PENERAPAN ALGORITMA YOLO V8 UNTUK PENGENALAN PENGENDARA SEPEDA MOTOR TANPA HELM DALAM SISTEM PEMANTAUAN PELANGGARAN LALU LINTAS

1st Muhammad Haidar Rais
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
haidarrais@student.telkomuniversity.ac.id

2nd Ir. Ahmad Musnansyah, M.Sc
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
ahmadanc@telkomuniversity.ac.id

3rd Dr. Hanif Fakhrurroja, S.Si, M.T
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
haniff@telkomuniversity.ac.id

Abstrak — Penggunaan teknologi kecerdasan buatan (AI) telah menjadi relevan dalam berbagai sektor, termasuk penegakan hukum lalu lintas. Berdasarkan Undang-Undang Nomor. 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan di Indonesia, Pasal 106 ayat (8) [1] memerintahkan setiap pengendara dan penumpang sepeda motor untuk memakai helm yang memenuhi Standar Nasional Indonesia (SNI) saat melakukan perjalanan. Dalam konteks ini, sistem e-tilang telah diterapkan di banyak kota di Indonesia.

Penelitian sebelumnya telah menunjukkan bahwa penggunaan YOLO V8 dapat meningkatkan tingkat akurasi deteksi plat nomor kendaraan dengan menggunakan kekuatan komputasi yang lebih sedikit dibandingkan dengan versi sebelumnya. Metodologi CRISP DM digunakan dalam penelitian ini, yang menghasilkan tingkat akurasi yang untuk pendeteksian pengguna sepeda motor yang menggunakan helm sebesar 86% untuk deteksi pengguna motor yang tidak menggunakan helm sebesar 87% untuk pengguna kendaraan roda dua yang menggunakan helm. Selain itu, framework Flask digunakan untuk mengembangkan website yang memungkinkan deteksi pelanggaran lalu lintas tidak menggunakan helm melalui akses online.

Kata kunci — YOLOV8, E-tilang, AI, deteksi helm, CRISP DM

I. PENDAHULUAN

Penggunaan teknologi kecerdasan buatan (AI) telah menjadi relevan dalam berbagai sektor, termasuk penegakan hukum lalu lintas. Di Indonesia, penegakan aturan lalu lintas adalah prioritas utama untuk meningkatkan keselamatan berlalu lintas dan mengurangi pelanggaran.

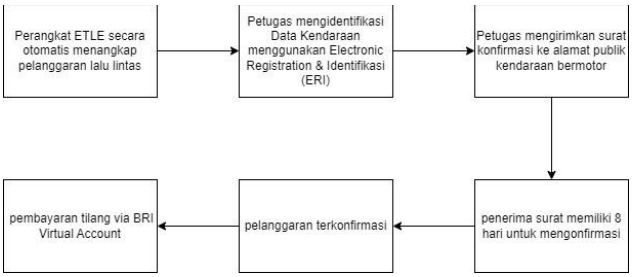
Menurut Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan di Indonesia, ketentuan mengenai penggunaan helm bagi pengendara sepeda motor terdapat dalam Pasal 106 ayat (8) [1], yang memerintahkan setiap pengendara dan penumpang sepeda motor untuk memakai helm yang memenuhi Standar Nasional Indonesia (SNI) saat melakukan perjalanan. Pelanggaran terhadap kewajiban ini, sebagaimana diatur dalam Pasal 287 [2], dapat mengakibatkan sanksi administratif dan denda bagi pelaku pelanggaran.

Salah satu masalah utama adalah pelanggaran penggunaan helm dan nomor plat kendaraan. Meskipun aturan sudah ada, pelanggaran masih sering terjadi, menurut Pusiknas Bareskrim Polri (2022) [3] 74,22% dari pelanggar adalah pengguna kendaraan roda dua. Inilah dimana e-tilang diberlakukan.

II. KAJIAN TEORI

A. Sistem E-Tilang

E-Tilang adalah sistem tiket yang diterapkan oleh pemerintah Indonesia untuk meningkatkan sistem pelayanan publik dalam pelanggaran lalu lintas. Ini adalah sistem digital yang mencatat dan mengelola pelanggaran lalu lintas secara elektronik. Penerapan e-tilang di Indonesia dipandang sebagai langkah yang diperlukan untuk mengatasi berbagai masalah seperti pemerasan, kurangnya akuntabilitas, dan pengelolaan denda yang tidak efisien. Berdasarkan artikel yang ditulis di *website* resmi Kapolda Kepulauan Riau [4] berikut adalah cara kerja sistem e-tilang:
Berikut ini mekanisme tilang menggunakan metode ETLE yang dikutip dari korlantas.polri.go.id:



Gambar 1
Mekanisme tilang ETLE eksisting

- Tahap 1

Perangkat ETLE secara otomatis menangkap pelanggaran lalu lintas yang dimonitor dan mengirimkan media barang bukti pelanggaran ke *Back Office* ETLE di RTMC Polda Metro Jaya.

- Tahap 2

Petugas mengidentifikasi Data Kendaraan menggunakan *Electronic Registration & Identification* (ERI) sebagai sumber data kendaraan.

- Tahap 3

Petugas mengirimkan surat konfirmasi ke alamat publik kendaraan bermotor untuk permohonan konfirmasi atas pelanggaran yang terjadi. Surat konfirmasi adalah langkah awal dari penindakan, yang mana pemilik kendaraan wajib mengkonfirmasi tentang kepemilikan kendaraan dan pengemudi kendaraan pada saat terjadinya pelanggaran. Jika kendaraan yang dimaksud sudah bukan menjadi kendaraan milik orang yang mendapat surat konfirmasi, maka hal itu harus segera dikonfirmasikan.

- Tahap 4

Penerima surat memiliki batas waktu sampai dengan 8 hari dari terjadinya pelanggaran untuk melakukan konfirmasi melalui *website* atau datang langsung ke kantor Sub Direktorat Penegakan Hukum.

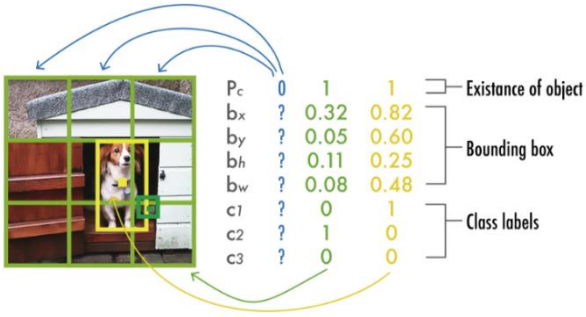
- Tahap 5

Setelah pelanggaran terkonfirmasi, petugas menerbitkan tilang dengan metode pembayaran via BRI *Virtual Account (BRIVA)* untuk setiap pelanggaran yang telah terverifikasi untuk penegakan hukum

B. YOLO V8

YOLO (*You Only Look Once*) adalah sistem deteksi objek *real-time* yang banyak digunakan dalam aplikasi visi komputer seperti robotika, mobil tanpa pengemudi, dan pemantauan video. Ini telah berevolusi melalui berbagai versi, termasuk YOLO V1 ke YOLO V8, YOLO-NAS, dan YOLO dengan *transformer*. Cara kerja YOLO adalah sebagai berikut [5]:

a. YoloV1 membagi gambar *input* menjadi *grid* S x S dan memprediksi kotak pembatas B untuk setiap elemen *grid*, bersama dengan skor kepercayaan untuk C kelas yang berbeda.
b. Setiap prediksi kotak pembatas terdiri dari lima nilai: skor kepercayaan, koordinat pusat kotak, dan tinggi dan lebar kotak relatif terhadap gambar lengkap.
c. *Output* YOLO V1 adalah *tensor* S x S x (B x 5 C), yang secara opsional diikuti oleh penekanan non-maksimum untuk menghapus deteksi duplikat.
d. YoloV1 mencapai presisi rata-rata 63,4 pada dataset PASCAL VOC2007, yang berisi 20 kelas, grid 7 x 7, dan paling banyak 2 kelas per elemen *grid*.



Gambar 2

menggambarkan model YOLO yang disederhanakan dengan *grid* tiga per tiga, tiga kelas, dan satu prediksi kelas per elemen *grid* untuk menghasilkan vektor delapan nilai.

C. *Computer Vision*

Computer Vision adalah bidang studi yang berfokus pada memungkinkan komputer untuk memahami dan menafsirkan informasi visual dari gambar atau video. Ini melibatkan pengembangan algoritma dan teknik yang memungkinkan komputer untuk mengekstrak informasi yang berarti dari data visual. Shorten menjelaskan sebagai berikut cara kerja visi computer [6]*:*

1. Algoritma visi komputer menggunakan jaringan syaraf konvolusi dalam (CNN) untuk memproses dan menganalisis data visual. CNN adalah jaringan saraf yang kuat yang dapat belajar mengenali pola dan fitur dalam gambar melalui pelatihan pada kumpulan *big data*.
2. Jaringan ini memanfaatkan kernel berparameter dan jarang terhubung untuk melestarikan karakteristik spasial gambar. Lapisan konvolusi secara berurutan menurunkan sampel resolusi spasial gambar sambil memperluas kedalaman peta fiturnya.
3. Keberhasilan CNN telah menyebabkan penerapannya dalam berbagai tugas visi komputer seperti klasifikasi gambar, deteksi objek, dan segmentasi gambar.
4. Untuk mengatasi tantangan data yang terbatas, teknik penambahan data digunakan untuk meningkatkan ukuran dan kualitas kumpulan data pelatihan. Ini termasuk transformasi

D. Deteksi Pengemudi yang tidak menggunakan helm

Deteksi pengemudi kendaraan bermotor yang tidak menggunakan helm bukanlah hal yang baru lagi, ada banyak penelitian yang dilakukan untuk mencapai hal ini dengan berbagai metode, berikut adalah cara kerja pendeteksian pengemudi yang tidak menggunakan helm sesuai yang dijelaskan Suryanto, berikut cara menggunakan YOLO [7]:

a) Pengumpulan dataset

Dataset diperoleh menggunakan video ataupun foto yang menampilkan pengemudi yang menggunakan dan tidak menggunakan helm.

b) Labeling

Labeling digunakan untuk menentukan data mana yang menampilkan pengemudi yang menggunakan helm dan yang tidak menggunakan *OpenCV*.

c) *Training* model YOLO

Dari hasil *labeling* dari data set yang sudah dikumpulkan, maka data tersebut diberikan ke YOLO untuk dipelajari oleh

sistem agar bisa mendeteksi pengemudi yang menggunakan helm atau tidak.

d) Evaluasi

Hasil pendeteksian dievaluasi dan di cek tingkat akurasinya, apakah sudah sesuai dengan yang dibutuhkan, apabila belum sesuai maka *training* model harus dilakukan ulang agar mendapatkan hasil akurasi yang diinginkan.

E. Flask

Flask adalah sebuah *micro-framework* yang dirancang untuk membuat aplikasi web dengan cepat dan efisien. *Framework* ini memberikan fleksibilitas kepada pengembang untuk menambahkan fitur sesuai kebutuhan selama implementasi. *Flask* adalah sebuah *lightweight* WSGI (*Web Server Gateway Interface*) *application framework* yang dapat digunakan untuk pengembangan *backend* maupun *frontend* [9]

F. CRISP DM



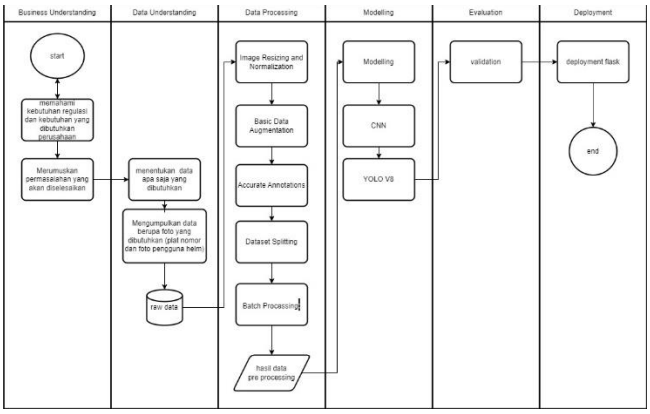
Gambar 4
Ilustrasi *framework CRISP DM*

CRISP-DM, atau *CRoss-Industry Standard Process for Data Mining,* adalah model proses yang banyak digunakan untuk mengembangkan proyek data *mining* dan penemuan pengetahuan. Menyediakan pendekatan terstruktur untuk memandu pengembangan proyek data *mining*, dengan menggabungkan prinsip-prinsip dari berbagai metodologi. Model ini berorientasi pada tujuan dan didorong oleh proses, cocok untuk proyek dengan tujuan yang jelas dan langkah-langkah yang telah ditentukan sebelumnya. CRISP-DM telah menjadi standar *de facto* untuk proyek data *mining* dan penemuan pengetahuan, tetapi bidang ini telah berkembang, dan sekarang istilah yang lebih disukai adalah data *science* [10]

III. METODE

A. Sistematika Penyelesaian Masalah

Sistematika penyelesaian menggunakan *framework* CRISP DM digambarkan di gambar 4 (A)



Gambar 5
Flowchart Sistematika Penyelesaian Masalah

a. Business Understanding

Memahami aturan yang berlaku di dalam e-tilang menurut Undang undang yang telah ditetapkan, lokasi strategis penerapan dan tata cara penilangan yang sudah ada agar dapat memahami seluruh nya

b. Data Understanding

Memahami dan mengumpulkan data yang diperlukan dan memahami apa yang bisa digunakan dan tidak bisa digunakan untuk penelitian, dari sini akan didapatkan *raw data* atau data mentah yang akan digunakan

c. Data Pre Processing

Di tahap data *pre-processing* ini, data yang sudah dimiliki pertama akan dilakukan *regulating* data, dengan cara penyamaan resolusi gambar dan ukuran gambar agar lebih mudah diproses oleh komputer dan mengurangi tingkat *error*, lalu data *cleaning* dilakukan untuk membersihkan data-data apa saja yang tidak dibutuhkan, lalu yang terakhir adalah *labeling*, *labeling* adalah proses di mana bagian yang dibutuhkan untuk dipelajari oleh *machine learning* ditandai, dalam kasus ini adalah plat nomor dan pengguna motor yang menggunakan helm dan tidak. Setelah seluruh tahap data *preprocessing* selesai dilakukan, maka didapatkan hasil data yang siap digunakan untuk *modeling*

d. Modeling

Modeling dilakukan menggunakan arsitektur *Convolutional Neural Network* (CNN) sedangkan framework yang akan digunakan adalah YOLO V8, *Modeling* akan dijalankan menggunakan *Python* sebagai bahasa pemrogramannya

e. Evaluation

Evaluasi dilakukan menggunakan *confusion matrix* untuk menghitung nilai *Recall*, *F1-Score*, *Precision*, dan *Accuracy*.

f. Deployment

Deployment akan dilakukan menggunakan *Flask*, yang masih menggunakan python sebagai bahasanya, ditujukan agar *deployment* lebih mudah

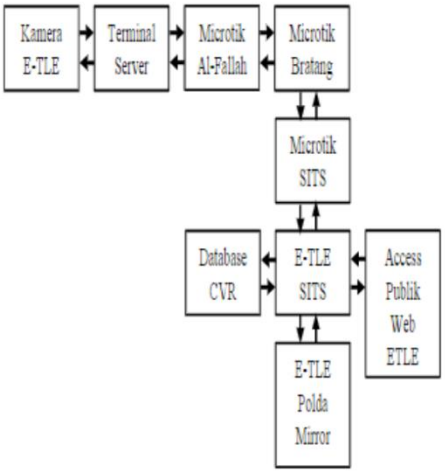
IV. HASIL DAN PEMBAHASAN

a) Business Understanding

Berdasarkan *website* KORLANTAS POLRI (*Ini Mekanisme Tilang Elektronik ETLE*, 2021) , berikut adalah mekanisme e-tilang yang sudah berlaku [11] :

1. Tahap 1
   Perangkat ETLE secara otomatis menangkap pelanggaran lalu lintas yang dimonitor dan mengirimkan media barang bukti pelanggaran ke Back Office ETLE di RTMC Polda Metro Jaya.
2. Tahap 2
   Petugas mengidentifikasi Data Kendaraan menggunakan *Electronic Registration & Identification* (ERI) sebagai sumber data kendaraan.
3. Tahap 3
   Petugas mengirimkan surat konfirmasi ke alamat publik kendaraan bermotor untuk permohonan konfirmasi atas pelanggaran yang terjadi. Surat konfirmasi adalah langkah awal dari penindakan, yang mana pemilik kendaraan wajib mengkonfirmasi tentang kepemilikan kendaraan dan pengemudi kendaraan pada saat terjadinya pelanggaran. Jika kendaraan yang dimaksud sudah bukan menjadi kendaraan milik orang yang mendapat surat konfirmasi, maka hal itu harus segera dikonfirmasikan.
4. Tahap 4
   Penerima surat memiliki batas waktu sampai dengan 8 hari dari terjadinya pelanggaran untuk melakukan konfirmasi melalui *website* atau datang langsung ke kantor Sub Direktorat Penegakan Hukum.
5. Tahap 5
   Setelah pelanggaran terkonfirmasi, petugas menerbitkan tilang dengan metode pembayaran via BRI *Virtual Account* (BRIVA) untuk setiap pelanggaran yang telah terverifikasi untuk penegakan hukum.

dan berdasarkan hasil penelitian sebelumnya [12] terkait e-tilang yang sudah berlaku di POLRESTABES Surabaya, berikut adalah diagram jaringan ETLE *eksisting* :



Gambar 6
Diagram Jaringan E-Tilang eksisting POLRESTABES Surabaya

b) Data Understanding

Untuk membuat model YOLO yang akurat dibutuhkan data gambar yang akan didapatkan dari data primer yaitu hasil rekaman dan foto, dan dari data sekunder yaitu foto yang berkaitan dengan pengguna motor, plat nomor, pengguna motor menggunakan helm, dan pengguna motor yang tidak menggunakan helm yang datanya didapatkan dari data yang disediakan oleh pengguna roboflow. Pengambilan data primer dilakukan dengan cara merekam video secara langsung menggunakan perangkat *smartphone* di malam dan siang hari. Dari data video yang didapat akan diubah menjadi gambar, konversi dilakukan 1 *frame* setiap 1 detik sampai 2 detik.

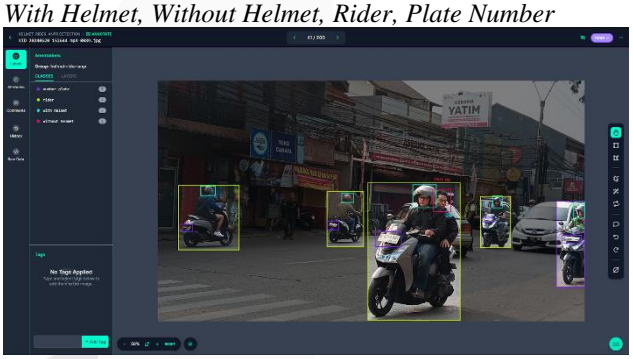


Gambar 7
Data video yang dikumpulkan

c) Data Processing

a. Labeling

Labeling dilakukan secara manual menggunakan *roboflow*, *class* klasifikasi dibuat sebanyak 4 :
With Helmet, Without Helmet, Rider, Plate Number



Gambar 8

Contoh hasil gambar yang sudah melewati proses *labeling*

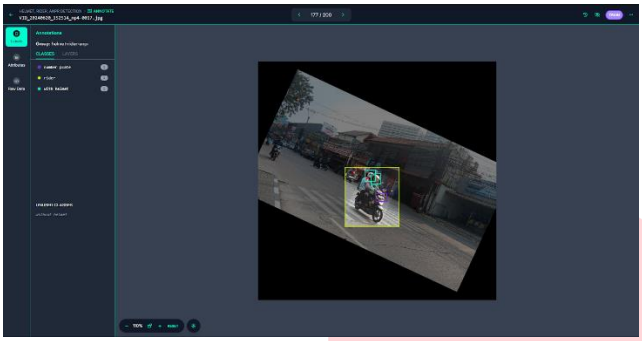
b. Preprocessing dan augmentasi

Dilakukan *preprocessing* dengan *Auto-orient* yang mengubah ukuran foto tersebut menjadi lebar gambar 640 piksel dan tinggi gambar 640 piksel dengan konfigurasi gambar fit in. Setelah melewati *preprocessing* dilakukan *augmentasi* berdasarkan hasil penelitian Harianto [13] *augmentasi* optimal yang dapat dilakukan melalui *platform roboflow* berupa:

Tabel 1
Operasi augmentasi *preprocessing* yang dilakukan

No	Preprocessing Operation	Parameter
1	Rotation Range	30-degree

2	Horizontal Flip	True
3	Brightness Range	(0.75, 1.15)
4	Output per training example	2



Gambar 9

Contoh hasil gambar yang sudah melewati proses *preprocessing* dan augmentasi

Tabel 2

Jumlah data sebelum augmentasi

Training Data	*Validation Data*	*Testing Data*
3065 gambar (70%)	874 gambar (20%)	437 gambar (10%)

Tabel 3

Jumlah data setelah augmentasi

Training Data	*Validation Data*	*Testing Data*
7083 gambar (91%)	462 gambar (6%)	254 gambar (3%)

d) Modeling

Model YOLOV8 yang digunakan dalam penelitian ini adalah YOLOV8n, dengan pertimbangan bahwa model YOLOV8n ini lebih cocok untuk perangkat yang memiliki kemampuan komputasi yang terbatas dan relatif kecil, dan juga mementingkan kecepatan, seperti yang dijelaskan dari dokumentasi YOLOV8n yang dirilis oleh *Ultralytics* [14].

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Gambar 10 Perbandingan akurasi dan kebutuhan kekuatan CPU

Modelling dilakukan menggunakan *Google Collab. Train* model dilakukan sebanyak 300 epochs dengan *image size* 640 *pixel,* mengikuti saran dari *Ultralytics* [15].

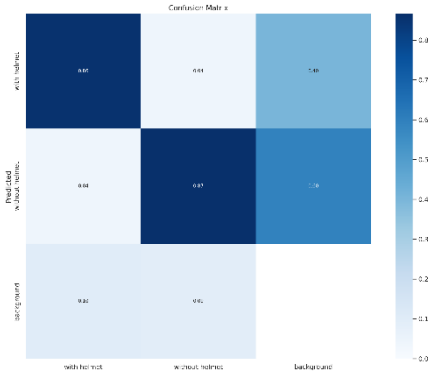
Berikut hasil validasi dari model yang sudah di *train*



Gambar 11 Hasil validasi dari model yang dibuat

e) Evaluation

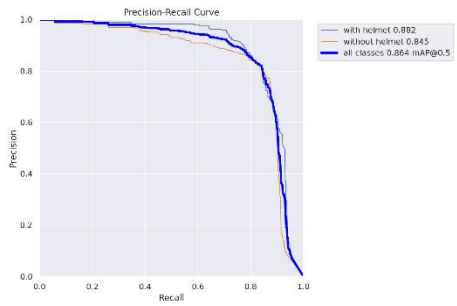
Untuk mengetahui tingkat akurasi mengklasifikasi data objek dengan tepat, diperlukan perhitungan menggunakan perhitungan dari package YOLO sendiri yaitu *confusion matrix*, yang akan menghasilkan F1, *Precision-Recall, Recall-confidence dan precision-confidence* seperti yang dijelaskan di bab 3 mengenai metode evaluasi.



Gambar 12 *Confusion Matrix*

Setiap sel dalam matriks ini memuat nilai yang merepresentasikan proporsi atau frekuensi relatif dari prediksi. Nilai-nilai pada diagonal utama matriks (dari kiri atas ke kanan bawah) menunjukkan akurasi prediksi yang benar untuk setiap kelas. Misalnya:

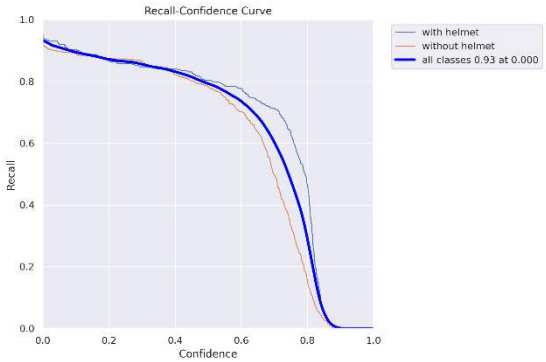
- *'With helmet'* memiliki akurasi prediksi 86%
- *'Without helmet'* diprediksi dengan akurasi 87%



Gambar 12 *Precision Recall Curve*

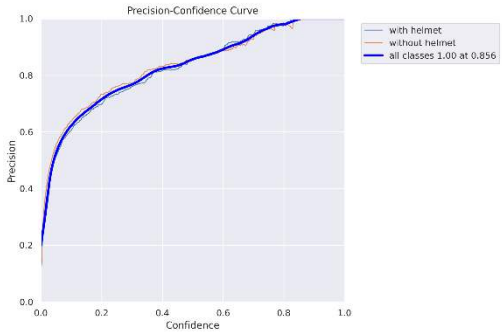
Kurva *Precision-Recall* menggambarkan hubungan antara presisi (sumbu y) dan recall (sumbu x) untuk dua kelas objek spesifik: *With helmet* : AP = 0.882, *Without helmet* : AP = 0.845

Kurva ini membantu dalam mengidentifikasi titik operasi optimal di mana keseimbangan antara presisi dan *recall* tercapai, yang krusial untuk aplikasi praktis. Nilai mAP 0.864 pada IoU 0.5 menunjukkan performa keseluruhan model yang cukup baik, mengingat kompleksitas tugas deteksi multi-kelas dalam konteks keselamatan berkendara. Analisis kurva precision-recall menunjukkan kinerja model deteksi objek yang signifikan. Kurva biru (with helmet, 0.882) mendemonstrasikan performa superior dalam identifikasi objek berhelm, sementara kurva oranye (without helmet, 0.845) menampilkan deteksi yang cukup akurat untuk objek tanpa helm. Kurva biru tebal (all classes, mAP@0.5 = 0.864) merepresentasikan performa agregat model, mengindikasikan efektivitas keseluruhan yang tinggi dalam deteksi multi-kelas. Nilai mAP@0.5 sebesar 0.864 menegaskan kemampuan model yang substansial dalam mengklasifikasikan dan mendeteksi objek secara akurat pada berbagai skenario.



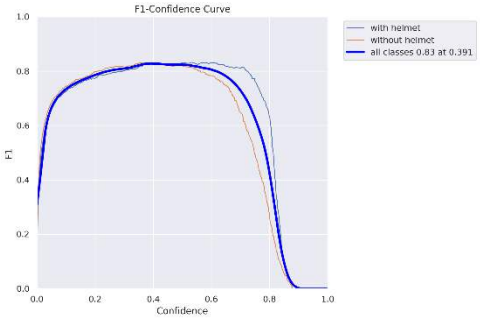
Gambar 13 *Recall Confidence Curve*

Kurva *Recall-Confidence* mengilustrasikan hubungan antara *recall* (sumbu y) dan ambang batas kepercayaan atau *confidence threshold* (sumbu x) untuk berbagai kelas objek yang dideteksi. Recall mengukur proporsi objek yang berhasil dideteksi dari total objek yang sebenarnya ada, sementara *confidence threshold* menentukan tingkat keyakinan minimum yang diperlukan untuk menerima sebuah deteksi. Kurva Biru Tebal (all classes, 1.00 at 0.856): Kurva ini menunjukkan rata-rata precision untuk semua kelas. Nilai 1.00 pada confidence 0.856 menunjukkan bahwa pada tingkat confidence tersebut, precision mencapai nilai maksimum, yaitu 100% untuk semua kelas.



Gambar 14
Precision-Confidence curve

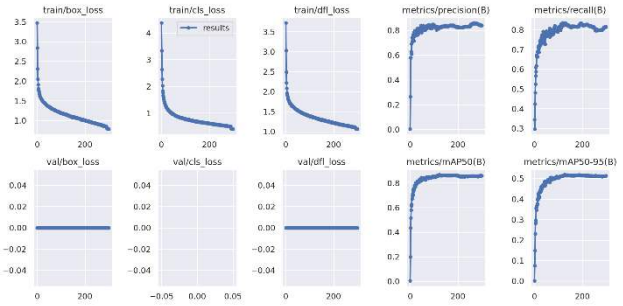
Kurva *Precision-Confidence* menggambarkan hubungan antara presisi (sumbu y) dan ambang batas kepercayaan atau *confidence threshold* (sumbu x) untuk berbagai kelas objek yang dideteksi. Presisi mengukur proporsi deteksi yang benar dari total deteksi yang dilakukan, sementara *confidence threshold* menentukan tingkat keyakinan minimum yang diperlukan untuk menerima sebuah deteksi. Analisis kurva *confidence-precision* menunjukkan performa model deteksi objek yang sangat baik. Kurva biru (with helmet) mendemonstrasikan peningkatan precision yang signifikan seiring naiknya confidence, mencapai akurasi maksimal pada confidence tertinggi. Kurva oranye (without helmet) menampilkan tren serupa meski sedikit lebih rendah. Kurva biru tebal (all classes) mencapai precision maksimum 1.00 pada confidence 0.856, mengindikasikan akurasi sempurna model pada tingkat kepercayaan tersebut untuk semua kelas. Secara keseluruhan, model menunjukkan kemampuan deteksi yang sangat akurat, terutama pada tingkat confidence yang tinggi.



Gambar 15
F1 Confidence curve

Kurva F1-Confidence menggambarkan hubungan antara nilai F1-score (sumbu y) dan tingkat kepercayaan atau *confidence threshold* (sumbu x) untuk berbagai kelas objek yang dideteksi. F1-score adalah ukuran keseimbangan antara presisi dan recall, sementara confidence threshold menentukan ambang batas probabilitas minimum untuk menerima sebuah deteksi. Analisis kurva confidence-F1 menunjukkan kinerja model deteksi objek. Kurva "with helmet" dan "without helmet" mencapai puncak F1-score sekitar confidence 0.4, lalu menurun. Kurva "all classes" mencapai F1-score tertinggi 0.83 pada confidence 0.391,

menandakan performa optimal model pada tingkat kepercayaan tersebut untuk semua kelas.



Gambar 16

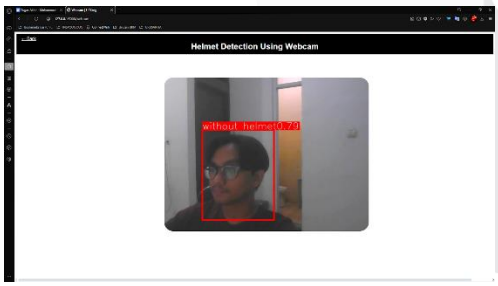
Train Loss Result Curve

Analisis grafik menunjukkan pelatihan efektif model YOLO V8 untuk deteksi helm, dengan penurunan loss signifikan dan peningkatan metrik utama. Stabilitas akhir mengindikasikan konvergensi model. Namun, perbedaan loss pelatihan-validasi menandakan potensi *overfitting*, memerlukan evaluasi lebih lanjut untuk meningkatkan generalisasi model.

f) Deployment

a. Flask

sebagai kerangka kerja pengembangan *web* untuk mengintegrasikan model *machine learning* dengan antarmuka pengguna yang interaktif. Pemilihan *Flask* didasarkan pada kesederhanaan, fleksibilitas, dan kemampuannya dalam menangani permintaan HTTP dengan efisien.Persiapan Lingkungan: Mengonfigurasi lingkungan virtual Python untuk mengisolasi dependensi proyek. Seluruh *library* yang diperlukan, termasuk Flask dan dependensi YOLO V8, di*instal* dalam lingkungan ini untuk memastikan konsistensi dan portabilitas. Pengembangan Antarmuka Web: Menggunakan *Flask*, kami merancang dan mengimplementasikan antarmuka *web* yang memungkinkan pengguna untuk mengunggah gambar kendaraan. *Templating engine Jinja2* dimanfaatkan untuk menghasilkan halaman HTML yang dinamis.



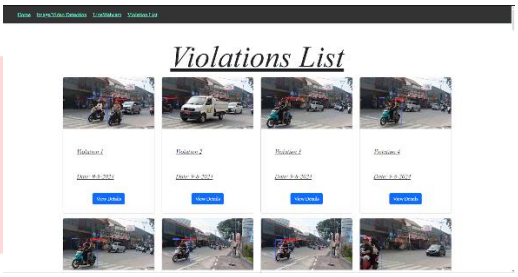
Gambar 17

Antarmuka deteksi *realtime*



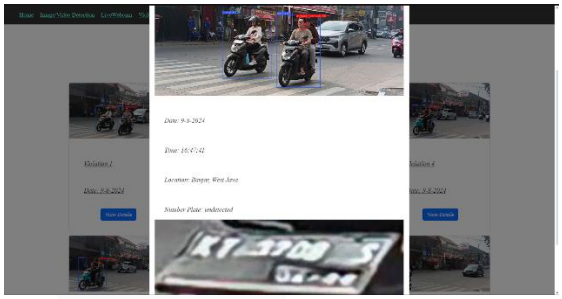
Gambar 18

Antarmuka deteksi menggunakan video atau foto



Gambar 19

Antarmuka list pelanggar



Gambar 20

Antarmuka detail list pelanggar

V. KESIMPULAN

Dari hasil percobaan pengembangan sistem dan model yang telah dilakukan, dapat disimpulkan bahwa penggunaan custom train YOLOV8n berhasil mencapai nilai F1-confidence sebesar 83%, precision-confidence sebesar 85%, mAP@0.5 sebesar 86%, dan recall-confidence sebesar 85%. Lewat CRISP DM di fase deployment, penggunaan flask sebagai framework pengembangan website, model YOLO yang sudah dilatih dapat di implementasikan dan dapat diakses melalui website oleh perangkat smartphone atau PC.

REFERENSI

[1] Pemerintah, Peraturan, Peraturan Pemerintah tentang Lalu Lintas dan Angkutan Jalan (PP Nomor 22 Pasal 106 Ayat 8 Tahun 2009).

[2] Peraturan Pemerintah, Peraturan Pemerintah tentang Lalu Lintas dan Angkutan Jalan (PP Nomor 22 Pasal 287 Ayat 8 Tahun 2009).

[3] P. B. Polri. [Online]. Available: https://pusiknas.polri.go.id/detail_artikel/_pelanggar_lalu_lintas_tak_lagi_ditindak_secara_manual#:~:text=Pengendara%20sepeda%20motor%20melakukan%20pelanggaran,wilayah%20hukum%20Polda%20Metro%20Jaya. [Accessed 16 Agustus 2024].

[4] susi, "APA ITU TILANG ELEKTRONIK ? BAGAIMANA CARA KERJA ETLE ?," 20 Mei 2022. [Online]. Available: https://pid.kepri.polri.go.id/48411/. [Accessed 28 November 2023].

[5] J. Terven, D.-M. Córdova-Esparza and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction,* vol. 5, no. 4, p. 1680, 2023.

[6] C. Shorten and T. Khoshgoftaar, "A survey on image data augmentation for Deep Learning," *Journal of Big Data,* vol. 6, no. 2, p. 2, 29.

[7] A. W. Suryanto and A. R. Kardian, "Deteksi Pelanggaran lalu lintas Tidak Menggunakan helm Dengan Yolo V4 pada sistem etle," *Jurnal Teknik Komputer,* vol. 9, no. 2, p. 129, 2023.

[8] H. Fakhrurroja, D. Pramesti, A. Rofi Hidayatullah, A. Arif Fashihullisan, H. Bangkit and D. Syamsi, "Automated License Plate Detection and Recognition using YOLOv8 and OCR With Tello Drone Camera," *2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA),* 2023.

[9] D. Ghimire, "Comparative study on Python web frameworks: Flask and Django," *Metropolia University of Applied Sciences Theses,* p. 9, 2020.

[10] F. M. Plumed, ""CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories," *IEEE Transactions on Knowledge and Data Engineering,* vol. 33, no. 8, pp. 3048-3061, 2021.

[11] Korlantas Polri, "Ini Mekanisme Tilang Elektronik ETLE," Korlantas Polri, 30 October 2023. [Online]. Available: https://korlantas.polri.go.id/news/ini-mekanisme-tilang-elektronik-etle/. [Accessed 26 December 2023].

[12] A. P. Nagendra and V. R. Sushanty, "EFEKTIVITAS PENERAPAN E-TILANG DALAM PENINDAKAN PELANGGARAN LALU LINTAS DI WILAYAH HUKUM POLRESTABES SURABAYA," *Jurnal Tatapamong,* vol. 4, no. 2, p. 149, 2022.

[13] R. A. Harianto, Y. M. Pranoto and T. P. Gunawan, "Data Augmentation and faster RCNN improve vehicle detection and recognition," in *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, Surabaya, 2021.

[14] Ultralytics, "Ultralytics," Ultralytics, 12 November 2023. [Online]. Available: https://docs.ultralytics.com/tasks/detect/#what-formats-can-i-export-a-yolov8-model-to. [Accessed 16 Agustus 2024].

[15] Ultralytics, "Tips for Best Training Results," Ultralytics, 12 November 2023. [Online]. Available: https://docs.ultralytics.com/modes/train/#what-are-the-common-training-settings-and-how-do-i-configure-them. [Accessed 16 Agustus 2024].

[16] J. Sanui, C. Thepporn and P. Chusuwan, "Thai license plate recognition using SSD mobilenet and easyocr," *Proceedings of the 2024 6th International Conference on Image Processing and Machine Vision,* 2024.