SISTEM PENDETEKSI OBJEK ROKOK MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO) PADA KAWASAN LARANGAN MEROKOK

LAPORAN TUGAS AKHIR

Laporan ini Disusun Untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Sarjana Strata 1 (S1) Pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung

DISUSUN OLEH :

MUHAMMAD AZMI ABDILLAH SULAEMAN
32602000044

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025

CIGARETTE OBJECT DETECTION SYSTEM USING THE YOU ONLY LOOK ONCE (YOLO) ALGORITHM IN A NO SMOKING AREA

FINAL PROJECT

Proposed to complete the requirement to obtain a bachelor's degree (S1)
at Informatics Engineering Departement of Industrial Technology Faculty
Sultan Agung Islamic University

Arranged By **:**

MUHAMMAD AZMI ABDILLAH SULAEMAN

32602000044

MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG

2025

LEMBAR PENGESAHAN TUGAS AKHIR

SISTEM PENDETEKSI OBJEK ROKOK MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO) PADA KAWASAN LARANGAN MEROKOK

MUHAMMAD AZMI ABDILLAH SULAEMAN
NIM 32602000044

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 03 Juni 2025

TIM PENGUJI UJIAN SARJANA :

Bagus Satrio Waluyo Poetro, S.Kom., M.Cs NIDN.1027118801 (Ketua Penguji)	.....	05/06/2025 .....
Dedy Kurniadi, ST., M.Kom NIDN.0622058802 (Anggota Penguji)	.....	10/06/2025 .....
Sam Farisa Chaerul Haviana, ST., M.Kom NIDN.0628028602 (Pembimbing)	.....	10/06/2025 .....

Semarang, 05 Juni 2025
Mengetahui,
Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, ST., MIT
NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama	: Muhammad Azmi Abdillah Sulaeman
NIM	: 32602000044
Judul Tugas Akhir	: Sistem Pendeteksi Objek Rokok Menggunakan Algoritma *You Only Look Once* (Yolo) Pada Kawasan Larangan Merokok

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apbila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 10 Juni 2025

Yang Menyatakan,

Muhammad Azmi Abdillah S.

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Muhammad Azmi Abdillah Sulaeman
NIM : 32602000044
Program Studi : Teknik Informatika
Fakultas : Teknologi industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : "Sistem Pendeteksi Objek Rokok Menggunakan Algoritma You Only Look Once (Yolo) Pada Kawasan Larangan Merokok" Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 10 Juni 2025
Yang menyatakan,

Muhammad Azmi Abdillah S.

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadirat Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul "Sistem Pendeteksi Objek Rokok Menggunakan Algoritma *You Only Look Once* (YOLO) Pada Kawasan Larangan Merokok" ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada :

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Novi Marlyana, S.T., M.T., IPU.,ASEAN. Eng
3. Dosen Pembimbing penulis Bapak Sam Farisa Chaerul Haviana, ST.M.Kom yang telah meluangkan waktu dan memberi ilmu..
4. Orang tua penulis, Ayahanda drg. Sulaeman, M.H dan Ibunda Tin Wartini, S.Pd yang telah mengizinkan untuk menyelesaikan laporan ini serta memberikan dukungan dan doa agar tugas akhir ini dapat berjalan dengan lancar.
5. Izzuma Tasya Rismayanti, S.H yang telah menemani dan selalu memberi dukungan kepada saya untuk menyelesaikan laporan ini.

Semarang, 20 Mei 2025

Muhammad Azmi Abdillah S.

DAFTAR ISI

HALAMAN JUDUL ..... **i**

LEMBAR PENGESAHAN TUGAS AKHIR ..... **iii**

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR ..... **iv**

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH .....**v**

KATA PENGANTAR ..... **vi**

DAFTAR ISI ..... **vii**

DAFTAR TABEL .....**x**

DAFTAR GAMBAR ..... **xi**

ABSTRAK ..... **xii**

BAB I PENDAHULUAN **1**

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Pembatasan Masalah 2

1.4 Tujuan Tugas Akhir 2

1.5 Manfaat 3

1.6 Sistematika Penulisan.....3

BAB II TINJAUAN PUSTAKA DAN DASAR TEORI **4**

2.1 Tinjauan Pustaka 4

2.2 Dasar Teori 10

2.2.1 *Object Detection* 10

2.2.2 *Convolutional Neural Network* (CNN).....11

2.2.3 *You Only Look Once* (YOLO) 12

2.2.4 Ultralytics dan YOLOv8 13
2.2.5 *Bounding Box* dan *Intersection Over Union* (IoU)....16
2.2.6 Evaluasi Model - *Precision, Recall, dan mAP* 16
2.2.7 Data *Augmentation* 16
2.2.8 Roboflow 16
BAB III METODOLOGI PENELITIAN 18
3.1 Deskripsi Sistem....18
3.2 Studi Literatur 18
3.3 Diagram Arsitektur Sistem....18
3.4 Perancangan Model 20
3.5.1 Alur Perancangan Model 20
3.5.2 *Collecting* Data 21
3.5.3 Anotasi Data 22
3.5.4 *Preprocessing* Data....22
3.5.5 Augmentasi Data 23
3.5.6 Training 24
3.5.7 Perbandingan Parameter *Optimizer* pada Pelatihan Model YOLOv8 28
3.5.8 Evaluasi 29
BAB IV HASIL DAN ANALISIS PENELITIAN....30
4.1 Hasil dan Analisis 30
4.1.1 Proses Pelatihan Model 30
4.1.2 Perbandingan Penggunaan *Optimizer*....30
4.1.3 Hasil Evaluasi Model....32

4.1.4 Visualisasi Hasil Evaluasi 33
4.1.5 Kesimpulan Pemilihan *Optimizer* 39
4.2 Hasil Implementasi Aplikasi 39
4.2.1 Tampilan *User Interface* 39
4.2.2 Pengujian Aplikasi Deteksi Rokok 44
BAB V KESIMPULAN DAN SARAN 47
5.1 Kesimpulan 47
5.2 Saran 47
DAFTAR PUSTAKA 49

UNIVERSITAS ISLAM SULTAN AGUNG
UNISSULA
جامعة سلطان أجونج الإسلامية

DAFTAR TABEL

Tabel 2. 1 Tinjaukan Pustaka 6
Tabel 3. 1 Perbandingan Parameter *Optimizer* 28
Tabel 4. 1 Hasil Evaluasi Model 32
Tabel 4. 2 Pengujian Menggunakan *Webcam* 44
Tabel 4. 3 Pengujian Menggunakan *Upload File* 45
Tabel 4. 4 Pengujian Menggunakan IP *Camera* 46

DAFTAR GAMBAR

Gambar 2. 1 Contoh Object Detection 11
Gambar 2. 2 Arsitektur *Convolutional Neural Network* 12
Gambar 2. 3 Arstitektur YOLOv8(Shen dkk, 2023) 14
Gambar 3. 1 Diagram Arsitektur Sistem 19
Gambar 3. 2 Alur Perancangan Model 20
Gambar 3. 3 Dataset 21
Gambar 3. 4 Anotasi Data 22
Gambar 3. 5 *Preprocessing* Data 23
Gambar 3. 6 *Split* Data 23
Gambar 3. 7 Augmentasi Data 24
Gambar 4. 1 *Confusion Matrix Optimizer* Auto 33
Gambar 4. 2 Grafik Hasil Pelatihan *Optimizer* Auto 34
Gambar 4. 3 *Confusion Matrix Optimizer* Adam 35
Gambar 4. 4 Grafik Hasil Pelatihan Optimizer Adam 36
Gambar 4. 5 *Confusion Matrix Optimizer* SGD 37
Gambar 4. 6 Grafik Hasil Pelatihan Optimizer SGD 38
Gambar 4. 7 Tampilan Utama Aplikasi 39
Gambar 4. 8 *Upload Model File* 40
Gambar 4. 9 *Video Source* 40
Gambar 4. 10 *Video Source Upload File* 41
Gambar 4. 11 *Video Source* IP *Camera* 41
Gambar 4. 12 Classes 41
Gambar 4. 13 *Tracking Object Option* 42
Gambar 4. 14 *Setting Confidence Threshold* 42
Gambar 4. 15 Tampilan Input dan Output Video 42
Gambar 4. 16 Notifikasi 43
Gambar 4. 17 Fitur *Object Counting* 43
Gambar 4. 18 FPS 43

ABSTRAK

Merokok di area publik masih menjadi masalah yang sulit dikendalikan, terutama di lingkungan yang telah menerapkan kawasan bebas rokok. Oleh karena itu, dibutuhkan sebuah sistem otomatis yang mampu mendeteksi aktivitas merokok secara *real-time*. Penelitian ini bertujuan untuk membangun sebuah sistem deteksi rokok berbasis model *You Only Look Once version 8* (YOLOv8) yang diintegrasikan dengan aplikasi web menggunakan Streamlit. Dataset yang digunakan diperoleh dari platform Roboflow dan dilatih menggunakan beberapa parameter, termasuk *optimizer* yang divariasikan untuk mengetahui pengaruhnya terhadap performa model. Model dilatih dengan menggunakan tiga jenis *optimizer*, yaitu SGD, Adam, dan *auto*. Hasil evaluasi menunjukkan bahwa *auto optimizer*, yang secara otomatis memilih algoritma AdamW, memberikan performa terbaik dengan nilai *precision* sebesar 0.878, *recall* sebesar 0.795, dan mAP50 sebesar 0.863. Selain itu, sistem diuji menggunakan tiga jenis sumber input, yaitu *webcam*, IP *camera*, dan *file* upload. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi objek rokok secara akurat dan menampilkannya melalui antarmuka pengguna dengan *bounding box* dan perhitungan jumlah objek terdeteksi. Dari hasil penelitian ini, dapat disimpulkan bahwa sistem deteksi rokok yang dibangun mampu bekerja secara efektif dalam berbagai kondisi *input* dan dapat digunakan sebagai alat bantu pengawasan untuk mendeteksi aktivitas merokok secara otomatis di area terlarang.

Kata kunci: YOLOv8, deteksi rokok, Streamlit, *object detection*, *optimizer*

ABSTRACT

Smoking in public areas remains a challenging issue to control, particularly in environments that have implemented smoke-free zones. Therefore, an automated system capable of detecting smoking activity in real-time is needed. This study aims to develop a cigarette detection system based on the You Only Look Once version 8 (YOLOv8) model integrated with a web application using Streamlit. The dataset used was obtained from the Roboflow platform and trained using several parameters, including varied optimizers to observe their impact on model performance. The model was trained using three types of optimizers: SGD, Adam, and Auto. Evaluation results show that the Auto optimizer, which automatically selected the AdamW algorithm, achieved the best performance with a precision of 0.878, recall of 0.795, and mAP50 of 0.863. Furthermore, the system was tested using three types of input sources: webcam, IP camera, and file upload. The testing results indicate that the system is capable of accurately detecting cigarette objects and displaying them through the user interface with bounding boxes and object counting. Based on the results of this research, it can be concluded that the developed cigarette detection system is able to function effectively across various input conditions and can be used as a surveillance tool to automatically detect smoking activity in prohibited areas.

***Keywords**: YOLOv8, cigarette detection, Streamlit, object detection, optimizer*

BAB I
PENDAHULUAN

1.1 Latar Belakang

Merokok adalah salah satu dari banyaknya kegiatan yang dari dulu hingga saat ini menjadi sebuah masalah yang terjadi di masyarakat. Permasalahan ini terjadi tidak hanya dalam skala nasional, akan tetapi sudah menjadi permasalahan dalam skala internasional. Merokok telah menjadi hal yang biasa terjadi di masyarakat, mulai dari kalangan atas hingga kalang menengah ke bawah dan sering kali kita temukan pada kehidupan sehari-hari. Pada skala internasional, menurut data saat ini angka perokok telah mencapai 1,3 milyar jiwa, dengan persentase sebesar 84% perokok berasal dari negara yang tergolong dalam kategori miskin bahkan Indonesia saat ini merupakan peringkat ketiga setelah negara China dan negara India. Dampak permasalahan yang timbul dari aktivitas merokok tidak hanya dari segi kesehatan saja, akan tetapi aktivitas merokok juga membawa dampak permasalahan dari segi lingkungan dan peraturan di masyarakat. Tindakan merokok secara ilegal di tempat yang merupakan kawasan bebas rokok, seperti rumah sakit, taman bermain anak, tempat ibadah, tempat kerja, transportasi umum dan kawasan bebas rokok lainnya, itu juga merupakan dampak permasalahan yang timbul dari aktivitas merokok (Hasa dkk., 2024).

Pada saat ini, sedang populer pada kalangan *programmer* sebuah teknologi yang dapat mendeteksi sebuah objek yang berbasis *deep learning*. Teknologi ini dapat mendeteksi sebuah objek dalam bentuk gambar ataupun dalam bentuk video dan teknologi ini memungkinkan sekali untuk mendeteksi sebuah objek rokok. YOLO atau *You Only Look Once* merupakan salah satu algoritma dari banyaknya algoritma yang sedang populer saat ini dalam teknologi pengenalan objek secara *real-time*. Algoritma YOLO dapat mendeteksi objek dengan cara membagi citra ke dalam beberapa *grid* dan algoritma YOLO ini dapat mendeteksi objek secara *real-time* (Aras dkk., 2024). Algoritma YOLO memiliki kelebihan dalam melakukan proses deteksi yang dapat membaca FPS video dengan sangat cepat, akan tetapi untuk akurasi yang dihasilkan oleh algoritma YOLO tidak berbeda jauh dengan

algoritma-algoritma *deep learning* lainnya (Gojali dkk., 2023).

Menanggapi permasalahan tersebut, dikembangkanlah sebuah sistem yang mampu mendeteksi keberadaan objek rokok guna mengidentifikasi individu yang merokok di area yang seharusnya bebas dari aktivitas merokok. Dalam penelitian ini, penulis merancang sistem berbasis algoritma YOLOv8, yang merupakan versi lanjutan dari metode YOLO yang telah digunakan pada penelitian-penelitian sebelumnya. Sistem ini mampu melakukan deteksi secara *real-time* dan diharapkan dapat memberikan informasi yang akurat serta tepat waktu bagi instansi atau pihak-pihak yang membutuhkan dukungan pengawasan secara otomatis.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, masalah penelitian ini dapat dirumuskan yaitu bagaimana mengetahui aktivitas merokok pada kawasan bebas rokok secara otomatis?

1.3 Pembatasan Masalah

Adapun Batasan masalah dari penelitian ini yaitu:

1. Penelitian ini membatasi untuk mendeteksi objek rokok dan tidak melibatkan asap rokok atau hal lainnya.
2. Sistem hanya dapat mendeteksi objek rokok dengan baik dalam kondisi pencahayaan yang baik.
3. Sistem hanya dapat mendeteksi objek rokok berwarna putih dan tidak dapat mendeteksi rokok berwarna selain putih maupun rokok cerutu.
4. Menggunakan metode *Convolution Neural Networks* (CNN) dengan algoritma *You Only Look Once* (YOLO).

1.4 Tujuan Tugas Akhir

Tujuan dari penelitian tugas akhir ini adalah merancang sistem pendeteksi objek secara *real-time* menggunakan metode *Convolutional Neural Networks* (CNN) dan algoritma *You Only Look Once* (YOLO) yang dapat mendeteksi objek rokok pada kawasan bebas rokok.

1.5 Manfaat

Berdasarkan permasalahan dan tujuan yang telah diuraikan, penelitian ini diharapkan dapat membantu dalam mengurangi dampak permasalahan yang terjadi akibat merokok yaitu aktivitas merokok secara ilegal di kawasan bebas rokok dengan pemantauan secara *realtime*.

1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir yang dipakai oleh penulis adalah sebagai berikut :

BAB I : PENDAHULUAN

Bab ini memuat latar belakang masalah, rumusan masalah, pembatasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan skripsi secara menyeluruh.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini berisi teori-teori dasar yang mendasari penelitian, termasuk penjelasan terkait deteksi objek, algoritma YOLO, serta kajian dari penelitian terdahulu yang relevan.

BAB III : METODE PENELITIAN

Bab ini menguraikan metode yang digunakan dalam pengembangan sistem, mulai dari tahapan perancangan sistem, pengumpulan data, pelatihan model, pemilihan parameter, hingga proses evaluasi dan pengujian model.

BAB IV : HASIL DAN ANALISIS PENELITIAN

Bab ini menyajikan hasil implementasi sistem, performa model berdasarkan pengujian yang dilakukan, serta analisis terhadap kinerja sistem berdasarkan parameter dan data yang digunakan.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian serta saran.

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

YOLOv8 adalah versi terbaru dari YOLO oleh Ultralytics. Sebagai model mutakhir dan canggih, YOLOv8 dibangun dengan memperkenalkan fitur-fitur baru dan peningkatan untuk meningkatkan kinerja, fleksibilitas, dan efisiensi. YOLOv8 mendukung berbagai tugas *Artificial Intelegence* atau AI, seperti deteksi, segmentasi, estimasi pose, pelacakan, dan klasifikasi. Fleksibilitas ini memungkinkan pengguna untuk memanfaatkan kemampuan YOLOv8 di berbagai aplikasi.

Sebelumnya, telah dilakukan oleh Lejar Satya, M. Ridwan Dwi Septian, Mochammad Wisuda Sarjono, Margi Cahyanti dan Ericks Rachmat Swedia, penelitian dengan topik pembangunan sistem pendeteksi pelat nomor polisi dengan arsitektur YOLOv8 secara otomatis. Dalam penelitian tersebut menghasilkan model dengan akurasi yang baik dengan skor *precision* sebesar 0,871 dan nilai *recall* sebesar 0,85 (Cahyanti, 2023).

Pada penelitian lain, yang telah dilakukan oleh Rayhand Aska Mukhlis, Maman Somantri dan M. Arfan, dengan topik pembangunan sistem keamanan deteksi pengunjung dan petugas pada tambak udang dengan menggunakan algoritma YOLOv8 dengan menggunakan 80 *epochs* pada tahap *training* model menunjukkan hasil yang cukup baik. Nilai rata-rata akurasi yang dihasilkan dalam mendeteksi objek petugas dan pengunjung mencapai 80,8%, kemudian nilai akurasi pendeteksian pada malam hari mencapai 96,7% (Mukhlis dkk., 2023).

Adapun dalam penelitian lainnya dengan topik penerapan algoritma YOLOv8 untuk identifikasi abjad bahasa isyarat Indonesia, yang di lakukan oleh Agung Ma'ruf dan Mardi Hardjianto, menghasilkan hasil yang memuaskan. Dari 26 abjad, 25 abjad terdeteksi dengan sempurna dan hanya 1 abjad yang mencapai persentase sebesar 96,5%. Pada proses evaluasi menghasilkan nilai akurasi sebesar 99,8%, presisi sebesar 99,4% dan nilai *recall* sebesar 99,8% (Hardjianto, 2023).

Selain itu, pada penelitian yang dilakukan oleh Arie Ardiansyah, Joko Triloka

dan Indera, dengan topik pembangunan sistem untuk evaluasi kinerja model YOLOv8 dalam deteksi kesegaran buah, yang dilatih dengan menggunakan 100 *epochs* dan ukuran *batch* 8 menghasilkan akurasi rata-rata sebesar 88%. Metrik umum pada tahap evaluasi menunjukkan nilai presisi sebesar 74% dan nilai *recall* sebesar 75% pada *epoch* ke 100 (Ardiansyah dkk., 2024).

Sejumlah penelitian sebelumnya telah memanfaatkan algoritma YOLOv8 dalam berbagai bidang dengan hasil yang cukup signifikan.(Gita Agustiguna Sidik, 2024) dalam studinya mengenai deteksi tindak kekerasan dan perundungan pada anak menggunakan model YOLOv8 berhasil mencapai akurasi sebesar 85%, dengan *precision* 81,8% dan *recall* 90%, menunjukkan performa yang cukup baik dalam mendeteksi perilaku negatif tersebut.

Di sisi lain, (Maulana dan Noviana, 2023) menerapkan YOLOv8 untuk pengembangan model deteksi udang, di mana proses pelatihan dilakukan selama 300 *epoch*. Hasil yang diperoleh menunjukkan *precision* rata-rata mencapai 92,57%, *recall* sebesar 88,2%, serta akurasi model mencapai 93,2%, menandakan efektivitas algoritma ini dalam klasifikasi objek di bidang perikanan.

Penelitian lain yang berfokus pada sistem deteksi kelengkapan penambat pada jalur rel kereta api juga mengadopsi YOLOv8 sebagai basis deteksi. Sistem ini mampu membedakan jenis penambat dengan akurasi tertinggi 92,47% pada tipe E *Clip*, disertai nilai RMSE terendah sebesar 0,77. Dari sumber video, akurasi maksimal dicapai pada penambat Ka *Clip* dengan angka 91,76% dan *error* 0,1. Sementara itu, pada pengujian dengan *webcam*, akurasi terbaik, yakni 76,37%, diperoleh ketika kamera diposisikan pada sudut 170° dan kecepatan lori 15 km/jam, dengan nilai RMSE terendah sebesar 1,96 pada kecepatan 25 km/jam (Harkansas, 2024)

Sementara itu, (Meilita and Yustanti, 2024) dalam penelitiannya tentang deteksi penyakit kulit kucing melalui aplikasi Android memanfaatkan YOLOv8 untuk identifikasi *real-time* terhadap beberapa jenis penyakit. Dengan *dataset* sebanyak 1498 gambar yang dibagi dalam proporsi 70:30, 80:20, dan 90:10 untuk pelatihan dan pengujian, model yang dihasilkan mencatat skor mAP sebesar 0.788, F1-score 0.75, *precision* 0.727, dan *recall* 0.769 pada konfigurasi terbaiknya, yaitu

100 *epoch*, *batch size* 16, *optimizer* AdamW, dan *learning rate* 0.00125. Uji coba pada fitur kamera mendapati akurasi optimal 0.92 pada jarak 20 cm, sedangkan fitur *import* gambar menunjukkan konsistensi akurasi serupa pada lima perangkat *smartphone* yang berbeda.

(Yudha dkk., 2021) mengembangkan sistem deteksi objek pada robot *transporter* yang memanfaatkan Darknet YOLOv8. Sistem ini terbukti mampu melakukan deteksi dan pelacakan secara *real-time* dengan tingkat *confidence* rata-rata 0,6792, menandakan kinerjanya yang andal dalam lingkungan dinamis.

Selain itu, penelitian oleh (Putu dkk, 2024) menggunakan YOLOv8 untuk mengidentifikasi tingkat kekeringan daun kelor dalam proses pengeringan secara *real-time*. Model yang dilatih dengan *dataset* gambar daun ini berhasil menunjukkan performa tinggi yang tercermin dari hasil evaluasi menggunakan *confusion matrix*, ROC *curve, precision-recall curve*, serta *cross-validation*. Pada *precision-recall curve*, nilai *precision* tercatat sebesar 0.67, sedangkan *recall* mencapai 1.0, yang mengindikasikan kemampuan model dalam mengenali seluruh daun yang sudah kering tanpa ada yang terlewat. Hasil ini membuktikan potensi YOLOv8 dalam mendukung proses *monitoring* kualitas daun secara efisien.

Tabel 2. 1 Tinjauan Pustaka

NO	Judul	Peneliti	Model	Hasil
1	PERANCANGAN POLA ALGORITMA DAN CITRA SISTEM KEAMANAN DENGAN MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*	Rayhand Aska Mukhlis, Maman Somantri dan M. Arfan	YOLOv8	Dalam proses identifikasi objek manusia, sistem memperoleh rata-rata akurasi deteksi sebesar 80,8%. Untuk pengenalan aktivitas, tingkat akurasi yang dicapai yaitu sebesar 94,25% pada aktivitas berjalan, 75% pada aktivitas berlari, 92,5% untuk aktivitas jongkok, serta 80% pada aktivitas duduk. Dalam pengenalan objek spesifik seperti petugas keamanan, sistem mampu mencapai tingkat akurasi 96,7%.
2	SISTEM	Lejar Satya, M	YOLOv8	Model yang dibangun

NO	Judul	Peneliti	Model	Hasil
	PENDETEKSI PLAT NOMOR POLISI KENDARAAN DENGAN MENGGUNAKAN ARSITEKTUR YOLOV8	Ridwan Dwi Septian, Mochammad Wisuda Sarjono, Margi Cahyanti dan Ericks Rachmat Swedia		menggunakan YOLOv8 mampu mengidentifikasi berbagai jenis kendaraan serta pelat nomor dengan tingkat akurasi yang cukup tinggi. Sistem ini mencatat nilai *precision* sebesar 0,871 dan *recall* sebesar 0,85, yang menunjukkan performa deteksi yang andal
3	PENERAPAN ALGORITME *YOU ONLY LOOK ONCE VERSION* 8 UNTUK IDENTIFIKASI ABJAD BAHASA ISYARAT INDONESIA	Agung Ma'ruf, Mardi Hardjianto	YOLOv8	Pengujian terhadap konfigurasi *hyperparameter* yang dilakukan sebanyak 100 iterasi dengan ukuran *batch* sebesar 16 dan *learning rate* sebesar 0,01 menghasilkan performa model yang sangat baik, dengan akurasi mencapai 99,8%, *precision* sebesar 99,4%, serta *recall* sebesar 99,8%.
4	EVALUASI KINERJA MODEL YOLOV8 DALAM DETEKSI KESEGARAN BUAH	Arie Ardiansyah, Joko Triloka, dan Indera	YOLOv8	Penelitian ini menghasilkan *confusion matrix* dengan akurasi sebesar 88%. Pada tingkat kepercayaan yang tinggi, model mampu mengenali tingkat kesegaran buah dengan rata-rata *precision* sebesar 97%. Sebaliknya, saat digunakan pada tingkat kepercayaan yang rendah, rata-rata nilai *recall* mencapai 87%. Ketika digunakan pada tingkat kepercayaan moderat (0.50), rata-rata nilai *precision*, *recall*, dan F1-Score tercatat sebesar 73%. Sementara itu, pengujian di luar batas

NO	Judul	Peneliti	Model	Hasil
				kepercayaan tersebut menunjukkan nilai *precision* sebesar 74% dan *recall* sebesar 75%.
5	DETEKSI TINDAK KEKERASAN DAN PERUNDUNGAN PADA ANAK BERBASIS YOLOV8 (*YOU ONLY LOOK ONCE*)	Gita Agustina Sidik	YOLOv8	Dalam penelitian ini, diperoleh hasil evaluasi model dengan tingkat akurasi mencapai 85%, nilai *precision* sebesar 81,8%, serta *recall* sebesar 90%.
6	*TRAINING CUSTOM* MODEL DETEKSI UDANG MENGGUNAKAN YOLOV8	Muhamad Irfan Maulana dan Rina Noviana	YOLOv8	Penelitian ini melibatkan proses pelatihan model selama 300 *epoch*. Hasil evaluasi menunjukkan bahwa model memperoleh nilai *precision* rata-rata sebesar 92,57%, *recall* sebesar 88,2%, serta tingkat akurasi sebesar 93,2%.
7	PENGEMBANGAN SISTEM APLIKASI DETEKSI KELENGKAPAN PENAMBAT PADA JALAN REL BERBASI *YOU ONLY LOOK ONCE* VERSI 8	Daud Daffa Harkansas, Sunaryo dan Agustinus Prasetyo E. W.	YOLOv8	Pengujian deteksi berbasis gambar menunjukkan bahwa komponen penambat E *Clip* menghasilkan rata-rata akurasi tertinggi sebesar 92,47%, disertai nilai galat (RMSE) terendah yakni 0,77. Sementara itu, pengujian dari sumber video menunjukkan bahwa penambat Ka *Clip* mencatat akurasi tertinggi sebesar 91,76% dengan nilai RMSE terendah sebesar 0,1. Untuk sumber *webcam*, akurasi tertinggi dicapai saat kamera dipasang pada sudut 170° dengan kecepatan lori 15 km/jam, yaitu sebesar 76,37%.

NO	Judul	Peneliti	Model	Hasil
				Adapun nilai RMSE terendah tercatat pada sudut kamera 170° dengan kecepatan lori 25 km/jam, dengan nilai galat sebesar 1,96.
8	SISTEM DETEKSI PENYAKIT KULIT KUCING MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO) V8	Bunga Meilita dan Wiyli Yustanti	YOLOv8	Proses pelatihan model dilakukan melalui pengujian berbagai konfigurasi *hyperparameter*. Dari hasil pengujian tersebut, diperoleh skor mAP sebesar 0,788, F1-score sebesar 0,75, *precision* sebesar 0,727, dan *recall* sebesar 0,769. Parameter terbaik yang digunakan dalam pelatihan ini meliputi jumlah *epoch* sebanyak 100, ukuran *batch* 16, penggunaan *optimizer* AdamW, serta *learning rate* sebesar 0,00125.
9	PERANCANGAN SISTEM DETEKSI OBJEK PADA ROBOT *TRANSPORTER* MENGGUNAKAN METODE DARKNET YOLOV8	Adhitya Ari Yudha, Yudha Febrian, Rasyid Ammary Yahya, Naufal Indra Ardhana, Melati Wangi Windari dan Ardy Seto Priambodo	YOLOv8	Pada penelitian ini, algoritma YOLOv8 digunakan untuk melakukan deteksi objek dalam kondisi yang dinamis. Sistem mampu mendeteksi serta melacak objek secara langsung (*real-time*). Hasil analisis performa menunjukkan bahwa sistem memiliki tingkat akurasi yang cukup tinggi dalam mengidentifikasi objek, dengan nilai rata-rata *Confidence* mencapai 0,6792.
10	IDENTIFIKASI PENGENALAN POLA DAUN KELOR KERING	I Putu Candra Jumariana dan Putu Sugiartawan	YOLOv8	Metode evaluasi pada penelitian ini, termasuk *confusion matrix*, kurva ROC, kurva presisi-recall, dan cross-*validation* menunjukkan

NO	Judul	Peneliti	Model	Hasil
	DENGAN YOLO V8			kinerja yang baik dari model. *Precision-recall curve* yang menunjukkan nilai *precision* sebesar 0.67 dan *recall* sebesar 1.0, yang mengindikasikan bahwa model memiliki tingkat presisi yang baik dalam mengenali daun kelor kering, sementara *recall*-nya mencapai nilai maksimum, menunjukkan bahwa model dapat mengidentifikasi semua daun kelor yang sebenarnya sudah kering

2.2 Dasar Teori

2.2.1 *Object Detection*

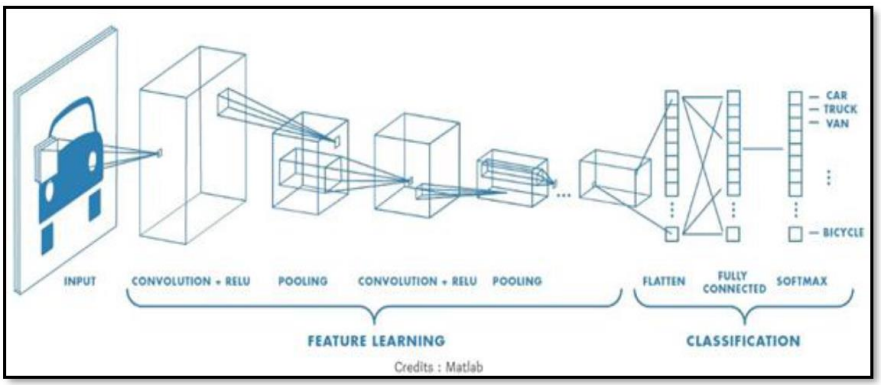
Object Detection merupakan salah satu teknik dalam bidang *computer vision* yang bertujuan untuk mengenali dan menemukan objek-objek tertentu di dalam gambar maupun video. Algoritma yang digunakan dalam proses ini umumnya mengandalkan *machine learning* atau *deep learning* untuk menghasilkan deteksi yang akurat. Jika manusia mampu secara alami mengidentifikasi dan menemukan objek dalam waktu singkat, komputer memerlukan proses komputasi yang lebih kompleks untuk mencapai hal serupa. Tujuan utama dari teknik ini adalah untuk meniru kemampuan visual manusia dalam mengenali benda menggunakan sistem komputer.

Gambar 2. 1 Contoh *Object Detection*

Pada prinsipnya, *object detection* bekerja dengan menentukan posisi objek dalam sebuah gambar serta menggambar *bounding box* yang mengelilingi objek tersebut. Proses ini umumnya terdiri dari dua tahap, yaitu mengklasifikasikan jenis objek yang terdeteksi dan selanjutnya menentukan lokasinya dengan menggambar kotak pembatas. Sekilas, konsep *image classification* dan *object detection* tampak mirip. Namun, secara garis besar, *image classification* hanya mengkategorikan gambar ke dalam kelas tertentu, sedangkan *object detection* tidak hanya mengidentifikasi jenis objek, tetapi juga menentukan posisi serta jumlah *instance* objek yang ada dalam gambar (Prisky Ratna Aningtiyas, 2020).

2.2.2 *Convolutional Neural Network* (CNN)

Convolutional Neural Networks (CNN) merupakan salah satu arsitektur jaringan *neural* yang dirancang secara khusus untuk meningkatkan efektivitas dalam tugas klasifikasi citra. Prinsip dasar dari CNN terletak pada operasi konvolusi yang diterapkan, di mana setiap fitur dari sebuah gambar diekstraksi guna membentuk pola-pola tertentu yang lebih mudah dikenali dan diklasifikasikan. Pendekatan ini memungkinkan proses pembelajaran citra menjadi lebih efisien dan praktis untuk diimplementasikan (Arsal dkk, 2020).



Gambar 2. 2 Arsitektur *Convolutional Neural Network*

Dalam penerapannya, metode CNN dikembangkan melalui sejumlah tahapan, yakni mulai dari model *building*, model *training*, model *evaluation*, hingga final model *prediction*. Struktur CNN sendiri terdiri dari dua bagian utama, yaitu tahap ekstraksi fitur dan proses klasifikasi. Pada tahap ekstraksi fitur, CNN memiliki beberapa *hidden* layer yang meliputi *convolution* layer, *activation function*, dan *pooling* layer. Sistem ini bekerja secara hierarkis, di mana *output* dari *convolution* layer pertama akan menjadi *input* bagi lapisan berikutnya. Sementara untuk proses klasifikasi, CNN memanfaatkan *fully-connected* layer. Kedua komponen tersebut umumnya dilengkapi dengan fungsi aktivasi seperti *softmax* untuk mengoptimalkan performa selama proses pelatihan jaringan (Nugroho and Puspaningrum, 2021).

2.2.3 *You Only Look Once* (YOLO)

Salah satu algoritma yang banyak diterapkan dalam bidang *object detection* adalah YOLO (*You Only Look Once*), yang termasuk dalam kategori algoritma CNN dan dirancang dengan mekanisme kerja yang meniru cara otak manusia memproses informasi visual. YOLO, yang diperkenalkan oleh Josep Remond pada tahun 2015, merupakan bagian dari metode *deep learning*. Algoritma ini memanfaatkan jaringan neural yang membagi citra *input* menjadi sejumlah *grid*, yang kemudian digunakan untuk mengenali objek dengan mendeteksi *bounding box* beserta probabilitas klasifikasinya (Setiyadi dkk, 2023).

Dalam operasinya, metode YOLO melakukan pembagian gambar ke dalam beberapa *grid cell* yang saling tumpang tindih, di mana masing-masing *cell* bertugas memprediksi *bounding box* dan kelas objek yang ada di dalamnya. Dengan

pendekatan ini, YOLO dapat menjalankan proses deteksi secara paralel, sehingga meningkatkan kecepatan deteksi secara signifikan. Keunggulan utama dari YOLO terletak pada kemampuannya dalam mendeteksi objek dengan kecepatan tinggi tanpa kehilangan tingkat akurasi yang berarti. Algoritma ini mampu melakukan deteksi secara *real-time* dengan performa yang stabil, sehingga sangat ideal diterapkan pada berbagai sistem yang memerlukan respon cepat, seperti kendaraan otonom, sistem keamanan, maupun aplikasi pengenalan wajah (Nafis Alfarizi dkk, 2023).

2.2.4 Ultralytics dan YOLOv8

Ultralytics YOLOv8 adalah pengembangan terbaru dalam keluarga algoritma YOLO yang dirancang untuk menyempurnakan kecepatan dan akurasi dalam deteksi objek. Berbeda dengan versi sebelumnya, YOLOv8 menggunakan arsitektur yang lebih modular dan ringan sehingga mampu bekerja secara optimal pada berbagai jenis perangkat, termasuk *edge device* seperti *smartphone* maupun CCTV. Selain itu, YOLOv8 mendukung beberapa *task* sekaligus seperti *object detection, image segmentation,* dan *image classification* dalam satu *framework* terpadu (Ultralytics, 2023)

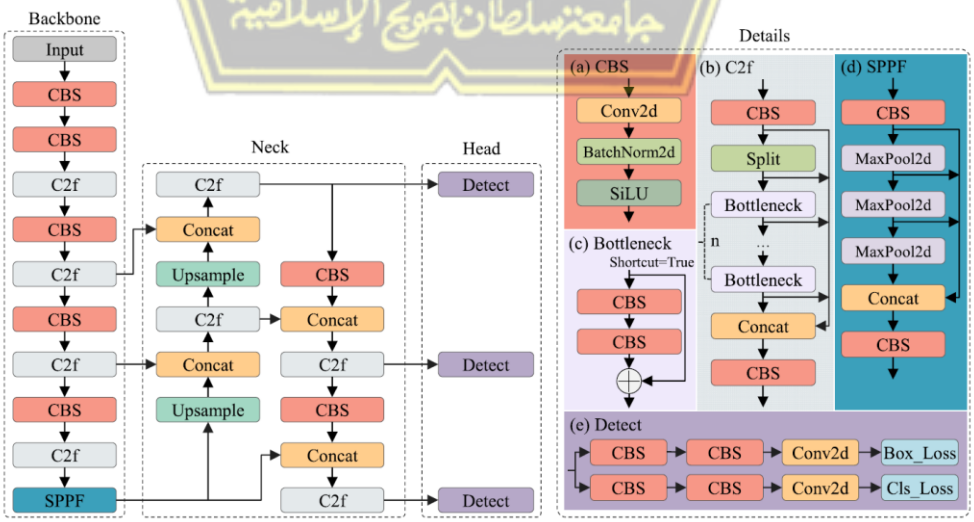
Dalam uji *benchmarking* yang dilakukan oleh tim peneliti, YOLOv8 menunjukkan peningkatan kinerja pada *dataset* COCO dan Pascal VOC, dengan mencapai skor *mean Average Precision* (mAP) yang lebih tinggi dibandingkan YOLOv7 dan EfficientDet. Arsitektur YOLOv8 memanfaatkan *anchor-free detection* yang membuat proses deteksi lebih cepat tanpa mengorbankan akurasi. Keunggulan lainnya, algoritma ini dapat diintegrasikan secara *real-time* pada aplikasi yang memerlukan kecepatan tinggi seperti kendaraan otonom, sistem pengawasan keamanan, dan robotika (Terven dkk, 2023).

Selain itu, YOLOv8 memperkenalkan pendekatan baru dalam proses *training* model dengan memanfaatkan data *augmentation* yang lebih adaptif serta penerapan *loss function* yang telah dioptimalkan untuk meningkatkan kestabilan selama pelatihan. Dengan arsitektur *anchor-free*, YOLOv8 berhasil mengurangi kompleksitas model serta menghilangkan kebutuhan untuk pengaturan *anchor*

boxes yang rumit, sehingga proses *deployment* menjadi lebih sederhana. Fitur auto-*learning bounding box* yang ada dalam YOLOv8 juga memungkinkan model untuk secara otomatis menyesuaikan prediksi posisi objek, yang berkontribusi terhadap meningkatnya nilai *recall* dalam pengujian.

Dari sisi efisiensi, YOLOv8 menunjukkan performa yang unggul dalam pemrosesan *real-time* dengan tetap menjaga konsumsi sumber daya komputasi yang rendah. Hal ini menjadikannya pilihan utama bagi pengembang aplikasi berbasis visi komputer yang memerlukan respons cepat tanpa infrastruktur berat, seperti sistem *monitoring* lalu lintas, aplikasi *mobile* untuk klasifikasi gambar, dan sistem pengenalan wajah. Dibandingkan dengan varian YOLO sebelumnya, YOLOv8 juga lebih fleksibel dalam hal integrasi dengan berbagai *framework deep learning* modern seperti PyTorch dan TensorFlow, sehingga dapat lebih mudah disesuaikan dengan kebutuhan spesifik penelitian atau aplikasi industri (Terven dkk, 2023).

Dengan menggunakan model *anchor-free* dengan kepala yang terpisah, YOLOv8 secara independen memproses tugas *objectness*, klasifikasi, dan regresi. Desain ini memungkinkan setiap cabang untuk fokus pada tugasnya sendiri, yang meningkatkan akurasi model secara keseluruhan. YOLOv8 menggunakan fungsi *softmax* untuk probabilitas kelas, yang menunjukkan kemungkinan objek termasuk dalam setiap kelas yang mungkin, dan fungsi *sigmoid* sebagai fungsi aktivasi untuk skor *objectness* pada lapisan *output*-nya.



Gambar 2. 3 Arstitektur YOLOv8(Shen dkk, 2023)

Backbone YOLOv8 terdiri dari tiga modul: CBS, C2f, dan SPPF. Modul CBS terdiri dari *Convolution* (Conv), *Batch Normalization* (BN), dan *Sigmoid* Linear Unit (SiLu). Modul C2f mengadopsi struktur *Cross Stage Partial* (CSP), di mana *output* dari modul CBS pertama dibagi menjadi dua cabang saluran. *Spatial Pyramid Pooling Fast* (SPPF) menggunakan tiga *pooling* layer dengan ukuran *kernel* yang sama (5 × 5) untuk mengurangi komputasi sekaligus meningkatkan efisiensi ekstraksi fitur. Melalui penggabungan fitur lokal dan global, SPPF meningkatkan bidang reseptif jaringan dan mencapai fitur *multiscale* melalui penggabungan, yang berkontribusi pada deteksi objek yang lebih baik.

Neck bertanggung jawab untuk penggabungan fitur dan mengadopsi struktur *Dual-Stream*, menggabungkan *Feature Pyramid Network* (FPN) dan *Path Aggregation Network* (PAN). Ini memadukan semantik dan informasi spasial dari *multi-scale feature maps*. Gambar akan melewati *deep networks*, bidang reseptif dan fitur semantik menguat, tetapi resolusinya menurun, sehingga menyulitkan untuk menangkap fitur objek kecil. FPN menggunakan pendekatan *top-down* untuk mengirimkan fitur semantik yang dalam ke *shallow layers* dan melakukan *tensor concatenation* dengan *feature maps* berukuran sama dari *backbone networks*, meningkatkan kinerja deteksi objek kecil. PAN menggunakan pendekatan *bottom-up* untuk mengirimkan warna, tepi, kontur, dan fitur spasial ke lapisan dalam sambil melakukan penggabungan *tensor* dengan peta fitur dari FPN. Hal ini menghasilkan penggabungan fitur *multiscale* yang komprehensif, mengkompensasi untuk kehilangan fitur yang disebabkan oleh pendalaman jaringan dan pengayaan informasi granularitas fitur.

Head menggunakan *Decoupled Head* untuk memisahkan pelokalan dan klasifikasi menjadi dua cabang. Ini memprediksi objek skala kecil, menengah, dan besar berdasarkan peta fitur P3, P4, dan peta fitur P5. YOLOv8 menggunakan *Anchor-Free Framework*, yang secara langsung memprediksi kotak pembatas di dalam *grid*, yang menghindari kebutuhan untuk menginisialisasi jangkar ukuran kotak dan *overhead* komputasi NMS, dengan demikian meningkatkan efisiensi pelokalan objek (Shen dkk., 2023).

2.2.5 *Bounding Box* dan *Intersection Over Union* (IoU)

Dalam *Object Detection, bounding box* adalah kotak yang menggambarkan lokasi objek dalam citra. Untuk menilai kualitas prediksi, digunakan metrik *Intersection over Union* (IoU), yang mengukur kesamaan antara *bounding box* yang diprediksi dengan yang sebenarnya. Nilai IoU dihitung dengan membandingkan luas irisan antara kedua kotak pembatas dan membaginya dengan luas gabungan kedua kotak tersebut. Semakin tinggi nilai IoU, semakin baik prediksi objek tersebut (Setiyadi, Utami and Ariatmanto, 2023).

2.2.6 Evaluasi Model - *Precision, Recall, dan mAP*

Untuk mengevaluasi performa model dalam deteksi objek, metrik seperti *Precision, Recall*, dan *mean Average Precision* (mAP) digunakan. *Precision* mengukur seberapa banyak prediksi positif yang benar, sedangkan *Recall* mengukur seberapa banyak objek yang benar-benar terdeteksi. mAP adalah rata-rata dari nilai *Average Precision* untuk setiap kelas objek. Metrik ini memberikan gambaran menyeluruh tentang akurasi dan kelengkapan deteksi objek (Nafis Alfarizi dkk., 2023)

2.2.7 Data *Augmentation*

Data *augmentation* adalah teknik yang digunakan untuk memperbesar *dataset* dengan melakukan transformasi pada citra, seperti rotasi, *flipping*, dan perubahan kecerahan. Teknik ini membantu mengurangi *overfitting* dengan menciptakan variasi data, serta meningkatkan kemampuan model dalam mendeteksi objek dalam berbagai kondisi (Arsal, Agus Wardijono and Anggraini, 2020).

2.2.8 Roboflow

Roboflow adalah platform yang menyediakan alat bantu dalam pengolahan data untuk pengembangan model deteksi objek, yang menawarkan berbagai fitur seperti anotasi data, augmentasi, serta pemrosesan *dataset*. Platform ini memfasilitasi pengembangan model *deep learning* dengan memberikan lebih dari 100.000 *dataset* yang telah dikurasi, dan memungkinkan para pengguna untuk memilih, membersihkan, dan mengonversi *dataset* ke dalam format yang

diperlukan untuk berbagai jenis model, seperti YOLO, SSD, dan EfficientDet. Roboflow mengintegrasikan teknologi auto-*labeling* berbasis model *pretrained* yang mempermudah pengguna dalam melakukan anotasi data secara otomatis, sehingga mengurangi waktu dan biaya yang diperlukan untuk menyiapkan dataset besar. Dengan menggunakan Roboflow, proses augmentasi gambar yang melibatkan teknik seperti rotasi, *flipping*, perubahan skala, dan perubahan kecerahan dapat dilakukan secara otomatis, yang pada gilirannya meningkatkan performa dan akurasi model deteksi objek (Kalhor and Bahrak, 2023).

Selain menyediakan *dataset*, Roboflow juga menawarkan alat yang memungkinkan pengguna untuk memantau dan mengevaluasi kualitas data mereka sepanjang siklus pelatihan model. Platform ini mendukung berbagai teknik evaluasi untuk memastikan bahwa data yang digunakan dalam pelatihan memiliki kualitas yang tinggi dan relevansi yang tepat untuk tugas deteksi objek yang dimaksud. Roboflow memungkinkan pengguna untuk memvisualisasikan dan menganalisis distribusi data, melihat jumlah gambar per kelas, dan memeriksa label yang mungkin salah atau kurang tepat (Kalhor and Bahrak, 2023).

BAB III
METODOLOGI PENELITIAN

3.1 Deskripsi Sistem

Aplikasi yang dikembangkan merupakan sistem deteksi objek rokok secara real-time dengan memanfaatkan algoritma *You Only Look Once* versi 8 (YOLOv8), serta diintegrasikan dengan antarmuka pengguna berbasis Streamlit. Sistem ini dirancang untuk menerima *input* langsung dari kamera (*webcam* atau IP cam), yang kemudian diproses dalam bentuk *frame* per *frame*.

Setiap *frame* yang diterima akan melalui tahapan awal seperti pengubahan ukuran dan normalisasi gambar, agar sesuai dengan format *input* yang dibutuhkan oleh model deteksi. Selanjutnya, proses inferensi dijalankan oleh YOLOv8 untuk mengidentifikasi objek rokok yang muncul pada *frame* tersebut.

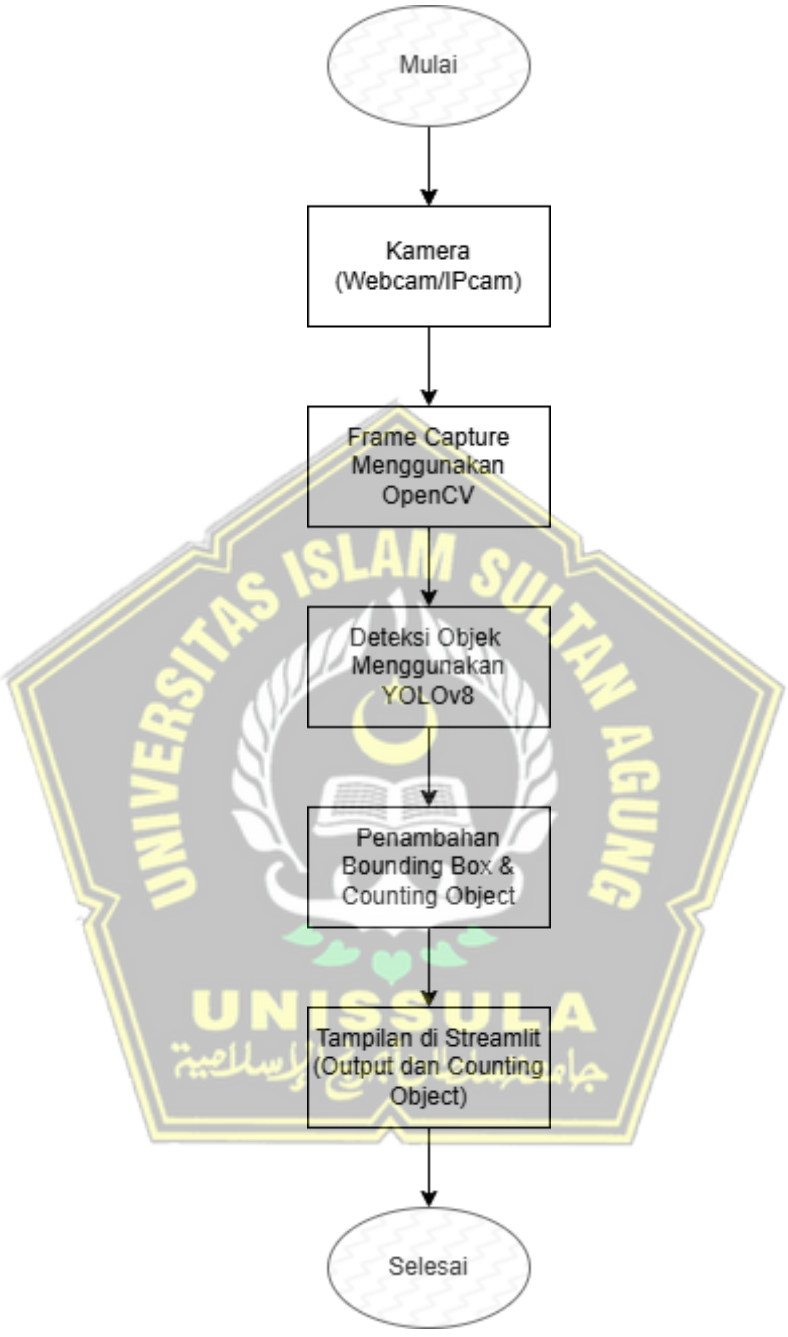
Objek yang terdeteksi akan ditampilkan dalam bentuk visual berupa *bounding box*, yaitu kotak yang mengelilingi objek dengan tambahan informasi tingkat kepercayaan (*confidence score*). Sistem ini juga melakukan perhitungan jumlah total objek rokok yang terdeteksi pada setiap frame (*object counting*), dan hasilnya divisualisasikan secara langsung melalui antarmuka Streamlit dalam tampilan web.

3.2 Studi Literatur

Penelitian ini diawali dengan kajian terhadap berbagai sumber literatur, baik dari jurnal, skripsi, maupun referensi daring. Tujuannya adalah untuk memahami dan mempelajari teori serta pendekatan teknis yang mendasari metode deteksi objek menggunakan algoritma YOLOv8.

3.3 Diagram Arsitektur Sistem

Dalam penelitian ini, penulis akan merancang dan membangun sebuah sistem deteksi rokok secara *real-time* dengan memanfaatkan algoritma YOLOv8 sebagai inti dari proses deteksi objek. Sistem ini dirancang menggunakan arsitektur yang sederhana namun efektif, di mana setiap komponen memiliki peran spesifik dalam mendukung keseluruhan proses deteksi.



Gambar 3. 1 Diagram Arsitektur Sistem

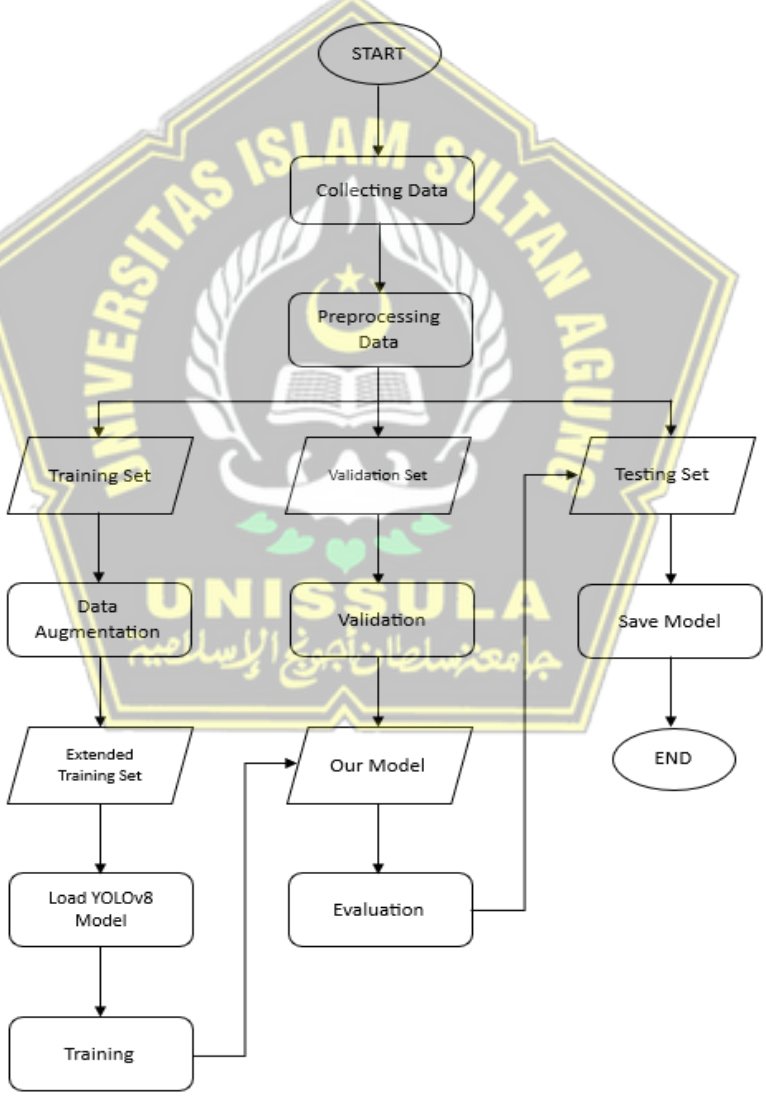
Sistem akan diawali dengan penggunaan kamera sebagai alat input utama untuk menangkap citra atau video secara langsung. Gambar yang diperoleh kemudian akan dikirimkan ke model deteksi objek YOLOv8 yang telah dilatih untuk mengenali objek rokok. Setiap *frame* yang masuk akan dianalisis oleh model, dan apabila objek rokok terdeteksi, sistem akan memberikan penandaan berupa

kotak pembatas (*bounding box*) pada area yang sesuai. Selain memberikan penandaan visual, sistem juga akan menghitung jumlah objek rokok yang berhasil dideteksi dalam setiap *frame* (*object counting*). Informasi ini akan ditampilkan secara *real-time* melalui antarmuka pengguna berbasis Streamlit.

3.4 Perancangan Model

3.5.1 Alur Perancangan Model

Pada tahap perancangan alur model, peneliti akan membuat alur model dalam bentuk *flowchart* atau diagram alir sebagai berikut :



Gambar 3. 2 Alur Perancangan Model

Pada tahap perancangan alur model terdapat beberapa tahapan yang dimulai dari *collecting* data atau pencarian dan pengumpulan data yang berupa gambar manusia yang sedang melakukan aktivitas merokok. Kemudian pada tahap *preprocessing* data terdiri dari *cropping*, *resize* gambar, augmentasi data, *labelling* data, anotasi data dan *split* data. Pada *split* data, data dibagi menjadi tiga, yaitu *training* set, *validation* set dan *testing* set. Pada *training* set dilakukan augmentasi data dan menghasilkan *extended training* data yang kemudian akan dimuat pada YOLOv8 model untuk dilakukan tahapan *training*. Setelah menghasilkan *output* model, kemudian dilakukan *testing* model pada tahapan evaluasi. Pada tahapan evaluasi terdapat *testing* model, visualisasi hasil *testing* yang menampilkan *precision, recall* dan visualisasi *confusion matrix*.

3.5.2 *Collecting* Data

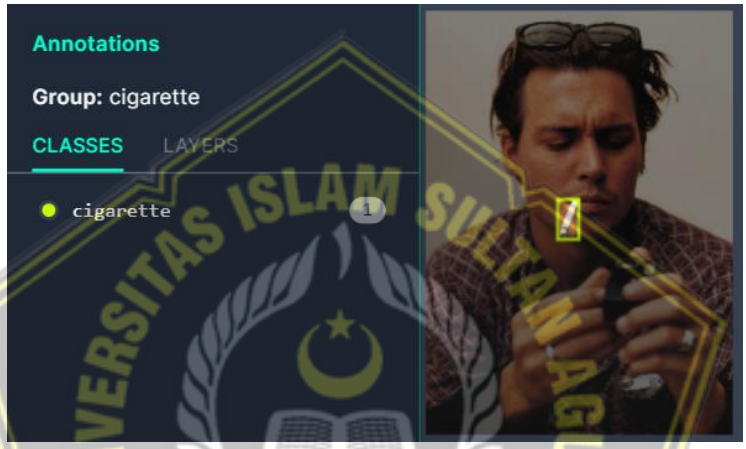
Pada penelitian ini, proses pengumpulan data yang berupa gambar diperoleh dari Roboflow (https://universe.roboflow.com/phonedt/cigarette-bkjaq). Sehingga jumlah gambar yang akan digunakan pada penelitian ini berjumlah 3029 gambar.

Gambar 3. 3 *Dataset*

Gambar di atas merupakan visualisasi *sample* dari *dataset* yang diperoleh peneliti.

3.5.3 Anotasi Data

Anotasi merupakan proses penandaan dengan menetapkan kotak pembatas (*bounding box*) yang memuat nama kelas sesuai dengan setiap objek (Akbar, 2021). Kotak pembatas tersebut diberikan label yang merepresentasikan jenis objek yang ada di dalamnya. Tahap ini sangat krusial dalam pelatihan model deteksi objek karena memberikan data mengenai posisi dan kategori objek dalam sebuah gambar kepada model.

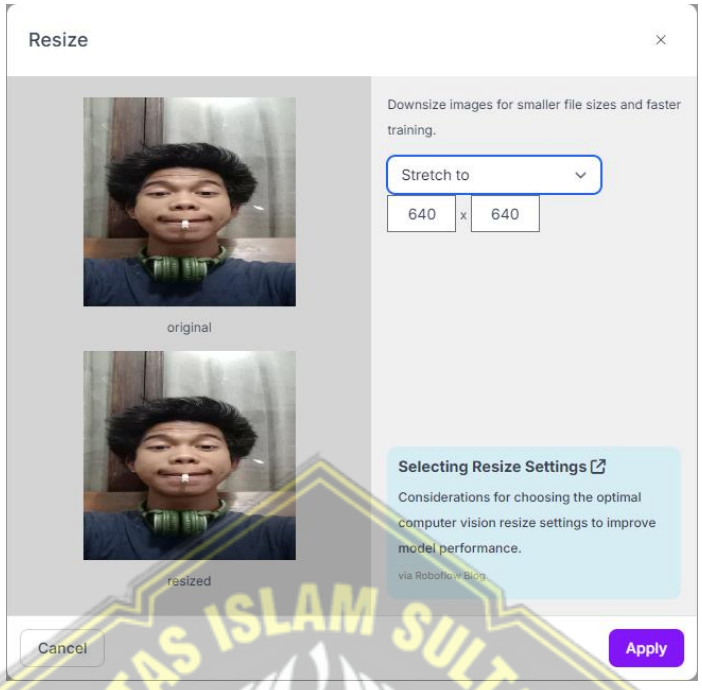


Gambar 3. 4 Anotasi Data

Pada gambar di atas merupakan proses anotasi data gambar yang dilakukan menggunakan Roboflow *Annotate* yang digunakan untuk membuat *bounding box*.

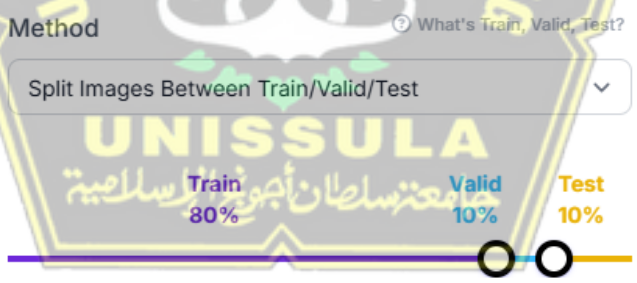
3.5.4 *Preprocessing* Data

Tahap awal dalam *preprocessing* gambar adalah melakukan *resize*. Proses ini bertujuan untuk menyeragamkan resolusi seluruh gambar dalam *dataset*, yang awalnya memiliki resolusi berbeda-beda, sehingga diubah menjadi ukuran yang konsisten pada setiap gambar. Selain itu, *resize* juga membantu mengurangi ukuran *dataset* secara keseluruhan, yang berdampak pada waktu yang diperlukan saat proses pelatihan model..



Gambar 3. 5 *Preprocessing* Data

Lalu *labelling* data yaitu proses pelabelan data agar dapat digunakan pada pelatihan model.

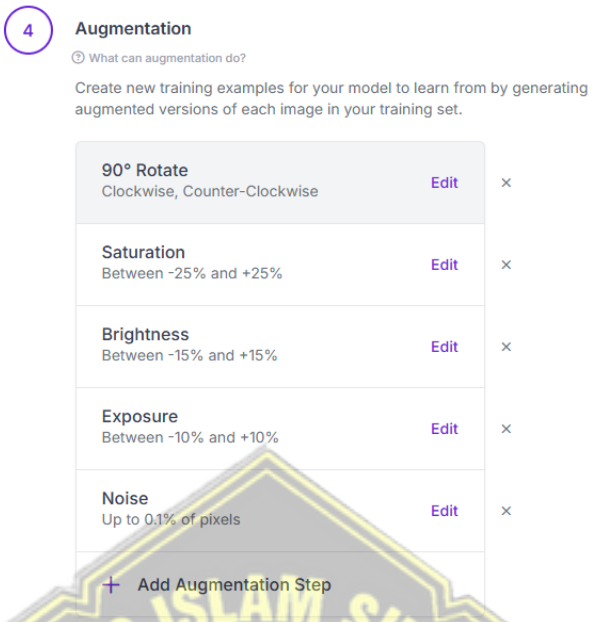


Gambar 3. 6 Split Data

Kemudian dilakukan *split* data yang merupakan proses pembagian data yang akan dibagi menjadi tiga yaitu data *training*, data testing serta data *validation*.

3.5.5 Augmentasi Data

Pada penelitian ini terdapat beberapa augmentasi data yang dilakukan pada *dataset* yaitu rotasi gambar, mengatur kecerahan gambar, mengatur *exposure* gambar, *mosaic* gambar dan menambahkan efek *noise* pada gambar.



Gambar 3. 7 Augmentasi Data

Rotasi gambar dilakukan untuk membantu model dalam pendeteksian ketika objek mengalami perubahan orientasi. Mengatur kecerahan dan *exposure* pada gambar dilakukan untuk membantu model dalam mendeteksi objek ketika terjadi perubahan pencahayaan. Menambahkan efek *noise* pada gambar dilakukan untuk membantu model dalam mendeteksi objek dengan kondisi kamera yang digunakan dalam pendeteksian mengalami *noise*. *Mosaic* gambar pada *dataset* dilakukan untuk membantu performa model dalam mendeteksi objek dalam ukuran kecil atau jauh.

3.5.6 Training

Pada tahap ini, peneliti akan melatih model deteksi objek dengan memanfaatkan algoritma YOLOv8. *Dataset* yang digunakan akan diambil dari platform Roboflow yang telah dianotasi sebelumnya. Data yang diperoleh akan dikonversi ke format yang sesuai untuk pelatihan model YOLO, yaitu terdiri dari *file* konfigurasi *.yaml* dan *file* label dalam format YOLO.

Dalam pelatihan model deteksi objek rokok menggunakan YOLOv8, terdapat beberapa tahapan penting yang berkaitan dengan struktur internal model. Setiap komponen arsitektur memiliki peran tersendiri dalam mengekstrak, memperkaya,

dan mengklasifikasikan informasi visual dari dataset citra rokok. Berikut adalah penjabaran dari masing-masing blok utama:

1. Convolution-BatchNorm-SiLU (CBS)

Blok awal pada YOLOv8 adalah CBS, yang terdiri dari tiga lapisan utama:

a. Lapisan Konvolusi (Convolutional Layer)

Berfungsi untuk mengenali fitur dasar gambar, seperti: tepi dari batang rokok, struktur silinder rokok, variasi intensitas cahaya dan bayangan, serta bagian ujung yang menyala atau filter. Citra input (misalnya 640×640×3) diproses oleh sejumlah filter (misalnya 32 kernel berukuran 3×3) yang melakukan konvolusi pada gambar. Proses ini menghasilkan feature map yang menyorot bagian-bagian penting. Contohnya, satu filter mungkin sensitif terhadap garis horizontal yang mencerminkan rokok yang dilihat dari samping.

b. Batch Normalization (BN)

Digunakan untuk menstabilkan proses pelatihan dengan mengatur distribusi output dari layer sebelumnya agar memiliki rata-rata nol dan deviasi standar satu. Hal ini mempercepat konvergensi dan menjaga kestabilan pelatihan.

c. Aktivasi SiLU

Fungsi aktivasi SiLU (Sigmoid Linear Unit) digunakan untuk memperkenalkan non-linearitas dan mempertajam fitur penting yang dihasilkan.

2. C2f (Concat-to-Fusion)

C2f adalah versi yang lebih efisien dari blok C3 yang digunakan pada YOLOv5/YOLOv7. Blok ini menyatukan dan mengolah fitur secara lebih dalam tanpa kehilangan informasi spasial. Struktur utama C2f: CBS awal, split channel, beberapa bottleneck blocks, concatenation, dan CBS akhir.

Alur kerjanya: fitur dari layer sebelumnya diteruskan ke C2f, dibagi menjadi beberapa saluran (channel), lalu diproses oleh bottleneck block. Masing-masing bottleneck memproses fitur secara mendalam, kemudian hasilnya digabung (concatenate), dan jika perlu, diproses ulang oleh CBS akhir.

3. SPPF (Spatial Pyramid Pooling – Fast)

Blok SPPF bertugas memperluas area cakupan fitur (receptive field) tanpa menambah dimensi data secara signifikan. Ini dilakukan dengan beberapa operasi max pooling menggunakan kernel ukuran tetap, lalu hasilnya digabung dan dikonvolusi. SPPF sangat berguna dalam menangkap objek dengan skala dan posisi bervariasi, seperti rokok yang tidak selalu berada di tengah gambar atau berukuran kecil.

4. Detect Layer

Bagian ini menghasilkan output akhir berupa: koordinat bounding box (x, y, lebar, tinggi), skor keyakinan objek (objectness score), dan kelas objek (rokok). Prediksi dilakukan pada beberapa skala untuk menjangkau objek dengan berbagai ukuran. Setelah itu, algoritma Non-Maximum Suppression (NMS) digunakan untuk menghilangkan prediksi yang tumpang tindih.

Semua komponen di atas bekerja secara end-to-end dalam pipeline pelatihan. Dataset gambar rokok dari Roboflow dimasukkan ke dalam pipeline YOLOv8, kemudian melalui tahapan: ekstraksi fitur awal (CBS), pengolahan dan penggabungan fitur (C2f), penyesuaian spasial (SPPF), dan prediksi akhir (Detect Layer).

Fungsi loss pada YOLOv8 menghitung selisih antara prediksi dan data ground truth, dan bobot jaringan diperbarui melalui proses backpropagation. Dengan pendekatan ini, model mampu membedakan dan mengenali objek rokok secara akurat, bahkan dalam kondisi visual yang kompleks.

Secara keseluruhan, arsitektur YOLOv8 dirancang untuk mendukung proses deteksi objek secara efisien dan akurat, menjadikannya pilihan yang tepat untuk sistem deteksi rokok otomatis.

Proses pelatihan direncanakan berjalan secara lokal menggunakan GPU, agar kecepatan komputasi lebih optimal. Adapun konfigurasi parameter pelatihan yang digunakan dalam proses ini dijelaskan sebagai berikut:

- *Number of Epochs* (epochs)

*Epoch*s merupakan jumlah siklus penuh saat seluruh data pelatihan diproses oleh model. Satu *epoch* berarti seluruh *dataset* telah digunakan satu kali dalam

proses pelatihan. Jumlah *epoch* yang lebih tinggi memungkinkan model belajar lebih dalam, namun dapat meningkatkan risiko *overfitting*.

- *Image Size* (imgsz)

Citra yang dimasukkan ke dalam model biasanya disesuaikan ke dalam ukuran tertentu agar konsisten. Penyesuaian ukuran ini bertujuan untuk mengurangi beban komputasi sekaligus memastikan *input* kompatibel dengan arsitektur model.

- *Batch Size* (batch)

Batch size adalah jumlah data yang diproses secara bersamaan dalam satu iterasi pelatihan. Pemilihan *batch size* memengaruhi performa komputasi, *batch* kecil cenderung lebih stabil namun lambat, sedangkan *batch* besar lebih cepat namun membutuhkan memori lebih banyak.

- *Initial Learning Rate* (lr0)

Nilai awal dari *learning rate*, yaitu besarnya langkah pembelajaran model dalam meng-*update* bobot. Nilai yang kecil seperti ini dipilih agar proses pelatihan berlangsung secara bertahap dan stabil.

- Final *Learning Rate* (lrf)

Menyatakan rasio akhir dari *learning rate* terhadap nilai awal. Dengan menurunkan nilai *learning rate* secara bertahap, model dapat lebih mudah menyempurnakan bobot di tahap akhir pelatihan.

- *Optimizer Type* (optimizer)

Optimizer adalah algoritma yang digunakan untuk memperbarui bobot model berdasarkan hasil evaluasi terhadap data pelatihan. Jenis *optimizer* yang umum digunakan antara lain SGD (*Stochastic Gradient Descent*), Adam dan *auto*.

- *Early Stopping* (patience)

Digunakan sebagai parameter *early stopping*. Jika performa model tidak mengalami peningkatan selama 50 *epoch* berturut-turut, maka proses pelatihan akan dihentikan lebih awal untuk menghindari pemborosan sumber daya.

- *Dropout Rate* (dropout)

Dropout merupakan metode regularisasi yang digunakan untuk mencegah *overfitting*. Teknik ini bekerja dengan cara mengabaikan (drop) sebagian neuron

secara acak selama pelatihan, sehingga model tidak terlalu bergantung pada fitur tertentu.

- *Device Selection* CPU/GPU (*device*)

Menandakan bahwa pelatihan dilakukan menggunakan GPU dengan indeks 1, yang memungkinkan proses berjalan lebih cepat dibandingkan menggunakan CPU.

3.5.7 Perbandingan Parameter *Optimizer* pada Pelatihan Model YOLOv8

Pada tahap pelatihan model, penulis akan menggunakan beberapa konfigurasi parameter yang telah disesuaikan agar proses pelatihan dapat berjalan secara optimal. Model yang digunakan dalam penelitian ini adalah YOLOv8n, yaitu varian ringan dari YOLOv8 yang cocok untuk kebutuhan real-time dengan keterbatasan perangkat keras.

Parameter-parameter pelatihan yang ditetapkan meliputi jumlah epoch, ukuran *batch*, nilai *learning rate* awal (lr0), rasio *learning rate* final (lrf), hingga pemilihan jenis *optimizer*. Adapun konfigurasi lengkap dapat dilihat pada tabel berikut:

Tabel 3. 1 Perbandingan Parameter *Optimizer*

No.	Model	epochs	patience	batch	lr0	lrf	optimizer
1	yolov8n	100	50	16	0.0001	0.1	SGD
2	yolov8n	100	50	16	0.0001	0.1	Adam
3	yolov8n	100	50	16	0.0001	0.1	auto

Pada tahap ini, penulis akan melakukan eksperimen dengan beberapa jenis *optimizer* yang berbeda, yaitu SGD, Adam, dan auto, guna mengetahui pengaruh masing-masing terhadap performa pelatihan model. Dengan pengaturan parameter yang konsisten di setiap percobaan, maka perbandingan antar *optimizer* dapat dilakukan secara adil dan terukur.

3.5.8 Evaluasi

Dalam deteksi objek, *Mean Average Precision* (mAP) digunakan sebagai metode evaluasi. YOLOv8 secara bawaan memakai metrik mAP50 dan mAP50-95. Matriks kebingungan (*confusion matrix*) digunakan untuk menghitung nilai *Precision* dan *Recall* pada setiap label dalam gambar sebagai bagian dari perhitungan mAP. Metrik mAP50 menggunakan ambang batas *Intersection over Union* (IoU) sebesar 0,5, sedangkan mAP50-95 merupakan rata-rata mAP yang dihitung dari sepuluh nilai IoU berbeda, yaitu mulai dari 0,5 hingga 0,95 dengan kenaikan 0,05. Untuk menentukan nilai *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN), matriks kebingungan pada deteksi objek memanfaatkan nilai IoU. Sedangkan *True Negative* (TN) tidak digunakan karena tidak memberikan pengaruh signifikan terhadap performa metode deteksi objek..

BAB IV
HASIL DAN ANALISIS PENELITIAN

4.1 Hasil dan Analisis

Merujuk pada tabel 3.1 yang telah dilakukan perbandingan parameter *optimizer* pada pelatihan model YOLOv8n dengan hasil dan analisis sebagai berikut.

4.1.1 Proses Pelatihan Model

Proses pelatihan model dilakukan menggunakan arsitektur YOLOv8n dengan bantuan pustaka Ultralytics. *Dataset* yang digunakan telah diambil dari Roboflow dan dikonversi ke dalam format yang sesuai dengan kebutuhan pelatihan. Seluruh pelatihan dilakukan dalam lingkungan Python menggunakan GPU untuk mempercepat proses komputasi.

Berikut merupakan konfigurasi pelatihan yang digunakan:

```
results   = model.train(
            data=yaml_file_path,          # Path to the dataset configuration file
            epochs=100,                   # Number of epochs to train for
            imgsz=640,                    # Size of input images as integer
            patience=50,                  # Early stopping of training
            batch=16,                     # Number of images per batch
            optimizer='auto',             # Optimizer to use
            lr0=0.0001,                   # Initial learning rate
            lrf=0.1,                      # Final learning rate (lr0 * lrf)
            dropout=0.1,                  # Use dropout regularization
            seed=0,                       # Random seed for reproducibility
            device=1                      # Use GPU
)
```

Konfigurasi tersebut menetapkan jumlah *epoch* sebanyak 100, dengan ukuran gambar (*image size*) sebesar 640 *pixel*. Pengaturan *batch size* sebesar 16 digunakan untuk mengontrol jumlah data yang diproses dalam satu iterasi pelatihan. *Learning rate* awal ditentukan sebesar 0.0001, dan akan diturunkan secara bertahap hingga 10% dari nilai awal sesuai dengan nilai lrf (*learning rate factor*). Selain itu, *dropout* digunakan sebesar 10% untuk mencegah *overfitting*, dan pelatihan dilakukan pada perangkat GPU untuk meningkatkan efisiensi proses.

4.1.2 Perbandingan Penggunaan *Optimizer*

Untuk mengevaluasi pengaruh jenis algoritma optimasi terhadap performa pelatihan, dilakukan eksperimen menggunakan tiga jenis *optimizer* yang berbeda,

yaitu auto, Adam, dan SGD. Tujuannya adalah untuk mengetahui sejauh mana masing-masing *optimizer* mempengaruhi akurasi dan kecepatan konvergensi model.

1. Auto

Pemilihan *optimizer* secara otomatis (auto) memungkinkan sistem menentukan metode optimasi terbaik berdasarkan konfigurasi dan karakteristik data yang digunakan.

```
results   = model.train(
            data=yaml_file_path,          # Path to the dataset configuration file
            epochs=100,                   # Number of epochs to train for
            imgsz=640,                    # Size of input images as integer
            patience=50,                  # Early stopping of training
            batch=16,                     # Number of images per batch
            optimizer='auto',             # Optimizer to use
            lr0=0.0001,                   # Initial learning rate
            lrf=0.1,                      # Final learning rate (lr0 * lrf)
            dropout=0.1,                  # Use dropout regularization
            seed=0,                       # Random seed for reproducibility
            device=1                      # Use GPU
)
```

Mode ini memudahkan pengguna karena tidak perlu menentukan jenis *optimizer* secara manual. Hasil pelatihan dengan auto menunjukkan konvergensi yang stabil dan waktu pelatihan yang efisien.

2. Adam

Optimizer Adam (*Adaptive Moment Estimation*) merupakan salah satu metode populer dalam pelatihan model *deep learning* karena mampu menyesuaikan laju pembelajaran (*learning rate*) secara dinamis.

```
results   = model.train(
            data=yaml_file_path,          # Path to the dataset configuration file
            epochs=100,                   # Number of epochs to train for
            imgsz=640,                    # Size of input images as integer
            patience=50,                  # Early stopping of training
            batch=16,                     # Number of images per batch
            optimizer='auto',             # Optimizer to use
            lr0=0.0001,                   # Initial learning rate
            lrf=0.1,                      # Final learning rate (lr0 * lrf)
            dropout=0.1,                  # Use dropout regularization
            seed=0,                       # Random seed for reproducibility
            device=1                      # Use GPU
)
```

Adam menggabungkan keuntungan dari RMSProp dan momentum, sehingga cocok digunakan untuk *dataset* yang kompleks. Penggunaan Adam pada

eksperimen ini menghasilkan konvergensi yang cepat dan performa yang cukup baik.

3. SGD

Stochastic Gradient Descent (SGD) adalah metode optimasi yang paling sederhana namun tetap efektif dalam banyak kasus.

```
results   = model.train(
            data=yaml_file_path,          # Path to the dataset configuration file
            epochs=100,                   # Number of epochs to train for
            imgsz=640,                    # Size of input images as integer
            patience=50,                  # Early stopping of training
            batch=16,                     # Number of images per batch
            optimizer='auto',             # Optimizer to use
            lr0=0.0001,                   # Initial learning rate
            lrf=0.1,                      # Final learning rate (lr0 * lrf)
            dropout=0.1,                  # Use dropout regularization
            seed=0,                       # Random seed for reproducibility
            device=1                      # Use GPU
)
```

Meskipun konvergensinya cenderung lebih lambat dibandingkan Adam, penggunaan SGD dengan *learning rate* yang tepat dapat menghasilkan model yang stabil. Namun demikian, dalam eksperimen ini, waktu pelatihan cenderung lebih lama dengan hasil yang relatif serupa terhadap *optimizer* lain.

4.1.3 Hasil Evaluasi Model

Eksperimen dilakukan dengan menggunakan tiga jenis optimizer, yaitu Auto, Adam, dan SGD, untuk mengetahui pengaruhnya terhadap performa model YOLOv8 dalam mendeteksi objek rokok secara *real-time*. Seluruh proses pelatihan dilakukan dengan parameter yang sama, di antaranya: epoch sebanyak 100, ukuran gambar 640 *pixel*, *batch size* 16, dan *learning rate* awal sebesar 0.0001.

Tabel 4. 1 Hasil Evaluasi Model

Optimizer	Precision	Recall	mAP50	mAP50-95	Fitness
Auto	0.878	0.795	0.863	0.468	0.507
Adam	0.886	0.761	0.830	0.450	0.488
SGD	0.736	0.654	0.712	0.355	0.391

Dari tabel tersebut, terlihat bahwa *optimizer* Auto memberikan hasil terbaik secara umum, terutama pada metrik mAP50 (0.863) dan *fitness* (0.507). Sementara

itu, Adam memberikan *precision* tertinggi (0.886), meskipun mAP50 sedikit lebih rendah dibanding Auto. *Optimizer* SGD menghasilkan nilai terendah pada semua metrik, yang menunjukkan bahwa *optimizer* ini kurang efektif tanpa penyetelan lanjutan.

4.1.4 Visualisasi Hasil Evaluasi

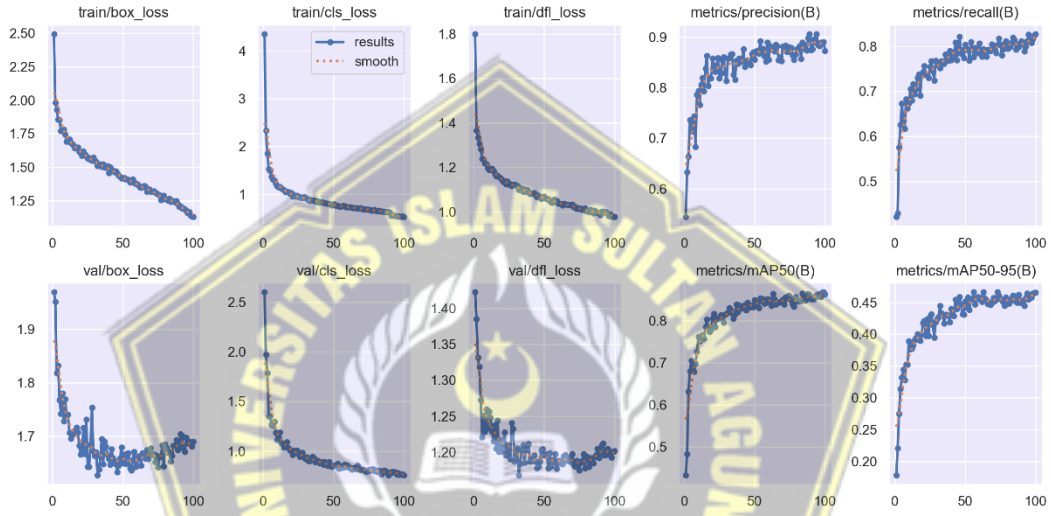
Grafik dan *confusion matrix* di bawah ini menggambarkan perbandingan metrik performa model berdasarkan jenis *optimizer* yang digunakan:

1. *Optimizer* = "Auto"

Pada proses pelatihan model YOLOv8 dengan parameter *optimizer* menggunakan opsi auto, sistem secara otomatis memilih algoritma optimasi yang dianggap paling sesuai berdasarkan konfigurasi dan data yang digunakan. Secara internal, Auto *optimizer* yang digunakan oleh Ultralytics secara otomatis memilih *optimizer* AdamW sebagai strategi optimasi. Hal ini menunjukkan bahwa AdamW dianggap paling cocok untuk karakteristik *dataset* yang digunakan dalam penelitian ini. Hasil evaluasi terhadap kinerja model ditampilkan melalui *confusion matrix* dan grafik perkembangan metrik selama proses pelatihan.

Gambar 4. 1 *Confusion Matrix Optimizer* Auto

Gambar *confusion matrix* menunjukkan bahwa model berhasil mengklasifikasikan objek rokok sebanyak 529 data secara benar sebagai kelas "*cigarette*" (*True Positive*). Namun, terdapat pula 64 data *background* yang salah diklasifikasikan sebagai rokok (*False Positive*), serta 86 data rokok yang tidak terdeteksi dan dianggap sebagai *background* (*False Negative*). Hasil ini mengindikasikan bahwa model memiliki tingkat ketelitian (*precision*) dan cakupan (*recall*) yang relatif baik.

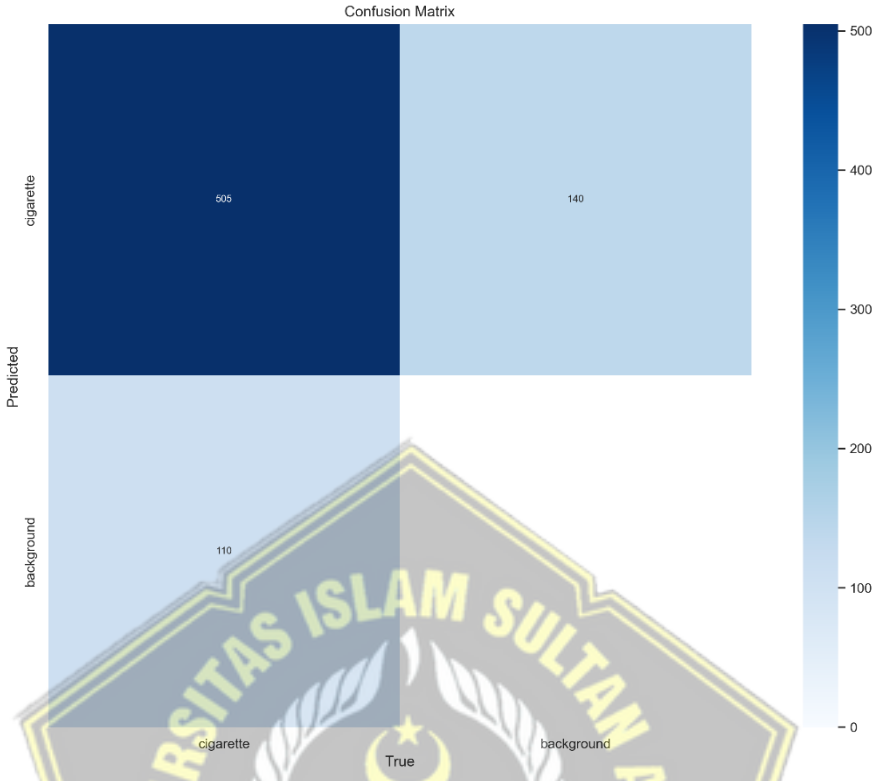


Gambar 4. 2 Grafik Hasil Pelatihan *Optimizer* Auto

Grafik hasil pelatihan memperlihatkan tren yang stabil dan menunjukkan adanya konvergensi model seiring bertambahnya *epoch*. Penurunan nilai *loss* pada metrik *box_loss, cls_loss*, dan *dfl_loss* menandakan bahwa model semakin mampu mengenali objek secara akurat. Selain itu, metrik evaluasi seperti *precision*, *recall*, mAP50, dan mAP50-95 menunjukkan peningkatan yang konsisten dari awal hingga akhir pelatihan. Nilai akhir dari mAP50 yang diperoleh adalah 0.863, sedangkan mAP50-95 sebesar 0.468.

2. *Optimizer* = "Adam"

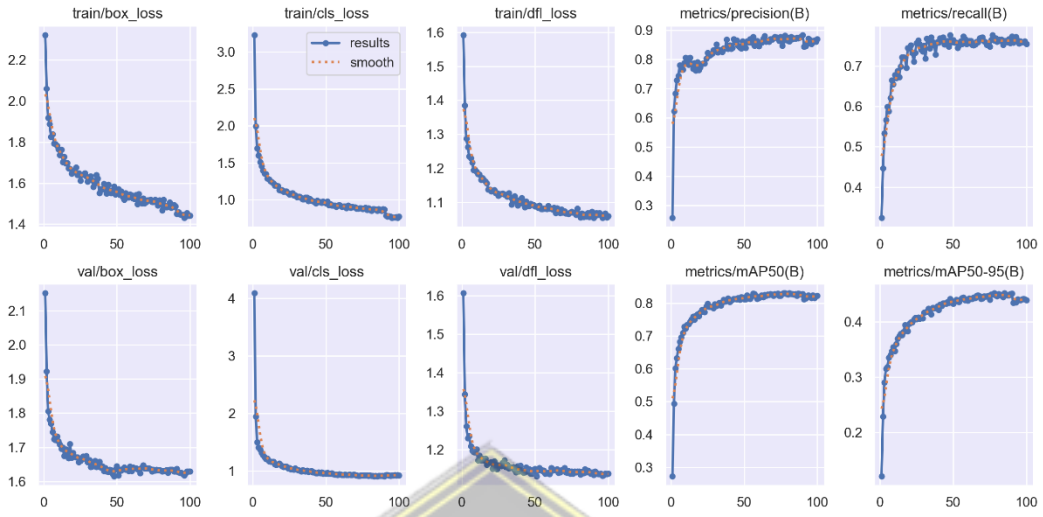
Berdasarkan hasil evaluasi menggunakan *confusion matrix*, model YOLOv8 yang dilatih dengan *optimizer* Adam menunjukkan performa yang cukup baik dalam mendeteksi objek rokok.



Gambar 4. 3 *Confusion Matrix Optimizer* Adam

Confusion matrix memperlihatkan bahwa model berhasil memprediksi objek rokok secara benar sebanyak 505 kali (*true positive*). Namun, terdapat juga prediksi yang salah, yaitu sebanyak 140 kali model memprediksi latar belakang sebagai rokok (*false positive*), dan sebanyak 110 kali model gagal mendeteksi keberadaan rokok padahal sebenarnya ada (*false negative*).

Dari hasil ini dapat disimpulkan bahwa model memiliki tingkat akurasi yang tinggi dalam mendeteksi objek rokok, meskipun masih terdapat kesalahan dalam klasifikasi latar belakang. Nilai *true positive* yang tinggi menunjukkan bahwa model memiliki sensitivitas yang baik terhadap objek target, sedangkan nilai *false positive* dan *false negative* yang masih ada menunjukkan bahwa model belum sepenuhnya optimal dalam membedakan antara objek rokok dan latar belakang.



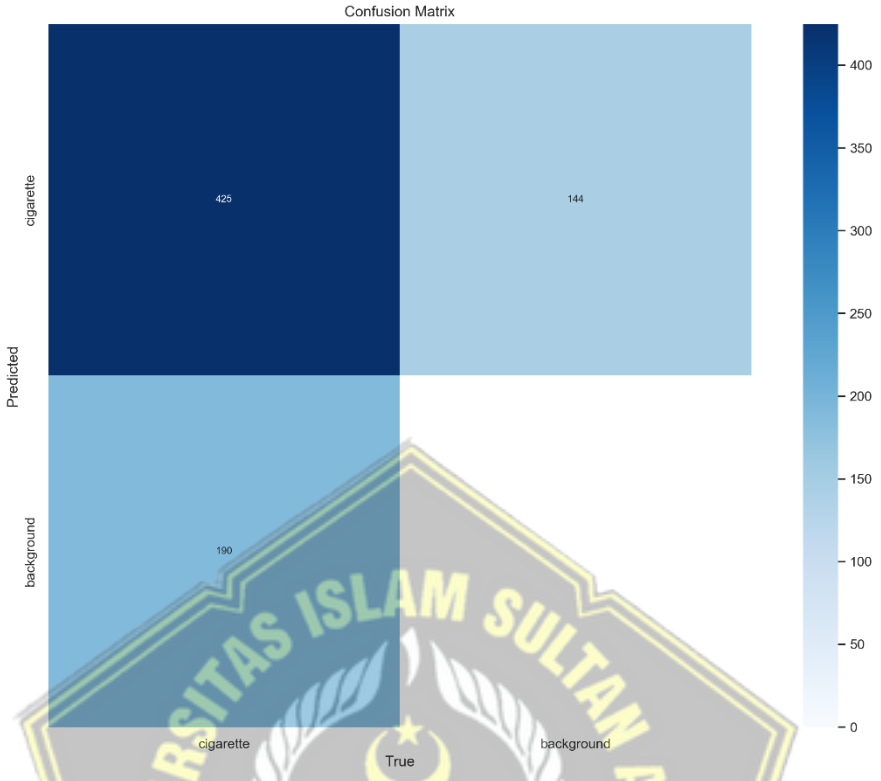
Gambar 4. 4 Grafik Hasil Pelatihan *Optimizer* Adam

Selama proses pelatihan, model menunjukkan peningkatan performa yang signifikan dari segi *loss function* dan *evaluation metrics*. Grafik pelatihan memperlihatkan bahwa nilai *box loss, classification loss*, dan *distribution focal loss* (DFL) pada data pelatihan dan validasi terus menurun secara konsisten hingga mencapai titik konvergen. Penurunan *loss* ini mencerminkan bahwa model berhasil belajar dari data dengan baik dan tidak mengalami *overfitting*.

Selain itu, grafik metrik evaluasi seperti *precision, recall*, dan *mean Average Precision* (mAP) juga menunjukkan tren positif. Nilai *precision* mencapai sekitar 0.87, yang menunjukkan bahwa sebagian besar prediksi objek rokok yang dilakukan oleh model adalah benar. Sementara itu, *recall* mencapai nilai sekitar 0.76, yang berarti model mampu mendeteksi sebagian besar objek rokok yang ada dalam gambar. Nilai mAP50 mencapai sekitar 0.84, sedangkan mAP50-95 berada pada angka 0.45, yang menandakan performa deteksi yang stabil dalam berbagai skenario dan tingkat kesulitan.

3. *Optimizer* = "SGD"

Berdasarkan hasil evaluasi menggunakan *confusion matrix*, model YOLOv8 yang dilatih dengan *optimizer* SGD menunjukkan performa yang cukup stabil dalam mendeteksi objek rokok akan tetapi sedikit tertinggal apabila dibandingkan dengan *optimizer* Adam.

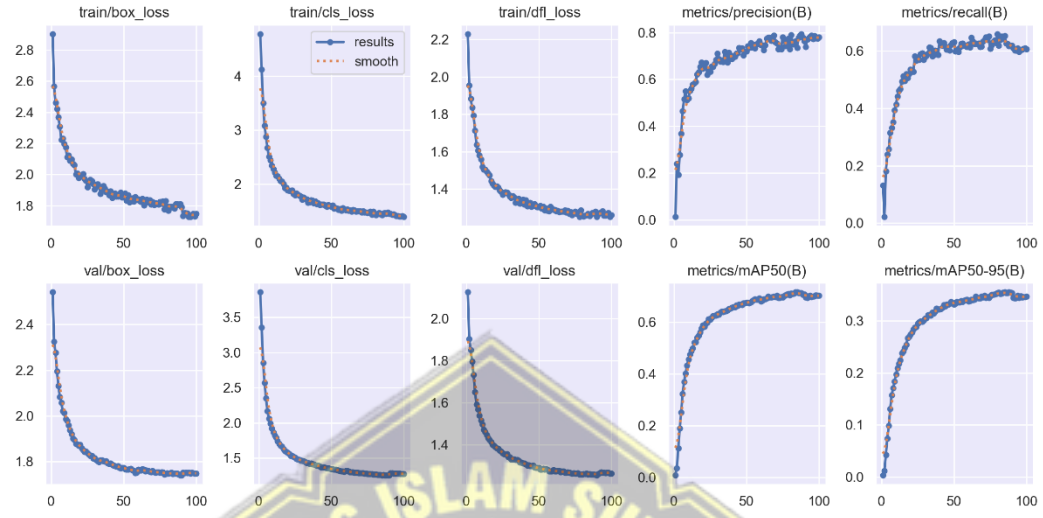


Gambar 4. 5 *Confusion Matrix Optimizer* SGD

Confusion matrix menunjukkan performa model dalam mengklasifikasikan objek menjadi dua kelas, yaitu *cigarette* dan *background*. Dari matriks tersebut terlihat bahwa model berhasil mengklasifikasikan 425 gambar *cigarette* dengan benar (*true positive*), dan 144 gambar *background* yang secara keliru diprediksi sebagai *cigarette* (*false positive*). Selain itu, terdapat 190 gambar *cigarette* yang diprediksi sebagai *background* (*false negative*), sedangkan prediksi benar terhadap *background* tidak ditampilkan secara eksplisit namun dapat diestimasi sebagai bagian dari *true negative*.

Nilai-nilai ini menunjukkan bahwa model cukup baik dalam mengenali objek *cigarette*, meskipun masih terdapat kesalahan klasifikasi terutama dalam membedakan antara *cigarette* dan *background*. Angka *false negative* yang lebih tinggi dari *false positive* menunjukkan bahwa model cenderung lebih sering melewatkan objek *cigarette* daripada salah mengklasifikasikan *background*. Hal ini penting untuk dievaluasi lebih lanjut karena dalam konteks deteksi rokok, kesalahan

melewatkan rokok (*false negative*) bisa lebih berdampak dibanding kesalahan mendeteksi *background* sebagai rokok (*false positive*).



Gambar 4. 6 Grafik Hasil Pelatihan *Optimizer* SGD

Gambar 4.6 menampilkan grafik hasil pelatihan dan validasi model menggunakan *optimizer* SGD selama 100 *epoch*. Grafik *loss* seperti *train/box_loss, train/cls_loss*, dan *train/dfl_loss* menunjukkan tren penurunan yang konsisten seiring bertambahnya *epoch*. Hal yang sama juga terlihat pada *val/box_loss, val/cls_loss*, dan val/dfl_loss, yang menandakan bahwa model mengalami proses pembelajaran yang baik tanpa gejala *overfitting* yang signifikan.

Pada metrik evaluasi, nilai *precision* dan *recall* meningkat secara bertahap hingga mencapai sekitar 0.75 dan 0.63. Hal ini menunjukkan bahwa model mampu mengenali objek dengan cukup baik, walaupun ada ruang untuk perbaikan terutama pada aspek *recall*, yang menunjukkan kemampuan model dalam mendeteksi semua objek positif (*cigarette*). Selain itu, metrik mAP50 dan mAP50-95 juga menunjukkan tren peningkatan yang stabil, dengan mAP50 mencapai mendekati 0.7 dan mAP50-95 mendekati 0.35.

4.1.5 Kesimpulan Pemilihan *Optimizer*

Berdasarkan hasil evaluasi performa model deteksi objek menggunakan tiga jenis *optimizer* yaitu SGD, Adam, dan Auto (AdamW), dapat disimpulkan bahwa *optimizer* Auto (AdamW) memberikan hasil yang paling optimal dalam pelatihan model YOLOv8 untuk mendeteksi objek rokok.

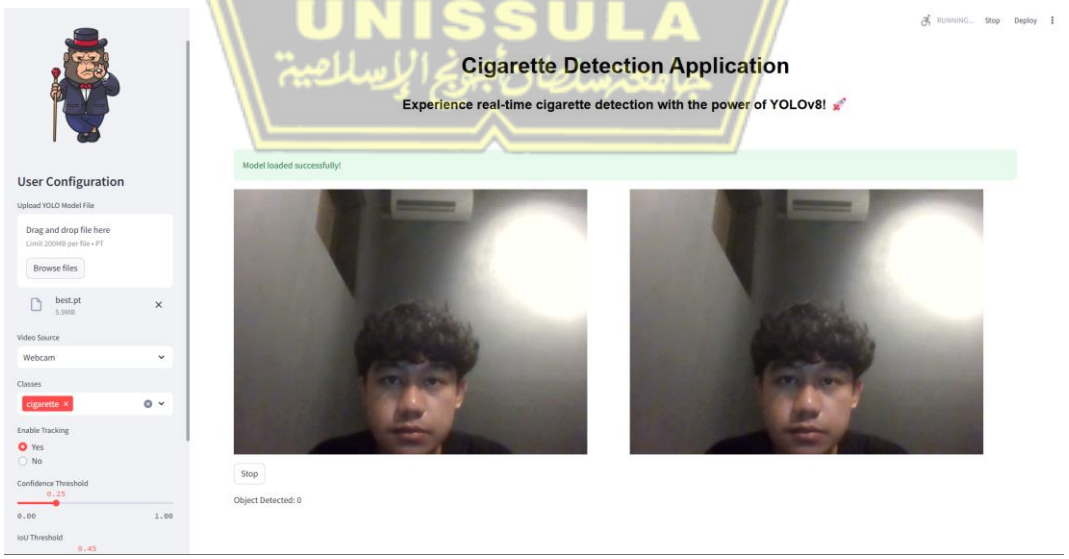
Optimizer Auto (AdamW) menunjukkan nilai *precision* dan *recall* yang lebih tinggi dibandingkan dua *optimizer* lainnya, yang berarti model lebih akurat dalam mendeteksi dan mengidentifikasi objek rokok tanpa banyak melewatkan atau salah mengklasifikasikan. Selain itu, nilai mAP50 dan mAP50-95 yang dihasilkan oleh *optimizer* ini juga lebih tinggi, menunjukkan bahwa model mampu mempertahankan performa yang baik pada berbagai ambang deteksi.

Dengan mempertimbangkan seluruh metrik evaluasi, maka dapat disimpulkan bahwa *optimizer* Auto (AdamW) merupakan pilihan terbaik untuk digunakan dalam pelatihan model deteksi rokok menggunakan YOLOv8, baik dari sisi akurasi, efisiensi pelatihan, maupun kestabilan proses pembelajaran.

4.2 Hasil Implementasi Aplikasi

4.2.1 Tampilan *User Interface*

Gambar di bawah ini menunjukkan tampilan antarmuka dari aplikasi deteksi rokok yang dibangun menggunakan model YOLOv8.

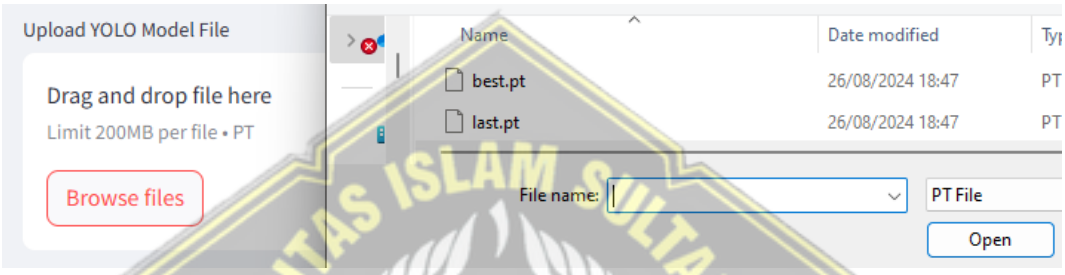


Gambar 4. 7 Tampilan Utama Aplikasi

Antarmuka pengguna (*user interface*) dari aplikasi deteksi rokok berbasis YOLOv8 dirancang menggunakan *framework* Streamlit, yang memungkinkan pengembangan aplikasi berbasis web secara cepat dan interaktif. Gambar 4.7 menunjukkan tampilan utama aplikasi ini, yang terbagi menjadi dua bagian utama: panel konfigurasi (*sidebar*) di sebelah kiri dan area tampilan utama (*main content area*) di sebelah kanan.

Berikut adalah penjelasan rinci dari masing-masing elemen antarmuka:

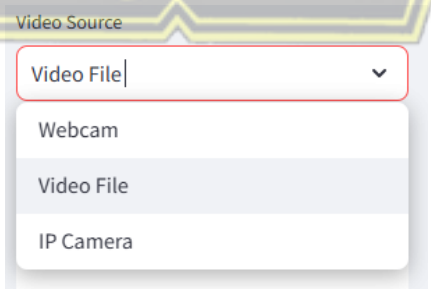
1. *Upload* YOLO Model *File*



Gambar 4. 8 *Upload* Model *File*

Bagian ini memungkinkan pengguna untuk mengunggah model deteksi berformat (*.pt*) yang sebelumnya telah dilatih menggunakan YOLOv8. Pengunggahan model ini bersifat fleksibel karena pengguna bisa mengganti model sewaktu-waktu tanpa perlu mengubah struktur kode. Model yang diunggah akan disimpan secara lokal dan digunakan sebagai dasar deteksi objek pada video atau *webcam*.

2. *Video Source* (Sumber Video)

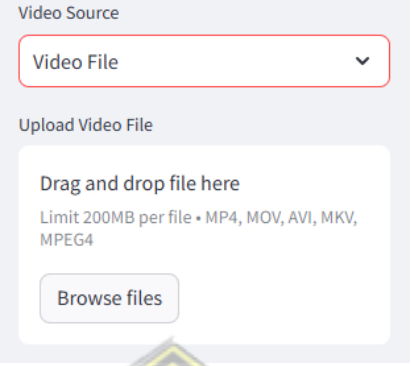


Gambar 4. 9 *Video Source*

Fitur ini memungkinkan pengguna untuk memilih sumber video yang akan digunakan sebagai *input* deteksi. Terdapat tiga opsi yang dapat dipilih:

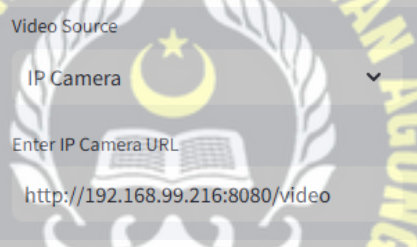
- *Webcam* : Menggunakan kamera internal perangkat (laptop/PC) secara langsung. Cocok untuk deteksi *real-time*.

- *Video File* : Pengguna dapat mengunggah *file* video (format MP4, AVI, dll.) untuk dideteksi oleh model.



Gambar 4. 10 *Video Source Upload File*

- IP *Camera* : Digunakan untuk *input* dari kamera berbasis IP, misalnya kamera CCTV yang terkoneksi jaringan, cukup dengan memasukkan URL *streaming*-nya.

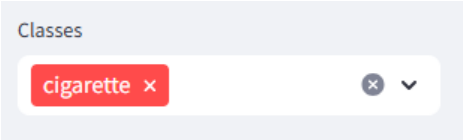


Gambar 4. 11 *Video Source* IP *Camera*

Fleksibilitas ini memudahkan pengguna untuk melakukan pengujian dalam berbagai skenario, baik secara langsung maupun menggunakan rekaman.

3. *Classes* (Kelas Objek)

Fitur ini menampilkan daftar kelas objek yang dapat dikenali oleh model.

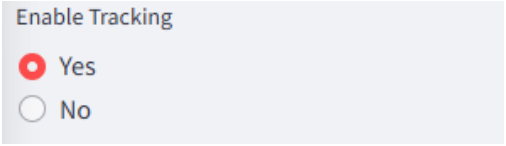


Gambar 4. 12 *Classes*

Dalam model yang telah dibuat sebelumnya, hanya terdapat satu kelas yaitu *cigarette* (rokok), karena model dilatih secara spesifik untuk mengenali objek tersebut. Pemilihan kelas ini berguna untuk mengurangi beban komputasi dan memfokuskan pendeteksian hanya pada objek yang dibutuhkan.

4. *Enable Tracking* (Aktifkan Pelacakan Objek)

Fitur ini memungkinkan pengguna untuk mengaktifkan atau menonaktifkan pelacakan objek (*tracking*).

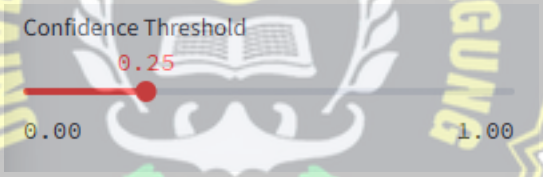


Gambar 4. 13 *Tracking Object Option*

Ketika pelacakan diaktifkan (*Yes*), sistem akan mencoba untuk mempertahankan identitas objek (ID) di antar *frame*, sehingga pergerakan rokok dapat diikuti secara konsisten. Ini berguna terutama dalam skenario di mana rokok berpindah posisi dalam *frame*. Jika pelacakan dimatikan (*No*), maka setiap *frame* akan dianalisis secara independen.

5. *Confidence Threshold* (Ambang Kepercayaan Deteksi)

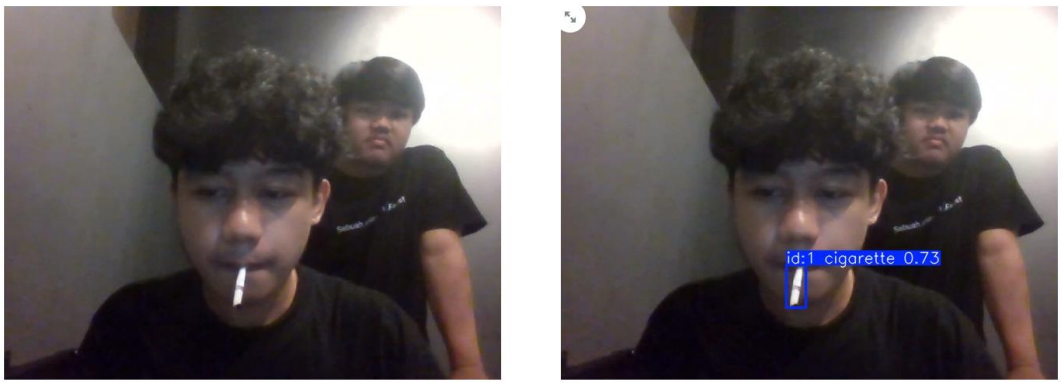
Fitur ini memungkinkan pengguna untuk mengatur *confidence threshold*, dengan rentang nilai antara 0.00 hingga 1.00.



Gambar 4. 14 *Setting Confidence Threshold*

Semakin tinggi nilai *threshold*, maka hanya deteksi dengan tingkat kepercayaan tinggi yang akan ditampilkan. Fitur ini penting untuk meminimalkan deteksi palsu (*false positive*).

6. Tampilan Frame Video



Gambar 4. 15 Tampilan *Input* dan *Output* Video

Aplikasi menampilkan dua kolom video secara berdampingan:

- Kolom Kiri (*Original Frame*) : Menampilkan gambar asli dari kamera atau video tanpa anotasi.
- Kolom Kanan (*Annotated Frame*) : Menampilkan hasil deteksi dengan anotasi berupa *bounding box*, label kelas (*cigarette*), dan skor *confidence* di atas setiap objek yang terdeteksi.

Dengan tampilan dua kolom ini, pengguna dapat membandingkan langsung antara *input* asli dan hasil proses deteksi.

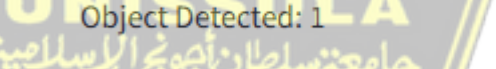
7. Notifikasi Deteksi

Saat model mendeteksi objek rokok di dalam *frame*, aplikasi akan memunculkan notifikasi peringatan berwarna kuning dengan tulisan "Objek Rokok Terdeteksi".

Gambar 4. 16 Notifikasi

Peringatan ini hanya muncul sekali setiap kali objek terdeteksi. Jika rokok tidak lagi terdeteksi, peringatan akan hilang secara otomatis.

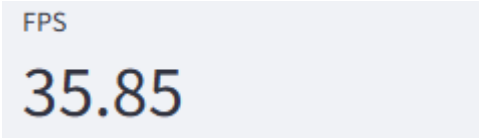
8. *Object Detected* (Jumlah Objek yang Terdeteksi)



Gambar 4. 17 Fitur *Object Counting*

Bagian ini menampilkan jumlah rokok yang terdeteksi dalam satu *frame* secara *real-time*. Angka ini diperbarui secara dinamis seiring dengan perubahan video.

9. FPS (*Frame per Second*)



Gambar 4. 18 FPS

Fitur ini menampilkan kecepatan pemrosesan *frame* oleh aplikasi (FPS). Nilai FPS yang tinggi menunjukkan bahwa sistem berjalan secara *real-time*. Sebaliknya,

jika nilai FPS rendah, hal ini bisa menandakan bahwa performa perangkat terbatas atau ukuran model terlalu besar. Pengguna dapat memantau FPS ini untuk mengevaluasi performa deteksi pada perangkat yang digunakan.

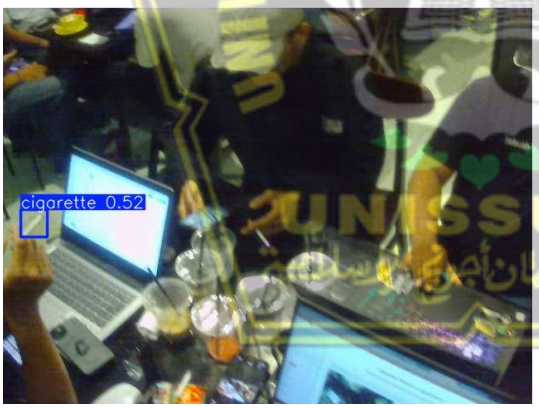
4.2.2 Pengujian Aplikasi Deteksi Rokok

Pengujian aplikasi pendeteksi rokok dilakukan dengan menggunakan tiga jenis sumber *input* video, yaitu melalui *webcam*, *upload file* video, dan IP *camera*. Pengujian bertujuan untuk mengetahui efektivitas dan akurasi deteksi objek rokok oleh model YOLOv8 dalam berbagai kondisi *input*.

1. Pengujian Menggunakan *Webcam*

Pengujian pertama dilakukan menggunakan *webcam* eksternal perangkat. Aplikasi dijalankan secara *real-time* dan mendeteksi objek rokok dari *feed* langsung *webcam*.

Tabel 4. 2 Pengujian Menggunakan Webcam

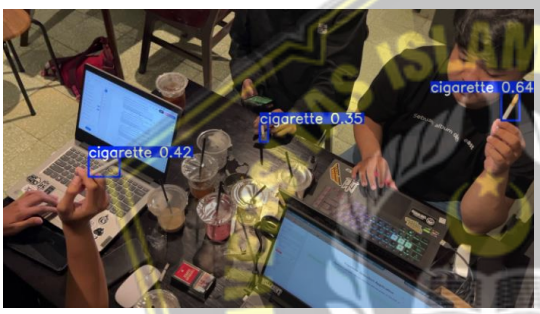
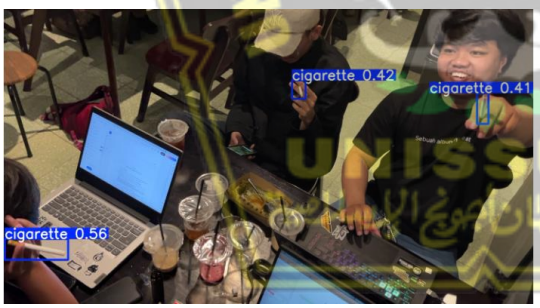
Gambar Input Pengujian	Hasil Prediksi	*Confidence*
	Berhasil	0.52
	Berhasil	0.7 dan 0.61

Berdasarkan Tabel 4.2, terlihat bahwa sistem berhasil mendeteksi 3 batang rokok dengan *confidence score* masing-masing sebesar 0.52, 0.61 dan 0.70. Setiap objek yang terdeteksi diberi *bounding box* berwarna biru dengan label "*cigarette*" dan nilai *confidence*-nya.

2. Pengujian Menggunakan *Upload File*

Pada pengujian kedua, *input* berasal dari *file* video yang diunggah ke dalam aplikasi.

Tabel 4. 3 Pengujian Menggunakan Upload File

Gambar Input Pengujian	Hasil Prediksi	*Confidence*
	Berhasil	0.42, 0.35 dan 0.64
	Berhasil	0.56, 0.42 dan 0.41

Hasil deteksi pada Tabel 4.3 menunjukkan bahwa sistem mampu mengidentifikasi tiga batang rokok setiap gambar dengan *confidence* yang bervariasi, yaitu 0.42, 0.35, dan 0.64 pada gambar pertama dan 0.56, 0.42, dan 0.41 pada gambar kedua. Perbedaan sudut pandang dan pencahayaan dalam video tidak menghalangi sistem untuk mengenali objek target, walaupun *confidence*-nya sedikit lebih rendah dibanding hasil dari *webcam*.

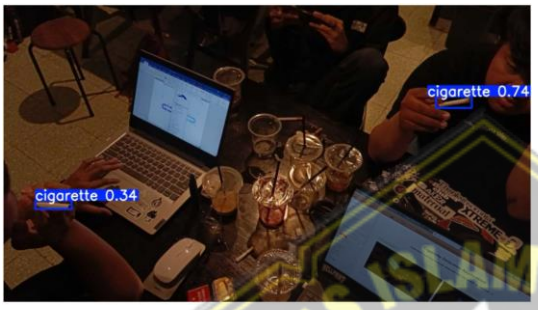
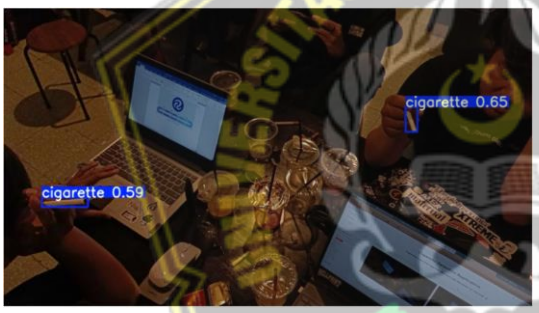
Hal ini menunjukkan bahwa model tetap responsif dalam kondisi pencahayaan dan perspektif kamera yang bervariasi. Namun, tidak adanya fitur pelacakan objek (*tracking* ID) pada mode ini membuat deteksi setiap *frame* bersifat

independen.

3. Pengujian Menggunakan IP *Camera*

Pengujian ketiga memanfaatkan *input* dari IP *camera* yang diakses melalui jaringan.

Tabel 4. 4 Pengujian Menggunakan IP Camera

Gambar Input Pengujian	Hasil Prediksi	*Confidence*
	Berhasil	0.34 dan 0.74
	Berhasil	0.59 dan 0.65

Gambar hasil pengujian ditunjukkan pada Tabel 4.4, di mana empat batang rokok berhasil dikenali dengan *confidence* masing-masing sebesar 0.34, 0.74, 0.59 dan 0.65. Meskipun kondisi pencahayaan dalam gambar cenderung gelap, model tetap dapat melakukan deteksi dengan cukup baik.

Penggunaan IP *camera* sangat bermanfaat untuk implementasi di area publik atau ruangan luas, seperti ruang kerja bersama atau tempat umum lainnya.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem deteksi rokok berbasis model YOLOv8 yang diintegrasikan dengan aplikasi web Streamlit berhasil dibangun dan diimplementasikan dengan baik. Sistem ini mampu mendeteksi keberadaan rokok dalam gambar atau video secara *real-time* melalui tiga jenis sumber *input*, yaitu *webcam*, IP *camera*, dan *file upload*.

Model telah dilatih menggunakan *dataset* dari Roboflow dengan parameter yang divariasikan, khususnya pada bagian *optimizer*. Tiga jenis *optimizer* yang digunakan adalah SGD, Adam, dan auto. Hasil evaluasi menunjukkan bahwa penggunaan *optimizer* auto, yang secara otomatis memilih algoritma AdamW, memberikan performa terbaik dengan nilai *precision* sebesar 0.878, *recall* sebesar 0.795, dan mAP50 sebesar 0.863 dan juga berdasarkan hasil yang ditampilkan *confusion matrix*, model mampu mendeteksi objek rokok dengan cukup baik, menghasilkan 529 prediksi benar (*true positive*). Namun, model masih menghasilkan 64 *false positive* dan 86 *false negative*, yang menunjukkan adanya beberapa kesalahan dalam mendeteksi rokok dan membedakannya dari latar belakang. Hal ini dapat diperbaiki dengan pelatihan lebih lanjut menggunakan data yang lebih beragam dan *augmentasi* yang tepat.

Pengujian dari ketiga jenis *input* menunjukkan bahwa sistem mampu bekerja secara responsif dan akurat dalam mendeteksi objek rokok, ditandai dengan tampilan *bounding box* dan perhitungan jumlah deteksi melalui antarmuka pengguna. Hal ini menunjukkan bahwa sistem dapat digunakan sebagai alat bantu dalam mengawasi aktivitas merokok di area-area yang telah ditetapkan sebagai kawasan bebas rokok

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, penulis memberikan

beberapa saran yang dapat dijadikan acuan untuk pengembangan sistem di masa mendatang, antara lain:

1. Penggunaan *Dataset* yang Lebih Variatif

   Agar model dapat bekerja lebih akurat di berbagai kondisi lingkungan, disarankan untuk menambah jumlah *dataset* dan memperluas variasi latar belakang, posisi rokok, dan jenis rokok (misalnya rokok elektrik, rokok berwarna selain putih dan cerutu).

2. Integrasi Sistem dengan Notifikasi Otomatis

   Pengembangan lebih lanjut dapat dilakukan dengan menambahkan fitur notifikasi otomatis, seperti pemberitahuan melalui email atau *dashboard* admin ketika terdeteksi aktivitas merokok.

3. Implementasi *Multi-Class Detection*

   Sistem dapat diperluas untuk mendeteksi aktivitas lain yang melanggar aturan (misalnya penggunaan *vape*, korek api, atau asap berlebihan) agar cakupan pengawasan menjadi lebih luas.

DAFTAR PUSTAKA

Akbar, M. (2021) 'Pengenalan Rambu Lalu Lintas Menggunakan Convolutional Neural Networks', *Jurnal Teknologi dan Sistem Komputer*, 9(2), pp. 120–125.

Aras, S., Tanra, P. and Bazhar, M. (2024) 'Deteksi Tingkat Kematangan Buah Tomat Menggunakan YOLOv5', *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(2), pp. 623–628. Available at: https://doi.org/10.57152/malcom.v4i2.1270.

Ardiansyah, A., Triloka, J. and Indera (2024) 'Evaluasi Kinerja Model YOLOv8 dalam Deteksi Kesegaran Buah', *JUPITER : Jurnal Penelitian Ilmu Dan Teknologi Komputer*, pp. 357–368.

Arsal, M., Agus Wardijono, B. and Anggraini, D. (2020) 'Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN', *Jurnal Nasional Teknologi dan Sistem Informasi*, 6(1), pp. 55–63. Available at: https://doi.org/10.25077/teknosi.v6i1.2020.55-63.

Cahyanti, M. (2023) 'SISTEM PENDETEKSI PLAT NOMOR POLISI KENDARAAN DENGAN ARSITEKTUR YOLOV8', 27(2), pp. 753–761. Available at: https://doi.org/10.46984/sebatik.v27i2.2374.

Gita Agustiguna Sidik (2024) 'Abstrak – Tindak kekerasan dan perundungan pada anak merupakan masalah serius yang memerlukan perhatian khusus, terutama di lingkungan sekolah. Deteksi kekerasan dan perundungan masih banyak dilakukan secara manual, yang rentan terhadap kesalahan dan mema', 3(9).

Gojali, M.I. and Tjiong, E.L. (2023) 'Pengembangan Aplikasi Deteksi Objek Rokok dan Kegiatan Merokok Menggunakan Algoritma YOLOv3', 10(02).

Hardjianto, M. (2023) 'PENERAPAN ALGORITME YOU ONLY LOOK ONCE VERSION 8 APPLICATION OF THE YOU ONLY LOOK ONCE VERSION 8 ALGORITHM FOR INDONESIAN SIGN LANGUAGE ALPHABET', 2(September), pp. 567–576.

Harkansas, D.D. (2024) 'KELENGKAPAN PENAMBAT PADA JALAN REL BERBASIS YOU ONLY LOOK ONCE VERSI 8', 2(1), pp. 1–10.

Hasa, M.F. *et al.* (2024) 'Deteksi Perokok Di Kawasan Bebas Rokok Menggunakan', 10(1), pp. 18–25.

Kalhor, G. and Bahrak, B. (2023) 'Analysis of Research Trends in Computer Science: A Network Approach', pp. 1–23. Available at: http://arxiv.org/abs/2311.09348.

Maulana, M.I. and Noviana, R. (2023) 'Training Custom Model Deteksi Udang menggunakan YOLOv8', *International Journal of Technology*, 47(1), p. 100950. Available at: https://doi.org/10.1016/j.tranpol.2019.01.002%0Ahttps://doi.org/10.1016/j.cstp.2023.100950%0Ahttps://doi.org/10.1016/j.geoforum.2021.04.007%0Ahttps://doi.org/10.1016/j.trd.2021.102816%0Ahttps://doi.org/10.1016/j.tra.2020.03.015%0Ahttps://doi.org/10.1016/j.

Meilita, B. and Yustanti, W. (2024) 'Sistem Deteksi Penyakit Kulit Kucing Menggunakan Algoritma You Only Look Once (YOLO) v8', 05(02), pp. 178–188.

Mukhlis, R.A. and Somantri, M. (2023) 'NEURAL NETWORK', 12(2), pp. 56–63.

Nafis Alfarizi, D. *et al.* (2023) 'Penggunaan Metode YOLO Pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis', *Jurnal Artificial Inteligent dan Sistem Penunjang Keputusan*, 1(1), pp. 54–63. Available at: https://jurnalmahasiswa.com/index.php/aidanspk.

Nugroho, B. and Puspaningrum, E.Y. (2021) 'Kinerja Metode CNN untuk Klasifikasi Pneumonia dengan Variasi Ukuran Citra Input', *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(3), pp. 533–538. Available at: https://doi.org/10.25126/jtiik.2021834515.

Prisky Ratna Aningtiyas, A.S. dan S.W. (2020) 'Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra - Terlatih', *Jurnal Ilmiah Komputasi*, 19(3), pp. 421–430. Available at: https://doi.org/10.32409/jikstik.19.3.68.

Putu, I., Jumariana, C. and Sugiartawan, P. (2024) 'Identifikasi Pengenalan Pola Daun Kelor Kering Dengan Yolo V8', *IJEIS (Indonesian Journal of*

Electronics and Instrumenations Systems, 14(1), pp. 91–100. Available at: https://doi.org/10.22146/ijeis.94871.

Setiyadi, A., Utami, E. and Ariatmanto, D. (2023) 'Analisa Kemampuan Algoritma YOLOv8 Dalam Deteksi Objek Manusia Dengan Metode Modifikasi Arsitektur', *Jurnal Sains Komputer & Informatika (J-SAKTI*, 7(2), pp. 891–901.

Shen, L., Lang, B. and Song, Z. (2023) 'Infrared Object Detection Method Based on DBD-YOLOv8', *IEEE Access*, 11(November), pp. 145853–145868. Available at: https://doi.org/10.1109/ACCESS.2023.3345889.

Terven, J., Córdova-Esparza, D.M. and Romero-González, J.A. (2023) 'A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS', *Machine Learning and Knowledge Extraction*, 5(4), pp. 1680–1716. Available at: https://doi.org/10.3390/make5040083.

Ultralytics (2023) *YOLOv8 Documentation*. Available at: https://docs.ultralytics.com.

Yudha, A.A. *et al.* (2021) 'Perancangan Sistem Deteksi Objek Pada Robot Transporter Menggunakan Metode Darknet YOLOv8'.