gambar 2 dan Gambar 3:

- a. Mengidentifikasi sumber data yang potensial.
- b. Mendownload atau merekam video dari berbagai lokasi dan kondisi.
- c. Menyimpan video dalam format yang sesuai untuk pemrosesan lebih lanjut.

Tabel 1. Dataset yang digunakan

| No | Video | Sumber | Ukuran | Format |
|----|---------|---------|---------|--------|
| 1 | Video A | Youtube | 1.40 MB | MP4 |
| 2 | Video B | GitHub | 13.3 mb | MP4 |



Gambar 2. Video Kecelakaan A di CCTV



Gambar 3. Video Kecelakaan B di CCTV

2. Pemrosesan Data

Memecah video menjadi frame-frame gambar yang dapat dilabeli dengan tujuan untuk mempersiapkan data gambar individual dari video untuk proses pelabelan seperti pada Tabel 2. dengan langkah-langkah sebagai berikut :

- a. Menggunakan alat seperti OpenCV untuk memecah video menjadi gambar.
- b. Menyimpan frame-frame gambar dalam direktori yang sudah ditentukan.

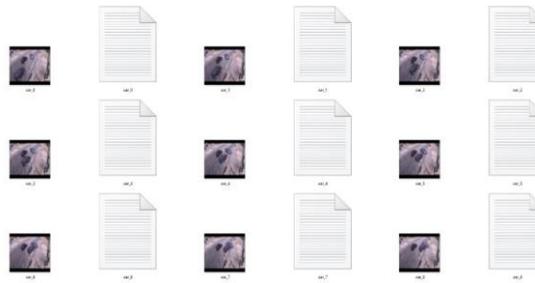
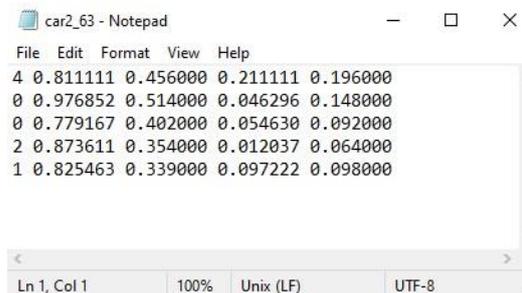
Tabel 2. Hasil Konversi Video

| No | Video | Jumlah Gambar | Frame Per Second (FPS) | Frame Size |
|----|---------|---------------|------------------------|------------|
| 1 | Video A | 129 | 19 | 1080 x 500 |
| 2 | Video B | 129 | 30 | 1080 x 500 |

3. Pelabelan Gambar

Melabeli setiap gambar / image menggunakan alat LabelImg dengan tujuan untuk membuat dataset yang berkualitas dengan pelabelan objek yang sesuai untuk melatih model YOLOv8 dan mendapatkan hasil pelabelan seperti pada Gambar 4 dan Gambar 5. Dengan langkah-langkah sebagai berikut :

- Membuka setiap frame gambar dalam LabelImg.
- Memberikan anotasi kepada objek-objek seperti car, truck, person, motorcycle, dan accident.
- Menyimpan label dalam format yang kompatibel dengan YOLOv8.

**Gambar 4.** Hasil pelabelan menggunakan labeling**Gambar 5.** Hasil pelabelan menggunakan format YOLO

4. Pelatihan Model

Pelatihan model adalah tahap penting dalam pengembangan sistem deteksi kecelakaan. Proses ini melibatkan beberapa langkah yang bertujuan untuk mengajarkan model YOLOv8 agar mampu mendeteksi objek seperti kendaraan dan kecelakaan. Berikut adalah langkah-langkah yang dilakukan dalam proses pelatihan model:

- Mempersiapkan dataset.

- b. Menyiapkan Google Collab dikarenakan pelatihan dilakukan di Google Collab.
- c. Melakukan Pelatihan

5. Pengembangan Fitur Notifikasi

Menambahkan fitur notifikasi yang mengirimkan pesan dan screenshot ke Telegram saat terdeteksi kecelakaan dengan tujuan untuk memberikan notifikasi secara real-time kepada pengguna atau pihak yang terkait. Dengan langkah-langkah sebagai berikut :

- a. Menggunakan API Telegram untuk mengirim pesan dan gambar.
- b. Mengonfigurasi bot Telegram untuk menerima dan menampilkan notifikasi.
- c. Mengintegrasikan notifikasi dalam skrip deteksi kecelakaan.

6. Pengujian

Menguji sistem deteksi kecelakaan dan fitur notifikasi dalam lingkungan terkendali dengan tujuan untuk mengevaluasi kinerja dan keandalan sistem dalam mendeteksi kecelakaan dan mengirim notifikasi. Dengan langkah-langkah sebagai berikut :

- a. Menyiapkan skenario pengujian dengan video yang mencakup berbagai kondisi kecelakaan.
- b. Mengukur akurasi deteksi, kecepatan respon, dan efektivitas notifikasi.
- c. Menganalisis hasil pengujian dan mengidentifikasi area untuk perbaikan.

C. Hasil dan Pembahasan

1. Melatih Model dengan YOLOv8

Melatih dataset dengan YOLOv8 sebanyak 100 epoch (putaran pelatihan) untuk menghasilkan ini digunakan untuk melatih model deteksi objek YOLOv8 dengan dataset khusus selama 100 epoch menggunakan gambar berukuran 800x800 piksel dan menghasilkan plot hasil pelatihan.

```
%cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt
data=/content/datasets/deteksikecelakaan/data.yaml
epochs=100 imgsz=800 plots=True
```

Gambar 6. Perintah untuk melakukan pelatihan

Gambar 6 merupakan perintah yang digunakan untuk melakukan pelatihan sebanyak 100 epoch , epoch sendiri adalah satu siklus lengkap di mana seluruh dataset pelatihan digunakan untuk melatih model. Setiap data training (gambar atau sampel) digunakan satu kali dalam satu epoch untuk menghitung gradient dan memperbarui parameter model.

Dalam proses pelatihan, dataset biasanya dibagi menjadi batch-batch kecil untuk mempercepat proses pembelajaran dan mengelola memori. Jadi, satu epoch berlangsung hingga seluruh batch-batch tersebut telah

digunakan untuk melatih model. Setelah satu epoch selesai, model dievaluasi pada set validasi (jika ada) untuk mengukur performanya.

Tujuan dari pelatihan model melalui beberapa epoch adalah untuk memungkinkan model mempelajari pola yang lebih kompleks dan mendalam dari data pelatihan seiring waktu. Dengan mengulangi epoch, model dapat menyesuaikan diri lebih baik terhadap data dan meningkatkan kinerjanya hasil pada pelatihan di tunjukkan pada Gambar 7. Namun, terlalu banyak epoch juga dapat menyebabkan overfitting, di mana model menjadi terlalu spesifik pada data pelatihan dan tidak umum untuk data data baru. Oleh karena itu, pemilihan jumlah epoch yang tepat adalah bagian penting dari proses pelatihan model.

```

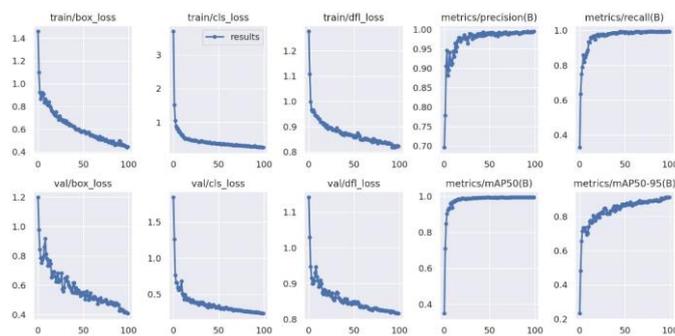
Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
98/100   11.2G    0.4588    0.2616    0.8227    18         800: 100% 17/17 [00:12<00:00, 1.311t/s]
Class   Images  Instances  Box(P)    R          mAP50  mAP50-95)
self.pid = os.fork()
Class   Images  Instances  Box(P)    R          mAP50  mAP50-95)
  all    260    2086    0.992    0.993    0.994    0.912
Motorcycle  260    472    0.993    1        0.994    0.92
Truck      260    181    0.996    1        0.995    0.893
Person    260    439    0.985    0.968    0.993    0.885
Car       260    845    0.990    0.999    0.995    0.963
Accident  260    69     0.992    1        0.995    0.982
/usr/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with
self.pid = os.fork()

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
99/100   11.2G    0.4394    0.2534    0.8242    22         800: 100% 17/17 [00:12<00:00, 1.331t/s]
Class   Images  Instances  Box(P)    R          mAP50  mAP50-95)
self.pid = os.fork()
Class   Images  Instances  Box(P)    R          mAP50  mAP50-95)
  all    260    2086    0.993    0.994    0.994    0.913
Motorcycle  260    472    0.993    1        0.994    0.921
Truck      260    181    0.996    1        0.995    0.896
Person    260    439    0.988    0.971    0.994    0.884
Car       260    845    0.997    0.999    0.995    0.965
Accident  260    69     0.993    1        0.995    0.982
/usr/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with
self.pid = os.fork()

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
100/100  11.2G    0.4442    0.2555    0.8216    37         800: 100% 17/17 [00:12<00:00, 1.311t/s]
Class   Images  Instances  Box(P)    R          mAP50  mAP50-95)
self.pid = os.fork()
Class   Images  Instances  Box(P)    R          mAP50  mAP50-95)
  all    260    2086    0.995    0.994    0.995    0.913
Motorcycle  260    472    0.995    1        0.994    0.921
Truck      260    181    0.997    1        0.995    0.893
Person    260    439    0.993    0.969    0.994    0.885
Car       260    845    0.997    0.999    0.995    0.967
Accident  260    69     0.993    1        0.995    0.981

100 epochs completed in 0.788 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.5MB
    
```

Gambar 7. Hasil pelatihan menggunakan YOLOv8



Gambar 8. Grafik pada saat pelatihan

Gambar 8 ini merupakan grafik hasil dari pelatihan menggunakan YOLOv8 dengan epochs sebanyak 100 yang berlangsung sekitaran 2 jam. Bisa dilihat pada Gambar 8, mAP (Mean Average Precision) di hasil pelatihan tersebut sangat memuaskan dengan rata-rata nilai mAP nya di atas 0.9 pada setiap kelas.

2. Pengujian

Pelatihan yang di lakukan menggunakan YOLOv8 akan menghasilkan sebuah file bobot (weights). Terdapat 2 file bobot yang dihasilkan setelah melakukan proses pelatihan yaitu file “best.pt” dan “last.pt”. Peneliti disini menggunakan bobot “best.pt” untuk pengujian, karena dalam bobot “best.pt” berisi parameter-parameter model yang memiliki penyesuaian yang terbaik sehingga dapat menghasilkan performa yang lebih baik.

Data yang digunakan untuk pengujian ini adalah data gambar yang sudah peneliti siapkan sebelumnya pada saat mengkonversi video menjadi gambar, yaitu gambar dengan 2 kondisi pada masing-masing video yaitu kondisi pada saat tidak terjadi kecelakaan dan saat terjadinya kecelakaan, pengujian pertama dilakukan pada Tabel 3.

Tabel 3. Pengujian Deteksi di Video A

| No | Gambar | Keterangan | Prediksi | Aktual |
|----|--|----------------------------------|---|---|
| 1 |  | Tidak sedang terjadi kecelakaan. | Car (1), Accident (0), Truck (1), Person (0), Motorcycle (4). | Car (1), Accident (0), Truck (1), Person (0), Motorcycle (4). |
| 2 |  | Sedang terjadi kecelakaan. | Car (0), Accident (1), Truck (1), Person (1), Motorcycle (3). | Car (0), Accident (1), Truck (1), Person (0), Motorcycle (3). |

Berdasarkan Tabel 3 gambar yang diambil dari Video A tersebut terdapat objek yang terdeteksi pada saat objek tersebut tidak ada ,seperti pada saat pengujian dalam kondisi sedang terjadinya kecelakaan, yang dimana seharusnya class Motorcycle hanya ada 3, model mendeteksi terdapat 4 motorcycle.

Tabel 4. Pengujian Deteksi di Video B

| No | Gambar | Keterangan | Prediksi | Aktual |
|----|---|----------------------------------|---|---|
| 1 |  | Tidak sedang terjadi kecelakaan. | Car (9), Accident (0), Truck (0), Person (2), Motorcycle (0). | Car (9), Accident (0), Truck (0), Person (2), Motorcycle (0). |
| 2 |  | Sedang terjadi kecelakaan. | Car (6), Accident (1), Truck (0), Person (2), Motorcycle (0). | Car (6), Accident (1), Truck (0), Person (2), Motorcycle (0). |

Selanjutnya pengujian kedua dilakukan dengan Video B, bisa dilihat pada Tabel 4 dimana pada pengujian kedua ini model tidak mengalami kesalahan dalam mendeteksi objek.

Setelah melakukan perbandingan secara manual antara objek prediksi dan objek actual selanjutnya peneliti melakukan analisis terhadap data dari hasil pengujian data yang sudah dilatih.

| Class | Images | Instances | Box(P | R | mAP50 | mAP50-95): |
|------------|--------|-----------|-------|-------|-------|------------|
| all | 260 | 2006 | 0.994 | 0.994 | 0.995 | 0.914 |
| Motorcycle | 260 | 472 | 0.993 | 1 | 0.994 | 0.921 |
| Truck | 260 | 181 | 0.996 | 1 | 0.995 | 0.896 |
| Person | 260 | 439 | 0.991 | 0.971 | 0.994 | 0.804 |
| Car | 260 | 845 | 0.997 | 0.999 | 0.995 | 0.965 |
| Accident | 260 | 69 | 0.993 | 1 | 0.995 | 0.982 |

Gambar 9. Nilai mAP dari pengujian model

Nilai ini bertujuan untuk mengukur kualitas dan keandalan dari sistem deteksi objek. mAP memberikan gambaran tentang seberapa baik model dapat mengenali dan memosisikan objek dengan tepat dalam gambar atau video. Dapat dilihat pada Gambar 9 nilai mAP seluruh kelas dengan IoU 0.5 bernilai sebesar 0.995 sedangkan dengan IoU 0.5 – 0.95 nilai mAP sebesar 0.914 yang dapat diartikan model dapat mendeteksi objek dengan akurasi yang terbilang sangat tinggi.

Untuk Langkah terakhir yaitu menambahkan pemberitahuan atau notifikasi berupa pesan dan screenshot melalui telegram pada saat objek kecelakaan (accident) terjadi.

Sebelum menggunakan notifikasi Telegram, langkah pertama yang perlu dilakukan adalah membuat bot di BotFather dengan nama Deteksikecelakaanbot dan mendapatkan token bot serta group chat ID melalui Get ID Bot. Token bot akan digunakan untuk mengautentikasi bot dalam mengirim pesan, sedangkan group chat ID digunakan untuk menentukan penerima pesan.

```
import requests
import tempfile

def send_telegram_message(message, photo_path=None):
    bot_token = '6642986074:AAEgV36oq8s_hNvGKIK-hEtaOWDDbqEGK_c'
    chat_id = '-4247482458'
    url = f'https://api.telegram.org/bot/{bot_token}/sendMessage'
    params = {'chat_id': chat_id, 'text': message}
    response = requests.post(url, params=params)
    if response.status_code == 200:
        print('Pesan berhasil dikirim ke Telegram!')
        if photo_path:
            send_telegram_photo(photo_path)
    else:
        print(f'Gagal mengirim pesan: {response.json()}')

def send_telegram_photo(photo_path):
    bot_token = '6642986074:AAEgV36oq8s_hNvGKIK-hEtaOWDDbqEGK_c'
    chat_id = '-4247482458'
    url = f'https://api.telegram.org/bot/{bot_token}/sendPhoto'
    files = {'photo': open(photo_path, 'rb')}
    params = {'chat_id': chat_id}
    response = requests.post(url, params=params, files=files)
    if response.status_code == 200:
        print('Screenshot berhasil dikirim ke Telegram!')
    else:
        print(f'Gagal mengirim Screenshot: {response.json()}')
```

Gambar 10. Perintah Fungsi untuk mengirim notifikasi text dan screenshot

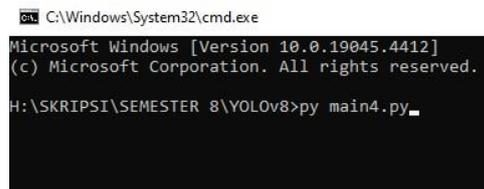
Perintah pada Gambar 10 akan memungkinkan sistem untuk mengirimkan notifikasi berbentuk pesan teks dan screenshot ke grup chat Telegram yang telah ditentukan. Setelah itu agar notifikasi ini dapat khusus terkirim pada saat terdeteksi kecelakaan (accident) terjadi perlu di tambahkan perintah berupa logika seperti pada Gambar 11.

```

for index, row in px.iterrows():
    x1 = int(row[0])
    y1 = int(row[1])
    x2 = int(row[2])
    y2 = int(row[3])
    d = int(row[5])
    c = class_list[d]
    if 'Accident' in c:
        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)
        cvzone.putTextRect(frame, f'{c}', (x1, y1), 1, 1)
        # mengambil screenshot pada saat terjadi kecelakaan (accident)
        _, temp_path = tempfile.mkstemp(suffix='.png')
        cv2.imwrite(temp_path, frame)
        # mengirim notifikasi berupa teks dan screenshot ke Telegram
        send_telegram_message(f'Kecelakaan terdeteksi!', photo_path=temp_path)
    else:
        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
        cvzone.putTextRect(frame, f'{c}', (x1, y1), 1, 1)
    
```

Gambar 11. Perintah logika untuk mengirim notifikasi saat terjadi kecelakaan

Setelah perintah sudah lengkap, Langkah berikutnya yaitu menguji dengan menjalankan source code utama pada Gambar 12 untuk mengetahui apakah perintah yang digunakan sudah berjalan dengan baik dan benar.



Gambar 12. Menjalankan program utama

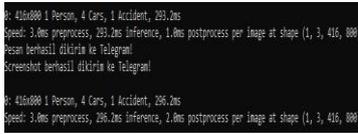
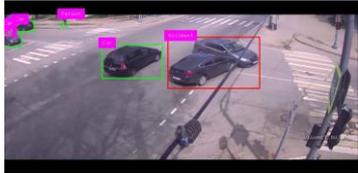
Tabel 5. Pengujian Notifikasi di Video A

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

Dari Tabel 5 bisa dilihat bahwa pengujian pada Video A berhasil dengan kecepatan Preprocess (5.0ms) Waktu yang dibutuhkan untuk tahap pra-pemrosesan gambar., Inference (327.1ms) Waktu yang dibutuhkan model untuk melakukan inferensi atau prediksi terhadap gambar., Postprocess (2.0ms) Waktu yang dibutuhkan untuk tahap pasca-pemrosesan setelah inferensi. Untuk mencari total kecepatan pengiriman

notifikasi nya dengan menjumlahkan Preprocess, Inference , dan Postprocess yaitu didapatkan kecepatan sebesar 334.1ms.

Tabel 6. Pengujian Notifikasi di Video B

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

Pada Tabel 6 ini pengujian dilakukan di Video B ,kali ini kecepatan yang didapatkan yaitu Preprocess (3.0ms), Inference (283.2ms) , dan Postprocess (1.0ms) dari penjumlahan nya didapatkan total kecepatan sebesar 287.2ms.

D. Simpulan

Dalam penelitian ini, telah dikembangkan sistem deteksi kecelakaan kendaraan menggunakan teknologi supervised machine learning dengan teknik klasifikasi menggunakan model YOLOv8. Berdasarkan hasil penelitian, dapat disimpulkan bahwa model YOLOv8 mampu mendeteksi kecelakaan dan mengklasifikasikan objek berdasarkan kelasnya dengan akurasi yang cukup tinggi. Model ini dapat mendeteksi kendaraan dan kecelakaan dengan tingkat presisi rata-rata 0.914, meskipun terdapat keterbatasan pada variasi dataset yang digunakan. Sistem deteksi kecelakaan yang dikembangkan berhasil diintegrasikan dengan bot Telegram untuk mengirim notifikasi secara real-time dengan rentang kecepatan pengiriman notifikasi 287.2ms hingga 334.1ms. Namun, keterbatasan jumlah dan variasi dataset yang digunakan dalam penelitian ini mempengaruhi performa dan generalisasi model. Dataset yang lebih banyak dan bervariasi akan membantu meningkatkan akurasi dan keandalan sistem deteksi kecelakaan.

E. Ucapan Terima Kasih

Terima kasih penulis ucapkan kepada Universitas Amikom Yogyakarta yang telah memberikan kesempatan kepada penulis untuk melakukan penelitian ini.

F. Referensi

- [1] A. Tenri Famauri, D. Utami Mas Bakar, M. Wenda Juniar, F. T. Hoesa, and A. Jalu Prasetyo, "LAW ENFORCEMENT CHALLENGES IN TRAFFIC FOR

- STUDENTS: BETWEEN REALITY AND IDEALISM," *Awang Long Law Rev.*, vol. 3, no. 1, pp. 44–54, 2020, doi: 10.30742/perspektif.v22i3.
- [2] P. Aliya Nabila, S. Soim, and A. Silvia Handayani, "Klasifikasi Kondisi Kendaraan Berpotensi Kecelakaan Berbasis Android Menggunakan Long Short Term Memory," *J. Media Inform. Budidarma*, vol. 8, no. 1, pp. 30–40, 2024, doi: 10.30865/mib.v8i1.7005.
- [3] C. Gao, Q. Cai, and S. Ming, "YoloV4 object detection algorithm with efficient channel attention mechanism," in *Proceedings - 2020 5th International Conference on Mechanical, Control and Computer Engineering, ICMCCE 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 1764–1770. doi: 10.1109/ICMCCE51767.2020.00387.
- [4] M. Zhao, Y. Zhong, D. Sun, and Y. Chen, "Accurate and efficient vehicle detection framework based on SSD algorithm," *IET Image Process.*, vol. 15, no. 13, pp. 3094–3104, Nov. 2021, doi: 10.1049/ipr2.12297.
- [5] N. Arora, Y. Kumar, R. Karkra, and M. Kumar, "Automatic vehicle detection system in different environment conditions using fast R-CNN," *Multimed. Tools Appl.*, vol. 81, no. 13, pp. 18715–18735, 2022, doi: 10.1007/s11042-022-12347-8.
- [6] D. F. Dos Santos, A. O. Françani, M. R. O. A. Maximo, and A. S. C. Ferreira, "Performance Comparison of Convolutional Neural Network Models for Object Detection in Tethered Balloon Imagery," in *2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)*, 2021, pp. 246–251. doi: 10.1109/LARS/SBR/WRE54079.2021.9605459.
- [7] H. Triana and U. Enri, "Penerapan Deep Learning Pada Kamera Pengawas Jalan Raya Dalam Mendeteksi Kecelakaan," *Syntax J. Inform.*, vol. 11, no. 02, pp. 46–55, 2022, doi: 10.35706/syji.v11i02.6356.
- [8] A. Rezky, A. Bagir, D. Pamerean, and F. Makhrus, "Deteksi Kecelakaan Lalu Lintas Otomatis Pada Rekaman CCTV Indonesia Menggunakan Deep Learning," *Bul. Pagelaran Mhs. Nas. Bid. Teknol. Inf. dan Komun.*, vol. 1, no. 1, pp. 1–5, 2023.
- [9] L. Atikah, D. Purwitasari, and N. Suciati, "Deteksi Kejadian Lalu Lintas pada Teks Twitter dengan Pendekatan Klasifikasi Multi-Label Berbasis Deep Learning," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 1, p. 87, 2022, doi: 10.25126/jtiik.2022915206.
- [10] M. F. Ikhsal, B. A. Dermawan, and R. I. Adam, "Peningkatan Deteksi Kecelakaan di Jalan Raya Menggunakan Real-ESRGAN pada Citra CCTV Persimpangan Jalan," *J. Appl. Informatics Comput.*, vol. 7, no. 1, pp. 51–56, 2023, doi: 10.30871/jaic.v7i1.5562.
- [11] M. Dio Riza Pratama, B. Priyatna, S. S. Hilabi, and A. L. Hananto, "Deteksi Objek Kecelakaan Pada Kendaraan Roda Empat Menggunakan Algoritma YOLOv5," *J. Ilm. Sist. Informas*, vol. 12, no. 2, pp. 15–26, 2022.
- [12] M. Ghulam, A. Binuri, M. Amirul, T. Haryanti, and A. Peningkatan, "Penerapan Algoritma Yolo V7 Sebagai Dekteksi Kecelakaan Pada Lalu Lintas," *Prosiding-Seminar Nas. Teknol. Inf. Ilmu Komput.*, vol. 2, no. 1, pp. 35–41, 2023, [Online]. Available: <https://journal.unilak.ac.id/index.php/Semaster/article/view/18402/5832>