

SKRIPSI

**PENERAPAN METODE YOLO (*YOU ONLY LOOK ONCE*) DALAM
MENDETEKSI OBJEK SEBAGAI PENGAWASAN KEAMANAN
SECARA *REAL TIME***

DISUSUN OLEH

LISA NATHASYA

2109020159



UMSU

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2025

**PENERAPAN METODE YOLO (*YOU ONLY LOOK ONCE*) DALAM
MENDETEKSI OBJEK SEBAGAI PENGAWASAN KEAMANAN
SECARA *REAL TIME***

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada
Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas
Muhammadiyah Sumatera Utara**

LISA NATHASYA

NPM. 2109020159

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2025

LEMBAR PENGESAHAN

Judul Skripsi : PENERAPAN METODE YOLO (*YOU ONLY LOOK ONCE*) DALAM MENDETEKSI OBJEK SEBAGAI PENGAWASAN KEAMANAN SECARA REAL TIME

Nama Mahasiswa : Lisa Nathasya

NPM : 2109020159

Program Studi : Teknologi Informasi

Menyetujui
Komisi Pembimbing



(Amrullah, S.Kom., M.Kom)
NIDN. 0125118604

Ketua Program Studi



(Fatma Sari Hutagalung, S.Kom., M.Kom.)
NIDN. 0117019301

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

PENERAPAN METODE YOLO (*YOU ONLY LOOK ONCE*) DALAM MENDETEKSI OBJEK SEBAGAI PENGAWASAN KEAMANAN SECARA REAL TIME

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, September 2025

Yang membuat pernyataan



Lisa Nathasya

NPM. 2109020159

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Lisa Nathasya
NPM : 2109020159
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**PENERAPAN METODE YOLO (*YOU ONLY LOOK ONCE*) DALAM
MENDETEKSI OBJEK SEBAGAI PENGAWASAN KEAMANAN
SECARA REAL TIME**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, September 2025

Yang membuat pernyataan



Lisa Nathasya

NPM. 2109020159

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Lisa Nathasya
Tempat dan Tanggal Lahir : Medan, 18 September 2002
Alamat Rumah : Jl. Pancing 1 Martubung Gang. Manggis
No. 219, Lingkungan IV, Kelurahan Besar,
Kec. Medan Labuhan, Kota Medan,
Sumatera Utara, Kode pos 202051
Telepon/HP : 0877 9108 8840
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD	: SDN 0609044	TAMAT : 2014
SMP	: SMP LAKS. MARTADINATA	TAMAT : 2017
SMA	: SMA LAKS. MARTADINATA	TAMAT : 2020

KATA PENGANTAR



Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “PENERAPAN METODE YOLO (*YOU ONLY LOOK ONCE*) DALAM MENDETEKSI OBJEK SEBAGAI PENGAWASAN KEAMANAN SECARA REAL TIME” ini dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknologi Informasi, Fakultas Ilmu Komputer & Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara. Shalawat serta salam tidak lupa selalu tercurahkan kepada baginda rasulullah, junjungan besar Nabi Muhammad SAW. suri teladan bagi seluruh umat muslim yang penuh kasih.

Penulisan skripsi ini tidak lepas dari bantuan, dukungan, serta bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Agussani, M. AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU).
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom., Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom. Ketua Program Studi Teknologi Informasi .

4. Bapak Mhd. Basri, S.Si, M.Kom Sekretaris Program Studi Teknologi Informasi.
5. Dosen pembimbing penulis, Bapak Amrullah, S.Kom., M.Kom. yang telah memberikan arahan dan koreksi sehingga penulis dapat menyelesaikan skripsi ini.
6. Penulis juga ingin menyampaikan ucapan terimakasih kepada semua dosen, yang telah mengajarkan dan mendidik penulis dengan penuh rasa sabar dan ikhlas. Sehingga ilmu yang penulis dapatkan di bangku perkuliahan dapat menjadi ilmu yang bermanfaat untuk banyak orang.
7. Teristimewa penulis ucapkan terimakasih kepada kedua orang tua tercinta yakni cinta pertama penulis Ayahanda Ujang Kiaman dan pintu surga penulis Ibunda Evi Tarigan, terimakasih atas setiap tetes keringat dalam setiap langkah pengorbanan dan kerja keras yang dilakukan untuk memberikan yang terbaik kepada penulis, mengusahakan segala kebutuhan penulis, mendidik, membimbing dan selalu memberikan kasih sayang yang tulus, motivasi, serta dukungan dan mendoakan penulis dalam keadaan apapun agar penulis mampu bertahan untuk melangkah setapak demi setapak dalam meraih mimpi di masa depan. Terimakasih untuk selalu berada di sisi penulis dan menjadi alasan bagi penulis dalam menyelesaikan penulisan skripsi ini hingga memperoleh gelar Sarjana. Ayah, Ibu, putri kecilmu sudah dewasa dan siap melanjutkan mimpi yang lebih tinggi lagi.
8. Saudara kakak kandung saya Meilyta Pratiwi, S.E., Elma Silvia, S.Pd., dan Indah Mawarni, S.Pd. yang turut memberikan motivasi, perhatian, nasihat dan semangat kepada penulis serta abang ipar penulis, yang secara tak langsung

turut membantu penulis secara material ataupun non-material. Tak lupa delapan keponakan penulis yang selalu menghibur ketika penulis merasa bosan dalam penulisan karya ini.

9. Teruntuk sahabat dan teman terkasih penulis yakni Salsabila Ernisyah, terima kasih sudah ingin menjadi *partner* bertumbuh dari bangku sekolah menengah pertama hingga saat ini dan selamanya. Tak hanya itu, terimakasih telah menjadi teman suka duka di segala kondisi yang terkadang tidak terduga, dan pendengar yang baik bagi penulis serta menjadi orang yang selalu memberikan semangat dan meyakinkan penulis bahwa segala masalah yang dihadapi selama proses skripsi ini akan berakhir.
10. Sahabat penulis yang juga tak kalah penting kehadirannya yaitu Cindy Aulinia Siregar dan Ananda Rahmawati, yang telah selalu memberikan dukungan serta doa baik kepada penulis selama mengerjakan skripsi ini.
11. Para sahabat penulis dibangku perkuliahan yang selalu kebersamai dalam empat tahun ini, yang banyak membantu penulis terutama dalam mengerjakan skripsi dan tak pernah henti saling menyemangati satu sama lain.
12. Terimakasih kepada teman – teman *online di social media platform X*, sesama *dreamzen* yang telah memberikan beberapa motivasi, serta dukungan penuh dan semangat yang telah diberikan, juga telah meyakinkan penulis bahwa setelah hujan badai maka nantinya akan muncul pelangi yang indah.
13. Penulis juga ingin menyampaikan rasa terima kasih yang tulus kepada Park Jisung, salah satu anggota *favorite* penulis dari grup musik NCT DREAM, yang telah menjadi salah satu sumber semangat dan motivasi dalam menjalani masa perkuliahan, khususnya selama proses penulisan skripsi ini.

Kepribadiannya yang pantang menyerah, dan selalu bekerja keras demi menggapai mimpi, telah menginspirasi penulis untuk terus maju dan tidak mudah menyerah dalam menghadapi proses akademik ataupun non-akademik. Tak lupa, penulis juga ingin mengucapkan terima kasih kepada *7dream* yang melalui karya – karya dan konten mereka, telah memberikan hiburan serta menjadi *moodbooster* pelipur lelah bagi penulis.

14. Terakhir, terimakasih kepada wanita sederhana yang memiliki keinginan tinggi namun terkadang sulit dimengerti isi kepalanya, sang penulis yakni diri saya sendiri, Lisa Nathasya. Seorang anak bungsu yang berjalan menuju usia 23 tahun yang keras kepala dan gengsi namun terkadang sifatnya seperti anak kecil pada umumnya. Terimakasih ya telah hadir di dunia dan sudah bertahan sampai sejauh ini melewati banyaknya tantangan dan rintangan yang alam semesta berikan. Terimakasih kamu hebat saya bangga atas pencapaian yang telah di raih dalam hidup mu dan selalu merayakan dirimu sendiri sampai dititik ini, walau seringkali pengharapan tidak sesuai dengan ekspektasi dan selalu merasa takut akan kegagalan di masa depan, namun harus tetap bersyukur terimakasih selalu mau berusaha, bekerjasama dan tidak lelah mencoba hal – hal positif saya yakin dengan usaha, dan do'a yang selalu kamu langitkan Allah sudah merencanakan memberikan pilihan yang tidak terduga pastinya terbaik buat dirimu. Berbahagialah selalu dimanapun kapanpun kamu berada, Lisa. Rayakan selalu kehadiranmu, jadilah lebih baik dan bersinar dimanapun kamu memijakkan kaki. Semoga langkah kebaikan terus berada padamu dan semoga Allah selalu meridhoi setiap perbuatanmu dan selalu dalam lindungan-Nya. Aamiin..

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa yang akan datang. Akhir kata, penulis berharap skripsi ini dapat bermanfaat bagi semua pihak yang berkepentingan, khususnya bagi perkembangan ilmu pengetahuan di bidang ilmu komputer.

Medan, Agustus 2025

A handwritten signature in black ink, appearing to read 'Lisa Nathasya', with a stylized flourish at the end.

Lisa Nathasya

NPM. 2109020159

PENERAPAN METODE YOLO (*YOU ONLY LOOK ONCE*) DALAM
MENDETEKSI OBJEK SEBAGAI PENGAWASAN KEAMANAN SECARA
REAL TIME

ABSTRAK

Pengawasan keamanan merupakan aspek penting dalam mencegah dan merespons tindakan kriminal secara cepat. Sistem konvensional seperti CCTV umumnya hanya merekam tanpa kemampuan analisis otomatis terhadap potensi ancaman. Penelitian ini bertujuan untuk mengembangkan sistem deteksi objek berbahaya secara real-time dengan menerapkan metode YOLOv8 (You Only Look Once). Model deteksi dilatih untuk mengenali tiga kelas objek, yaitu manusia, senjata tajam (knife), dan senjata api (gun), menggunakan dataset yang telah diproses melalui platform Roboflow. Sistem ini diimplementasikan menggunakan bahasa Python dan framework Flask sebagai backend web, serta terintegrasi dengan kamera CCTV sebagai sumber input video. Hasil pengujian menunjukkan bahwa model YOLOv8 mampu mendeteksi objek secara akurat dan cepat, dengan nilai *precision* tertinggi sebesar 0.91 untuk kelas “Human” dan rata-rata *F1-score* sebesar 0.87. Sistem juga dilengkapi dengan fitur alarm suara dan notifikasi email secara otomatis ketika objek berbahaya terdeteksi, sehingga mendukung pengawasan keamanan yang lebih responsif dan efisien.

Kata Kunci: YOLOv8, Deteksi Objek, Keamanan, Real-Time, Kamera CCTV, Deep Learning

PENERAPAN METODE YOLO (*YOU ONLY LOOK ONCE*) DALAM
MENDETEKSI OBJEK SEBAGAI PENGAWASAN KEAMANAN SECARA
REAL TIME

ABSTRACT

Security surveillance is a crucial aspect in preventing and responding to criminal activities quickly. Conventional systems such as CCTV typically only record without the ability to automatically analyze potential threats. This research aims to develop a real-time dangerous object detection system by implementing the YOLOv8 (You Only Look Once) method. The detection model is trained to recognize three object classes: human, knife, and gun, using a custom dataset processed via the Roboflow platform. The system is implemented using Python and the Flask framework for the web backend, and it is integrated with CCTV cameras as the video input source. Test results show that the YOLOv8 model can detect objects accurately and efficiently, with the highest precision score of 0.91 for the “Human” class and an average F1-score of 0.87. The system is also equipped with an automatic alarm and email notification feature when dangerous objects are detected, enabling more responsive and effective security surveillance.

Keywords: YOLOv8, Object Detection, Security, Real-Time, CCTV Camera, Deep Learning

DAFTAR ISI

LEMBAR PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS	iii
PERNYATAAN.....	iv
RIWAYAT HIDUP	v
KATA PENGANTAR	vi
ABSTRAK	xi
ABSTRACT	xii
DAFTAR GAMBAR.....	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
1.5.1 Manfaat Teoritis.....	5
1.5.2 Manfaat Praktis	5
BAB II LANDASAN TEORI.....	7
2.1 Deteksi.....	7
2.2 Objek	8
2.2.1 Manusia	8
2.2.2 Senjata	8
2.3 Deep Learning	8
2.4 Algoritma.....	9
2.5 Algoritma YOLO (You Only Look Once)	10
2.6 YOLOv8.....	12
2.7 Computer Vision	13
2.8 OpenCV.....	15

2.9 Python.....	15
2.10 Collab Studio Google	16
2.11 Confusion Matrix.....	17
2.12 Penelitian yang Relevan	17
BAB III METODOLOGI PENELITIAN	22
3.1 Jenis Penelitian	22
3.2 Analisis Alat Penelitian	22
3.2.1 Perangkat Keras (<i>hardware</i>)	22
3.2.2 Perangkat Lunak (<i>software</i>).....	24
3.3 Tahapan Penelitian Sistem	25
3.3.1 Pengumpulan Dataset	26
3.3.2 <i>Pre-processing</i> Dataset	28
3.3.3 Pelatihan Dataset	30
3.3.4 Perancangan Sistem untuk Deteksi Objek	34
3.3.5 Pengujian Sistem dan Evaluasi	39
3.4 Jadwal Penelitian.....	41
BAB IV HASIL DAN PEMBAHASAN	42
4.1 Gambaran Umum Sistem	42
4.2 Analisis Model Training YOLOv8.....	42
4.3 Analisis Testing Model Pada Video	47
4.3.1 Hasil Deteksi Objek dalam Video.....	47
4.3.2 Dokumentasi Visual Hasil Uji Video.....	48
4.3.3 Evaluasi Respons Sistem	49
4.4 Analisis Performa Sistem Deteksi Objek.....	50
4.5 Implementasi Antarmuka WEB dan Backend	53
4.5.1 Integrasi Sistem	53
4.5.2 Tampilan Antarmuka WEB	55
4.6 Ujicoba Sistem dengan Kamera CCTV	57
BAB V KESIMPULAN DAN SARAN.....	60
5.1 Kesimpulan.....	60

5.2	Saran.....	61
	DAFTAR PUSTAKA	63

DAFTAR TABEL

HALAMAN

Tabel 2.1 Penelitian yang Relevan	18
Tabel 3.1 Spesifikasi Laptop	23
Tabel 3.2 Spesifikasi Kamera (IP/CCTV).....	23
Tabel 3.3 Jumlah Objek Deteksi	28
Tabel 3.4 Kode Pemanggilan Model YOLOv8.....	36
Tabel 3.5 Kode Fungsi Alarm	37
Tabel 3.6 Kode Fungsi Pengiriman Email	37
Tabel 3.7 Kode Fungsi Deteksi Frame	38
Tabel 3.8 Jadwal Penelitian	41
Tabel 4.1 Hasil Metrik Evaluasi Model Training.....	44
Tabel 4.2 Hasil Evaluasi Respons Sistem pada Pengujian Video	49
Tabel 4.3 Hasil Evaluasi Performa Sistem Deteksi Objek.....	50
Tabel 4.4 Hasil Uji Sistem dengan Kamera CCTV.....	58

DAFTAR GAMBAR

HALAMAN

Gambar 2. 1 Kerumunan Manusia	8
Gambar 2. 2 Proses Pemecahan Masalah (Algoritma)	10
Gambar 2. 3 Arsitektur YOLO	11
Gambar 2. 4 Ilustrasi YOLO	12
Gambar 2. 5 Logo YOLOv8	13
Gambar 2. 6 Identifikasi Objek	14
Gambar 2. 7 Motion tracking Objek Bergerak	14
Gambar 2. 8 Logo Python	16
Gambar 3. 1 Diagram Alur Penelitian	26
Gambar 3. 2 Dataset Senjata	27
Gambar 3. 3 Dataset Manusia	28
Gambar 3. 4 Proses Labelling Dataset	29
Gambar 3. 5 Hasil Pre-processing Dataset	30
Gambar 3. 6 Ilustrasi Penerapan Sistem Deteksi Objek Berbahaya	35
Gambar 3. 7 Diagram Alir Sistem Deteksi Objek	35
Gambar 4. 1 Confusion Matrix Model Training	43
Gambar 4. 2 Kurva Precision-Recall Model Training	46
Gambar 4. 3 Deteksi Objek ‘Knife’ dan ‘Gun’	48
Gambar 4.4 Contoh Kasus False Positive	51
Gambar 4.5 Contoh Kasus False Negative	52
Gambar 4.6 Tampilan Notifikasi Email Admin	54
Gambar 4.7 Halaman Utama Web	55
Gambar 4. 8 Tampilan Kamera Deteksi Web	56
Gambar 4. 9 Hasil Ujicoba Sistem di Siang Hari	58
Gambar 4. 10 Hasil Ujicoba pada Malam Hari	59

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Di Indonesia, terdapat lebih banyak kejahatan yang terjadi dibandingkan dengan beberapa negara tetangga. Pada tahun 2023, terdapat hampir 585.000 kejahatan, yang berarti sekitar 214 kejahatan terjadi untuk setiap 100.000 orang. Itu seperti mengatakan bahwa satu kejahatan terjadi kira-kira setiap 53 detik. Tahun lalu, pada tahun 2022, jumlah kejahatan lebih sedikit sekitar 373.000 dan kejadiannya lebih jarang, sekitar setiap 1 menit dan 24 detik. Jadi, kejahatan terjadi lebih cepat pada tahun 2023 dibandingkan dengan tahun 2022..

Telah terjadi peningkatan kejahatan, yang membuat orang khawatir tentang merasa aman. Meskipun polisi lebih baik dalam melacak kejahatan, masih sulit untuk menghentikan kejahatan agar tidak terjadi. Itulah mengapa kita perlu menemukan cara-cara baru untuk mencegah kejahatan sebelum terjadi. Sejauh ini pada tahun 2025, hingga 21 Januari, polisi telah menangani 24.327 kasus kejahatan, seperti pencurian, perkelahian, masalah narkoba, dan kekerasan.

Pengawasan keamanan menjadi salah satu aspek yang sangat penting dalam kehidupan sekarang, terutama untuk daerah perkotaan yang padat atau ramai penduduk. Sistem pemantauan seperti CCTV telah lama digunakan untuk mengawasi dan memantau aktivitas di ruang publik dengan kemampuan merekam video secara *real-time*. Namun, sistem ini belum dapat mendeteksi potensi ancaman, seperti senjata tajam atau api untuk mencegah tindakan kriminal,

pengawasan ini dapat memakan waktu yang lama dan juga berpotensi melewatkan ancaman. (Dahlan et al., 2021)

Pada kenyataannya pengawasan CCTV masih membutuhkan tenaga manusia untuk memantau secara manual, hal ini dapat mengakibatkan keterlambatan dalam respon terhadap situasi yang berbahaya. Penelitian oleh Iqbal dkk. (2021) menunjukkan bahwa sistem CCTV konvensional tidak dapat mendeteksi senjata tajam secara otomatis, yang membatasi kemampuan untuk memberikan respons cepat.

Deteksi objek merupakan salah satu aspek krusial dalam pengolahan citra serta visi komputer, yang memiliki beragam penerapan, antara lain pada sistem keamanan, interaksi manusia dengan komputer, maupun kegiatan pengawasan. Keberadaan metode deteksi objek yang efektif dan presisi menjadi sangat penting untuk menghadapi berbagai tantangan di lingkungan yang bersifat dinamis dan kompleks. Pada beberapa tahun terakhir ini, perkembangan dari teknologi khususnya pada bidang *deep learning*, telah memberikan kontribusi besar terhadap peningkatan kemampuan sistem deteksi objek (Prabowo & Abdullah, 2020).

Perkembangan kemajuan teknologi khususnya pada bidang *computer vision*, memiliki peluang untuk mengatasi keterbatasan ini semakin terbuka. YOLO (*You Look Only Look Once*) telah menunjukkan potensi besar dalam meningkatkan pengawasan *real-time*. Sistem berbasis YOLO dapat mendeteksi dan melacak objek dengan kecepatan tinggi dan akurasi yang baik dalam berbagai kondisi lingkungan. (Virgiawan et al., 2024)

Dalam penelitian ini, penerapan metode YOLO dapat mendeteksi objek dengan efisien akurat dan tepat dan penelitian ini bertujuan untuk

mengimplementasi metode YOLO untuk mendeteksi objek tidak dikenal dan menjadi bentuk pengawasan terhadap keamanan dari kejahatan secara *real-time*. YOLO dipilih sebagai model utama berdasarkan kemampuannya yang sangat unggul dalam deteksi objek secara *real-time* dengan efisiensi tinggi. Perihal ini dapat memberikan kontribusi yang signifikan dalam mengawasi keamanan dari situasi berbahaya dengan tepat juga cepat.

1.2 Rumusan Masalah

Dilihat dari uraian latar belakang tersebut, maka terdapat rumusan masalah pada penelitian ini yang dapat dijabarkan sebagai berikut:

1. Sejauh mana tingkat efektivitas metode YOLO dalam mendeteksi berbagai jenis objek pada sistem pengawasan keamanan?
2. Bagaimana metode YOLO dapat diintegrasikan dengan sistem pengawasan keamanan yang telah tersedia, seperti kamera CCTV maupun perangkat lunak analisis video?
3. Dataset apa saja yang digunakan dalam penerapan metode YOLO dalam mendeteksi suatu objek secara *real-time* melalui media video?
4. Bagaimanakah langkah-langkah penerapan metode YOLO dalam proses deteksi objek?

1.3 Batasan Masalah

Terdapat beberapa batasan permasalahan untuk menghindari pembahasan yang meluas, yaitu :

1. Fokus pada deteksi objek tertentu, seperti manusia, dan barang – barang mencurigakan tanpa mencakup objek lain yang mungkin relevan.

2. Algoritma yang digunakan adalah YOLOv8, dengan evaluasi kinerja model dibatasi pada metrik tertentu yaitu meliputi akurasi, presisi, *recall*, dan F1-*score*.
3. Penerapan metode YOLO akan dilakukan menggunakan perangkat keras tertentu, seperti kamera CCTV dengan resolusi tertentu. Kemudian pengembangan prototipe sistem deteksi objek mencakup integrasi dengan sistem alarm atau respon keamanan.
4. Algoritma pelatihan YOLO akan diimplementasikan dengan menggunakan *platform google collab*.

1.4 Tujuan Penelitian

Berdasarkan latar belakang serta rumusan masalah yang sudah dipaparkan, tujuan dari penelitian ini ialah:

1. Dikembangkan sekaligus mengimplementasikan model YOLOv8 yang dapat melakukan deteksi objek dengan tingkat akurasi dan efisiensi yang optimal.
2. Menilai kecepatan pemrosesan gambar oleh metode YOLO dan kemampuannya dalam memberikan respons cepat dalam situasi pengawasan keamanan yang memerlukan tindakan segera.
3. Menganalisis efektivitas metode YOLO dalam mendeteksi objek – objek tertentu, seperti manusia, dan barang mencurigakan dalam konteks pengawasan keamanan secara *real-time*.
4. Membandingkan keefektifitas metode YOLO dengan metode deteksi objek lain dalam konteks pengawasan keamanan secara *real-time*.

1.5 Manfaat Penelitian

1.5.1 Manfaat Teoritis

Manfaat teoritis berkaitan dengan kontribusi observasi terhadap pengembangan pengetahuan - pengetahuan ilmiah, konsep, serta teori pemantauan keamanan lingkungan penduduk menggunakan teknologi dengan metode YOLOv8 *Computer Vision*. Beberapa manfaat teoritis dari penelitian ini antara lain yaitu:

- a. Penelitian ini nantinya diharapkan memberikan kontribusi pada pengembangan ilmu pengetahuan di bidang pengolahan citra digital dan kecerdasan buatan, khususnya dalam metode deteksi objek menggunakan algoritma YOLO.
- b. Penelitian ini diharapkan dapat memperdalam pemahaman tentang cara kerja metode YOLO, termasuk kelebihan dan kekurangan dalam konteks mendeteksi objek secara *real time*.
- c. Hasil dari percobaan ini dapat dijadikan referensi atau rujukan bagi peneliti selanjutnya yang sedia mengembangkan atau menerapkan metode YOLO dalam berbagai aplikasi, baik di bidang keamanan maupun di bidang lainnya.
- d. Penelitian ini dapat menghasilkan model teoritis yang dapat digunakan untuk merancang sistem pengawasan keamanan yang lebih efektif dengan memanfaatkan teknologi deteksi objek.

1.5.2 Manfaat Praktis

Manfaat selanjutnya yakni manfaat praktis pada penelitian berfokus pada penerapan langsung dari sistem yang dikembangkan untuk meningkatkan tingkat keamanan masyarakat dan memudahkan pihak berwajib untuk pengawasan serta mengatasi tindakan kejahatan. Beberapa manfaat praktis dari penelitian ini antara lain:

- a. Hasil dari pengkajian ini dapat digunakan untuk meningkatkan sistem pengawasan keamanan yang ada, dengan menerapkan metode YOLO untuk deteksi objek secara *real time*, sehingga dapat mengurangi risiko ancaman keamanan.
- b. Penelitian ini dapat mendorong teknologi modern dalam sistem pengawasan, memberikan solusi yang lebih efisien dan efektif dalam mendeteksi objek yang berpotensi membahayakan.
- c. Dengan penerapan metode YOLO, sistem pengawasan dapat beroperasi dengan lebih efisien, yang pada gilirannya dapat mengurangi biaya operasional dan penggunaan sumber daya manusia dalam pengawasan.
- d. Dengan deteksi objek yang cepat dan akurat, sistem pengawasan dapat memberikan respons yang lebih cepat terhadap situasi darurat, meningkatkan keselamatan publik dan keamanan di area yang diawasi.

BAB II

LANDASAN TEORI

2.1 Deteksi

Deteksi atau mengenali objek, seperti mengajari komputer untuk melihat dan mengidentifikasi benda di gambar atau video. Misalnya, ini dapat membantu komputer menemukan orang, hewan, mobil, atau bangunan dalam foto. Tujuannya adalah agar komputer memahami apa yang mereka lihat sehingga mereka dapat melakukan hal-hal berguna, seperti menghitung berapa banyak orang yang ada dalam sebuah gambar, menemukan wajah, membaca kata-kata dalam gambar, atau mengenali plat nomor mobil. Ini sangat penting karena mengenali objek membantu komputer melakukan banyak tugas cerdas lainnya dengan gambar. Pada bagian bab ini, peneliti akan belajar lebih dalam mengenai bagaimana komputer belajar mengenali objek serta langkah - langkah apa yang akan peneliti tersebut ambil untuk melakukannya eksperimennya. (Marpaung et al., 2022)

Menurut (He & Eddie Law, 2024) menyatakan bahwa tugas deteksi objek dapat dilakukan secara otomatis dalam sistem komputasi untuk menemukan objek dalam gambar digital. Kemudian aplikasi dapat terus melakukan operasi visual lainnya, seperti menentukan kelas dan lokasi objek pada gambar.

Tujuan dari deteksi ialah untuk mereplikasi kecerdasan yang dimiliki manusia dalam melihat objek dengan menggunakan komputer. Cara kerja dari deteksi objek itu sendiri ialah dengan memposisikan keberadaan objek di dalam gambar serta memberikan kotak pembatas pada sekitar objek tersebut. (Robots et al., 2023).

2.2 Objek

2.2.1 Manusia

Definisi manusia dalam Kamus Besar Bahasa Indonesia (KBBI), merupakan suatu makhluk yang dianugerahi penalaran serta memiliki kemampuan yang mengalahkan makhluk lain. Tahap kehidupan yang dimiliki manusia terdapat lima tahap yaitu masa bayi, anak - anak, remaja, dewasa, hingga lanjut usia (lansia). Adapun lanjut usia (lansia) diartikan oleh Organisasi Kesehatan Dunia (WHO) tahun 2014 sebagai individu yang berupa 60 tahun ke atas.



Gambar 2. 1 Kerumunan Manusia

2.2.2 Senjata

Senjata dapat didefinisikan sebagai setiap objek yang digunakan untuk menyerang atau mempertahankan diri, yang dapat mencakup alat tradisional seperti pedang dan panah, hingga teknologi modern seperti senjata tajam sejenis pisau, senjata api, dan senjata canggih lainnya. (Muhamad Zulkarnaen, 2024).

2.3 Deep Learning

Menurut pendapat Batubara (Pembelajaran dalam (bahasa inggris: *Deep Learning*) atau pembelajaran struktural mendalam, yang juga dikenal sebagai

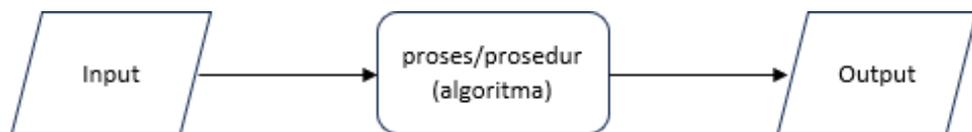
pembelajaran hierarki merupakan cara komputer belajar hal-hal, agak mirip dengan cara otak Anda bekerja. Ini menggunakan banyak lapisan, seperti balok bangunan yang disusun di atas satu sama lain, untuk memahami ide-ide yang rumit. Setiap lapisan membantu komputer memahami data dengan lebih baik, sehingga dapat mengenali pola, seperti gambar atau kata, dengan lebih mudah. (Pratiwi et al., 2021).

Deep Learning atau Pembelajaran Mendalam adalah cara baru bagi komputer untuk belajar memahami hal-hal dengan lebih baik. Ini dibangun di atas sesuatu yang disebut Jaringan Saraf Buatan, yang seperti otak kecil di dalam komputer. Dengan Pembelajaran Mendalam, komputer dapat melihat gambar atau mendengarkan suara dan mengetahui apa itu, seperti yang kita lakukan. Salah satu metode khusus yang digunakan dalam Pembelajaran Mendalam disebut Jaringan Saraf Konvolusional, atau CNN. Tipe 'otak' komputer ini dibuat untuk melihat benda datar seperti gambar dan suara, membantu komputer mengenali objek dalam foto atau memahami apa yang sedang dikatakan. (Muhamad Zulkarnaen, 2024).

2.4 Algoritma

Sebuah algoritma dapat dipahami sebagai rangkaian tindakan atau prosedur yang disusun dengan logis dan metodis dalam menyelesaikan permasalahan serta memberikan hasil yang diinginkan. Algoritma sangat penting dalam bidang komputasi dikarenakan berfungsi sebagai aturan sistematis yang harus diikuti oleh komputer. Algoritma yang tidak efektif dapat diibaratkan seperti memotong kayu dengan gunting yang tidak efisien dan memakan banyak waktu sedangkan algoritma yang baik diumpamakan seperti menggunakan peralatan yang tepat di sebuah bengkel. (Muhamad Zulkarnaen, 2024).

Algoritma adalah serangkaian instruksi yang tepat dan terorganisir yang dapat diikuti oleh komputer untuk melaksanakan tugas tertentu, termasuk pengolahan data, perhitungan, atau pengambilan keputusan, dalam konteks komputer dan bahasa pemrograman. Berbagai disiplin ilmu, termasuk matematika, ilmu komputer, dan rekayasa, juga dapat memperoleh manfaat dari penggunaan algoritma.



Gambar 2. 2 Proses Pemecahan Masalah (Algoritma)

(Sumber : Fahri Muhammad Zulkarnaen, 2024)

Algoritma atau prosedur pemecahan masalah ditunjukkan pada Gambar 2.2. Tahapan – tahapan proses ini terdiri atas tiga langkah, yaitu input, pemrosesan input, dan menghasilkan output.

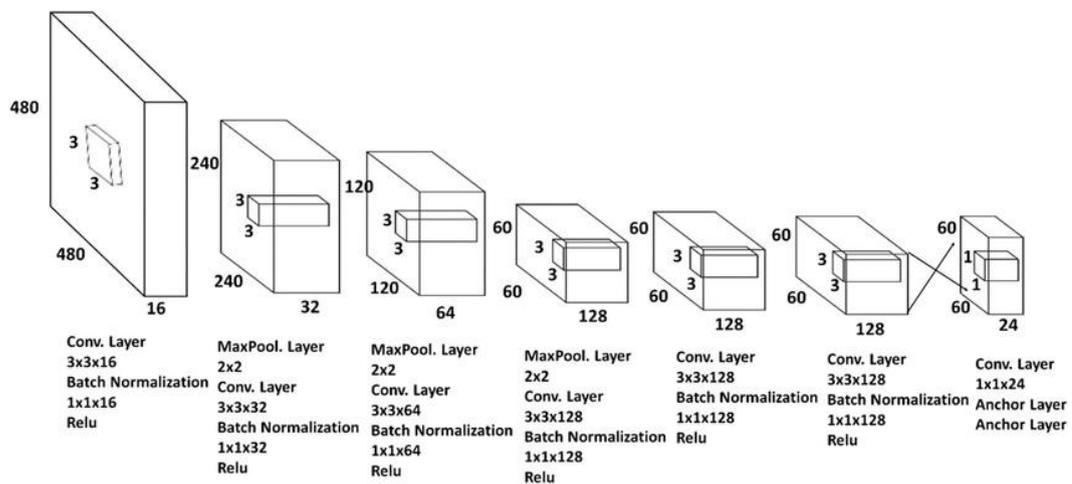
2.5 Algoritma YOLO (*You Only Look Once*)

You Only Look Once (YOLO) merupakan salah satu algoritma deteksi objek yang sangat populer pada bidang visi komputer. YOLO dirancang untuk dapat mendeteksi juga mengklasifikasikan objek dalam bentuk gambar atau video secara *real-time*. Keunikan dari YOLO terletak pada pendekatannya yang efisien, dimana YOLO sendiri dapat memproses seluruh gambar dalam satu kali pengamatan (*You Only Look Once*), berbeda dengan metode deteksi objek tradisional biasanya memproses gambar dalam beberapa tahap. (Hidayatulloh, 2021).

Algoritma YOLO memiliki beberapa versi mulai dari YOLO, YOLOv2, YOLOv3, *scaled*-YOLOv4 hingga YOLOv8. Algoritma YOLO hingga YOLOv4

sudah banyak digunakan untuk penelitian deteksi objek, Sebagaimana ditunjukkan oleh nilai AP sebesar 65%, algoritma YOLOv8 dalam penelitian ini digunakan untuk mengenali objek dengan cepat, efektif, dan andal.

Jaringan deteksi pada YOLO berjumlah 24 lapisan konvolusi (*convolutional layer*) diikuti dengan dua lapisan yang terhubung penuh (*fully connected layer*). Seperti terlihat pada gambar di bawah ini, macam – macam lapisan konvolusi menggunakan bagian lapisan reduksi 1x1 sebagai alternatif dalam mengurangi kedalaman pada *feature maps* kemudian diikuti dengan lapisan konvolusi 3x3.

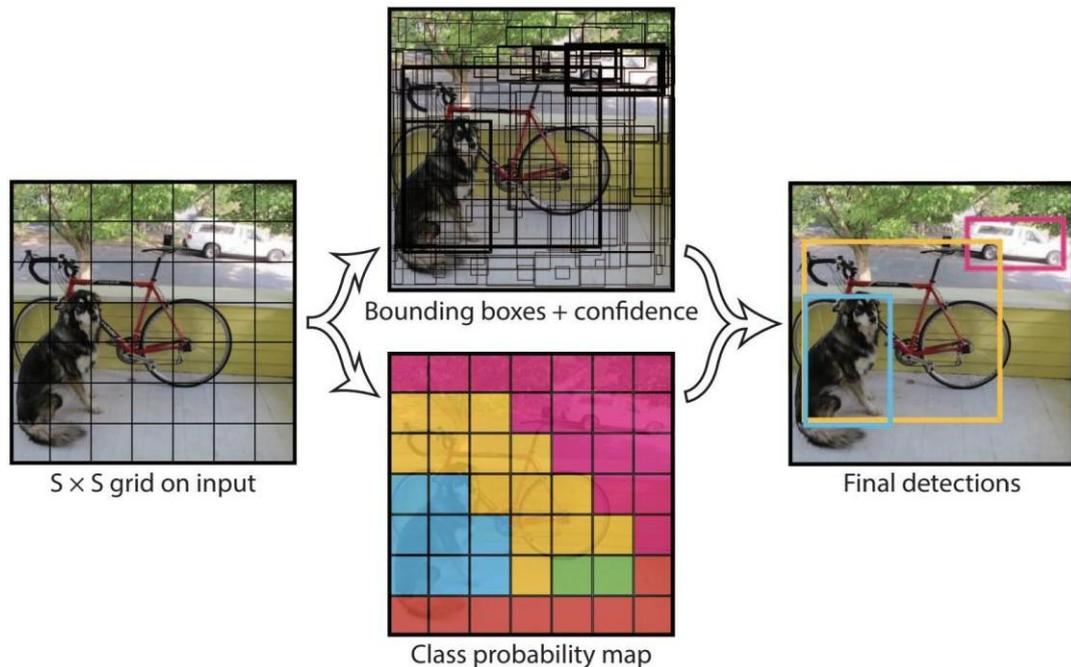


Gambar 2. 3 Arsitektur YOLO

(Sumber : Muhammad Rizky Pribadi, 2023)

YOLO menggunakan *Convolutional Neural Network* (CNN) yang pertama – tama memperbesar gambar input, kemudian membaginya menjadi grid dengan ukuran $S \times S$. melalui proses konvolusi, YOLO mengeluarkan kotak pembatas yang mencakup informasi mengenai koordinat x dan y , lebar, tinggi, juga skor kepercayaan dinormalisasi dalam rentang nilai 0 hingga 1, sedangkan koordinat x dan y dinormalisasi agar sesuai dengan titik kiri atas gambar, dan dimensi kotak disesuaikan dengan ukuran gambar. Proses ini digambarkan dalam gambar 2.4

setelah kotak pembatas dan skor kepercayaan diperoleh, langkah selanjutnya adalah melakukan pemrosesan lebih lanjut menggunakan jaringan saraf konvolusional pada masing – masing kotak tersebut. (Sunarto et al., 2024)



Gambar 2. 4 Ilustrasi YOLO

2.6 YOLOv8

Teknik YOLO dapat berjalan dengan kecepatan 45 fps (frame per detik) memakai kartu grafis Titan X berdasarkan hasil uji coba. Oleh karena itu, teknik ini sangat ideal untuk aplikasi yang memerlukan pengenalan objek secara real time pada foto maupun video. Seiring dengan meningkatnya jumlah penelitian mengenai identifikasi objek dalam bidang computer vision, YOLO dengan cepat berkembang hingga mencapai iterasi terbarunya, yaitu YOLOv8. Perbedaan antara semua versi YOLO terdapat pada tingkat presisi dan kecepatan dalam identifikasi objek. (Sunarto et al., 2024).

YOLOv8 juga merupakan algoritma yang terdiri dari berbagai layer. Layer pada algoritma YOLOv8 terbagi dalam 3 komponen utama yaitu *backbone*, *neck*, dan *head*. (Setiyadi et al., 2023). Implementasi YOLOv8 mengidentifikasi setiap piksel yang termasuk dalam objek, dan memberikan hasil yang lebih presisi dibandingkan deteksi objek biasa. YOLOv8 juga menggunakan teknik augmentasi data mosaik yang diterapkan selama pelatihan. *Mosaic* berfungsi untuk menggabungkan empat gambar input menjadi satu gambar, yang dapat membantu model untuk belajar mengenali objek dalam berbagai konteks. (Manurung et al., 2024).



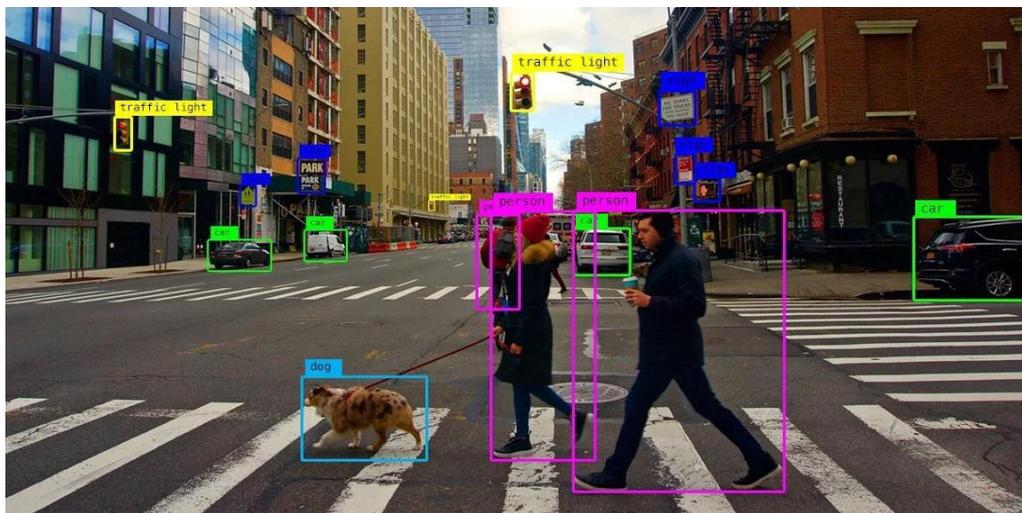
Gambar 2. 5 Logo YOLOv8

2.7 Computer Vision

Sejak awal tahun 1970-an, computer vision dianggap sebagai bagian dari persepsi visual dalam upaya ambisius para ilmuwan untuk mereplikasi kecerdasan manusia dan memberikan perilaku cerdas yang serupa dengan manusia pada mesin. Pada saat itu, beberapa orang berpendapat bahwa penemuan kecerdasan buatan pada tahun 1950-an akan mempermudah penggunaan computer vision dalam berbagai proyek penelitian.

Tujuan dari disiplin ilmu komputer pada bidang computer vision adalah untuk memberikan kemampuan kepada komputer atau perangkat lain agar dapat memahami, mengevaluasi, dan menafsirkan data visual dari gambar dan video.

Objek utama dari *computer vision* adalah menciptakan sistem yang dapat melihat. Tujuan dari penglihatan komputer adalah untuk membuat komputer melihat dan memahami hal-hal di sekitar mereka, sama seperti yang dilakukan manusia. Ini berarti komputer dapat mengenali wajah, melihat dalam 3D, melihat gambar medis, membaca tulisan tangan, memperhatikan gerakan, mengikuti objek, dan mengetahui apa itu. Ini adalah beberapa hal menarik yang bisa dilakukan komputer dengan 'mata' mereka. (Muhamad Zulkarnaen, 2024).



Gambar 2. 6 Identifikasi Objek



Gambar 2. 7 Motion tracking Objek Bergerak.

2.8 OpenCV

OpenCV adalah alat khusus yang membantu komputer melihat dan memahami gambar serta video secara real-time. Ini gratis digunakan, sehingga siapa pun dapat memanfaatkannya untuk proyek sekolah atau untuk menciptakan aplikasi keren. OpenCV dapat dipergunakan di berbagai macam komputer serta ponsel, seperti Windows, Mac, Linux, dan Android. Selanjutnya digunakan dengan berbagai bahasa pemrograman, seperti C++, Python, dan Java. Ini memiliki lebih dari 2.500 trik pintar (disebut algoritma) yang membuat penglihatan komputer menjadi lebih baik. (Muhamad Zulkarnaen, 2024).

Menurut (Mughtar et al., 2019), definisi dari OpenCV (*Open Source Computer Vision Library*) ialah pustaka komputer *open source* yang memiliki visi serta perangkat lunak pembelajaran didalamnya.

2.9 Python

Python memiliki semantik dinamis dan merupakan bahasa pemrograman yang berorientasi objek. Karena Python menggunakan tata bahasa yang khas dengan fokus pada keterbacaan, bahasa ini tergolong cukup sederhana. Dibandingkan dengan bahasa pemrograman lainnya, kode Python jauh lebih mudah dibaca dan dipahami oleh pengembang. Daripada itu, python juga sangat mendukung penggunaan modul juga paket, yang memungkinkan kode tersebut digunakan kembali dalam berbagai aplikasi serta memungkinkan sistem dikembangkan secara modular.



Gambar 2. 8 Logo Python

Salah satu keuntungan yang dimiliki bahasa pemrograman Python yaitu pustaka standarnya dapat diakses secara gratis dengan bentuk sumber maupun biner. Python adalah jenis bahasa komputer yang digunakan orang untuk memberi tahu komputer apa yang harus dilakukan. Bahasa ini diciptakan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991. Banyak orang menyukai Python karena sangat populer dan bermanfaat. (Triyanto et al., 2024).

2.10 Collab Studio Google

Google Colab adalah platform layanan komputasi awan yang mengarahkan pengguna untuk menjalankan kode bahasa pemrograman python langsung dari browser web mereka sehingga tidak perlu menginstal berbagai perangkat lunak tambahan. Platform ini terintegrasi dengan *Jupyter Notebooks* dan disediakan akses ke unit pemrosesan grafis (GPU) serta unit pemrosesan tensor (TPU) dimana sangat bermanfaat untuk pengembangan , analisis data, dan pembelajaran mesin. (Sunarto et al., 2024).

Google Colab memberikan kemudahan bagi *user* atau peneliti untuk melakukan eksperimen dan menjalankan kode program dalam lingkungan yang terintegrasi,

mendukung berbagai kebutuhan pemrosesan data dan algoritma pembelajaran mesin secara efisien.

2.11 Confusion Matrix

Hasil prediksi dari suatu permasalahan klasifikasi dirangkum dalam confusion matrix, yang juga dikenal sebagai error matrix. Untuk memberikan informasi mengenai jenis kesalahan yang dihasilkan oleh classifier, confusion matrix menggabungkan semua hasil klasifikasi benar serta klasifikasi salah menggunakan nilai hitung, setelah itu memecahkannya berdasarkan kelas masing-masing. (Umar et al., 2020). Ketentuan dari Confusion Matrix sebagai berikut :

1. Positive (P) : menunjukkan nilai aktual positif, sedangkan Negative (N) menunjukkan nilai aktual negatif.
2. True Positive (TP): terjadi ketika nilai prediksi positif dan nilai aktual juga positif.
3. True Negative (TN): nilai prediksi negatif dan nilai aktual juga negatif.
4. False Positive (FP): terjadi ketika nilai prediksi positif dan nilai aktualnya negatif.
5. False Negative (FN): terjadi ketika nilai prediksi negatif dan nilai aktualnya positif.

2.12 Penelitian yang Relevan

Metodologi penelitian ini dapat dikaitkan dengan sejumlah penelitian - penelitian sebelumnya. Berikut adalah beberapa penelitian yang relevan dengan penelitian ini :

Tabel 2.1 Penelitian yang Relevan

No	Judul	Penulis	Tahun	Kesimpulan
1	Sistem Keamanan Ruang dengan <i>Human Detection</i> Menggunakan Sensor Kamera Berbasis <i>Deep Learning</i>	(Surya et al., 2024)	2024	Pada penelitian ini dilakukan serangkaian langkah menggunakan metodologi CNN (Convolutional Neural Networks), Hasil akhir dari eksperimen tersebut model yang mereka buat dapat menemukan dan mengidentifikasi objek dengan sangat baik.
2	Deteksi Manusia dengan Algoritma YOLO untuk Pemutaran Audio Otomatis di Area Tertentu	(Pratiwi, 2025)	2025	Model yang digunakan pada penelitian ini adalah YOLOv5 pre-trained untuk mendeksi manusia, penelitian ini menunjukkan bahwa program telah menghasilkan ID disetiap objek dimana telah terdeteksi serta dapat memainkan audio dengan jarak waktu respons yaitu sekitar 0 sampai 0,7 milidetik.
3	Implementasi Metode <i>Object Detection</i> dengan Algoritma <i>You Only Look Once</i> (YOLO) Untuk Deteksi Kecurangan Di Dalam Ruang Ujian	(Nur et al., 2023)	2023	Penelitian ini memanfaatkan YOLOv4 untuk mendeteksi kecurangan di ruang ujian, seperti penggunaan handphone atau interaksi antar peserta ujian. Hasil menunjukkan mAP sebesar 86% tetapi terdapat tantangan dalam mengurangi overfitting dan meningkatkan generalisasi model.
4	Perancangan Sistem Deteksi Pelanggaran Penggunaan Helm Dengan Metode <i>Deep Learning</i>	(Batubara et al., 2024)	2024	Sistem deteksi pelanggaran helm pada penelitian ini dirancang dengan model akhir hasil pelatihan yang menggunakan YOLOv5

	Menggunakan YOLOv5 <i>Ultralytic</i>			memperoleh nilai mean average precision (mAP) rata-rata yaitu 0,938 sehingga memanfaatkan teknik deep learning berdasarkan pada algoritma dari convolutional neural network, yang menunjukkan kemampuan model dalam mendeteksi objek secara andal dan tepat.
5	Deteksi Objek Pisau Menggunakan Yolo <i>Machine Learning</i>	(Andreas Restu Priatama, 2023)	2023	Penelitian ini mengaplikasikan deteksi objek menggunakan YOLO untuk mendeteksi objek pisau serta penggunaan machine learning dan deep learning. Pengujian dengan jumlah data 600 menghasilkan akurasi sebesar 56%. Berdasarkan persentase akurasi yang dihasilkan masih terbilang rendah bisa jadi dikarenakan oleh jumlah dataset yang terbatas.
6	Sistem Presensi Menggunakan Deteksi Objek Wajah Mahasiswa Berbasis YOLOv5	(Rahayu et al., 2024)	2024	Penelitian ini memanfaatkan deep learning dan augmentasi data untuk meningkatkan akurasi dalam identifikasi mahasiswa. Penerapan sistem presensi mahasiswa dengan metode YOLO-V5 berhasil mendeteksi kehadiran mahasiswa secara realtime dengan tingkat akurat yang signifikan.
7	Implementasi Algoritma You Only Look Once (YOLO) menggunakan Web Camera untuk	(Zophie & Triharminto, 2022)	2022	Algoritma YOLO digunakan dalam penelitian ini bersama dengan bahasa pemrograman Python dan C. Dua skenario digunakan untuk menilai keandalan

	Mendeteksi Objek Statis dan Dinamis.			algoritma YOLO. Pada skenario pertama, dilakukan pengujian secara langsung menggunakan webcam yang telah terintegrasi dengan YOLO. Pada skenario kedua, yang bersifat tidak langsung, objek citra berupa video MP4 dan gambar JPG. Kedua skenario tersebut kemudian divariasikan lebih lanjut berdasarkan intensitas cahaya dan jarak.
--	--------------------------------------	--	--	--

BAB III

METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian atau observasi ini memanfaatkan metode eksperimental dengan pendekatan kuantitatif untuk mengevaluasi kinerja algoritma YOLO. Metode ini bertujuan untuk menguji model deteksi objek dalam pengawasan keamanan menggunakan algoritma YOLOv8, yang terkenal unggul dalam deteksi objek secara *real time* menggunakan teknik *deep learning*. (A. Bayu et al., 2023)

Model YOLO dipilih sebagai deteksi objek dikarenakan dapat mengenali sesuatu dengan cepat dan akurat, model YOLO digunakan untuk deteksi objek. Untuk meningkatkan keefektifan pengawasan keamanan atau melaporkan aktivitas yang mencurigakan, sistem ini dapat langsung menangkap dan memproses video dan mengirimkan peringatan dini melalui pemberitahuan secara *real time*.

3.2 Analisis Alat Penelitian

Pada proses analisa ini dipaparkan alat yang dibutuhkan, pada observasi ini terbagi dalam dua bagian yaitu perangkat keras (*hardware*) serta perangkat lunak (*software*), diantaranya :

3.2.1 Perangkat Keras (*hardware*)

Perangkat keras yang digunakan pada penelitian ini yaitu :

1. Laptop Acer *Aspire 5 14* sebagai server atau komputer sebelum memproses data serta pelatihan dataset dengan basis YOLOv8 untuk model pelatihan yang digunakan dalam sistem deteksi. Berikut spesifikasi laptop yang digunakan pada penelitian ini tertera pada Tabel 3.1.

Tabel 3.1 Spesifikasi Laptop

Komponen	Spesifikasi
Model Sistem	ACER Aspire 5 14" A514-56P-57Q8 (SKU : NX.KHRSN.001)
Prosesor	Intel Core i5-1335U
GPU	Intel Iris Xe Graphics (80 EU)
Storage	512 GB PCIe Gen 4 NVMe SSD (M.2)
Sistem Operasi	Windows 11 Home + Office Home Student 2021
Dimensi	318.8 × 225.5 × 17.99 mm
Memori	8 GB LPDDR5

2. Kamera Pengawas, seperti yang disebut kamera IP atau CCTV, adalah kamera khusus yang memantau dan mengirim video langsung untuk membantu mengidentifikasi orang dan barang seperti senjata. Kamera pengawas (IP/CCTV) terhubung ke internet sehingga dapat menunjukkan apa yang terjadi secara real-time dan juga menyimpan gambar atau video. Kamera ini memiliki gambar yang jelas dan detail sehingga dapat mendeteksi sesuatu dengan lebih mudah. Dalam penelitian ini, kami menggunakan kamera yang disebut IMOU, dan rinciannya ditampilkan dalam Tabel 3.2.

Tabel 3.2 Spesifikasi Kamera (IP/CCTV)

Spesifikasi	Nilai
Model	IMOU Bullet 2E
Sensor	1/2.8" Progressive CMOS
Resolusi	2 MP (1080p) atau 3/5 MP
Lensa fixed	2.8 mm / 3.6 mm (opsional 6 mm)
Sudut pandang	~102°–120° diagonally
Penyimpanan	microSD up to 256 GB (atau 512 GB dukungan)
Jaringan & ONVIF	Wi-Fi 2.4 GHz dual antenna + Ethernet 100 Mbps, ONVIF kompatibel
Zoom Digital	8x zoom
Audio	Mic + speaker (2-way audio)
Night Vision	30 m infrared + spotlight
Kompresi & Frame rate	H.265/H.264, up to 25/30 fps

3. Perangkat jaringan (*Router & Switch*) untuk menghubungkan kamera dengan unit pemrosesan dan memungkinkan pengiriman data secara *real time*. *Switch* dapat digunakan untuk menghubungkan semua perangkat kamera ke jaringan.

3.2.2 Perangkat Lunak (*software*)

Pada eksperimen ini digunakan berbagai jenis perangkat lunak diantaranya yaitu :

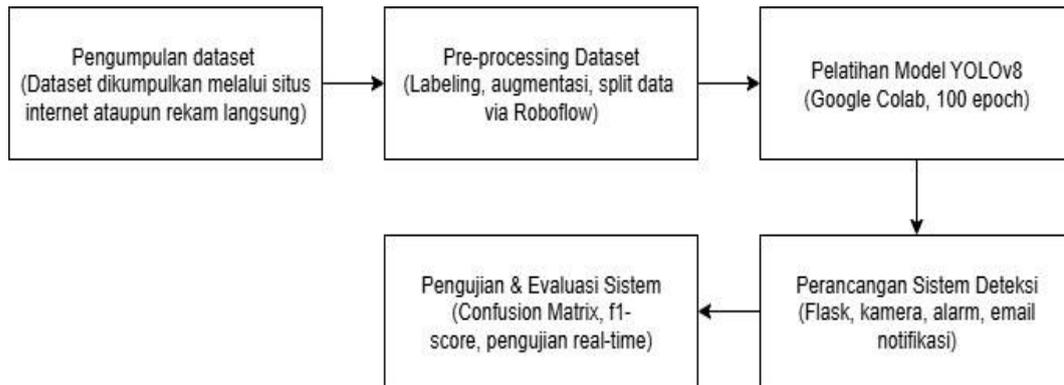
1. Bahasa pemrograman *Python* yang mudah dipahami dan digunakan untuk berbagai aplikasi perangkat lunak. *Python* akan memproses input data dari cctv, melakukan *pre-processing* data, membuat model deteksi objek manusia dan senjata menggunakan algoritma *deep learning* seperti YOLOv8, menguji model, dan menampilkan output berupa hasil deteksi objek manusia dan senjata tersebut dalam suatu bentuk visualisasi.
2. *Roboflow* sebagai platform berbasis web yang digunakan untuk memberi label pada gambar objek, mengelola, dan mempersiapkan dataset berupa gambar – gambar yang mengandung objek manusia, senjata api, dan senjata tajam agar digunakan untuk melatih model YOLOv8 secara optimal.
3. Platform Google Colab digunakan sebagai proses pelatihan (*training*) model deteksi objek, dikarenakan google colab menyediakan sumber daya komputasi berbasis *cloud* yang dilengkapi dengan GPU secara gratis. Selain itu, Google Colab juga memudahkan integrasi dengan platform *Roboflow*, sehingga dataset yang telah diberi anotasi di *Roboflow* dapat langsung digunakan dalam proses pelatihan tanpa harus diunduh secara manual. Google Colab juga menyediakan menjalankan dan memantau setiap langkah pelatihan, mulai dari pra-

pemrosesan data, konfigurasi model, pelatihan, hingga evaluasi kinerja model berdasarkan metrik mAP, *Precision*, dan *Recall*.

4. Visual Studio Code berperan sebagai alat yang digunakan untuk menjalankan kode program deteksi objek dengan menggunakan bahasa pemrograman *Python* serta membangun aplikasi berbasis web menggunakan *framework Flask*. Pada VS Code, model hasil pelatihan diintegrasikan ke dalam program yang mampu mendeteksi objek secara langsung melalui kamera, memberikan peringatan visual berupa *bounding box*, memicu alarm suara, serta mengirimkan email notifikasi jika terdeteksi objek manusia yang membawa senjata.
5. Email digunakan sebagai salah satu mekanisme notifikasi otomatis ketika sistem mendeteksi keberadaan objek yang berbahaya, seperti manusia yang membawa senjata (senjata api atau senjata tajam). Tujuan utama dari penggunaan email adalah untuk memberikan peringatan jarak jauh secara real time kepada pengguna atau petugas keamanan bahwa telah terjadi potensi ancaman yang terdeteksi oleh sistem.

3.3 Tahapan Penelitian Sistem

Tahapan penelitian sistem ini dirancang untuk menghasilkan sistem pendeteksi objek berbahaya dengan real-time berbasis metode YOLOv8 (*You Only Look Once*). Penelitian ini menggunakan pendekatan eksperimen dan pengembangan sistem, dimulai dari tahap perencanaan, pengumpulan dataset, *pre-processing* dataset, pelatihan dataset, hingga pengujian sistem. Tahapan – tahapan dalam proses pengembangan sistem digambarkan secara sistematis seperti berikut.



Gambar 3. 1 Diagram Alur Penelitian

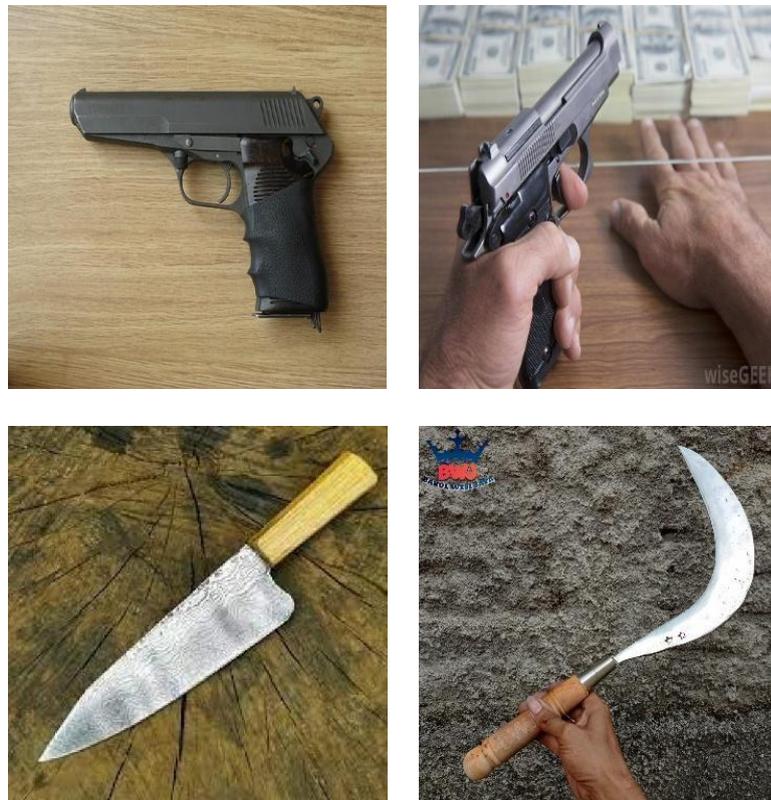
Pada Gambar 3.1, prosedur - prosedur tertentu wajib dilaksanakan dalam observasi ini agar dapat memperoleh hasil akhir yang diinginkan dan dalam proses pengembangan sistem digambarkan secara sistematis agar setiap proses proses dapat dilakukan secara terstruktur, efisien, dan terdokumentasi dengan baik. Dimulai dari pengumpulan dataset yang kemudian dataset tersebut akan dilakukan proses pelabelan dan pelatihan, dan kemudian melakukan perancangan sistem untuk deteksi objek. Dataset yang telah siap akan dilakukan uji coba dan evaluasi.

3.3.1 Pengumpulan Dataset

Pengumpulan dataset merupakan proses pengumpulan citra yang berupa gambar atau *image* yang telah didapat melalui *google* atau video yang di ambil pada lapangan. (Dadang Iskandar Mulyana & M. Ainur Rofik, 2022). Pada tahapan awal penelitian ini, proses pengumpulan dataset yang digunakan untuk pelatihan dan pengujian model deteksi objek berbasis YOLOv8 merupakan kumpulan gambar yang merepresentasikan objek yang ingin dideteksi, yaitu manusia (*human*), senjata tajam (*knife*), dan senjata api (*gun*).

Dataset dikumpulkan dari dua sumber utama, yaitu melalui internet atau mesin pencari seperti *Google Images* dengan kata kunci terkait ketiga objek tersebut

dan dipilih sesuai dengan gambar yang relevan dan jelas memperlihatkan objek target dari berbagai sudut, kondisi pencahayaan, dan latar belakang yang berbeda untuk memperkaya variasi data. Sumber lain yaitu melalui rekaman atau tangkapan gambar langsung menggunakan kamera *handphone*. Pada Gambar 3.2 dan Gambar 3.3 adalah contoh kumpulan dataset senjata tajam, senjata api dan dataset manusia yang telah dikumpulkan dan kemudian akan dilakukan pelabelan pada masing – masing gambar dan pelatihan.



Gambar 3. 2 Dataset Senjata



Gambar 3. 3 Dataset Manusia

Data terdiri dari tiga kelas, yaitu dataset manusia (human), dataset senjata tajam atau pisau (knife), dan dataset senjata api (gun). Dimana terdapat total keseluruhan dataset adalah sebanyak 1.824 gambar.

Tabel 3.3 Jumlah Objek Deteksi

Objek	Jumlah
Manusia	608
Senjata Tajam	608
Senjata Api	608
Total	1.824

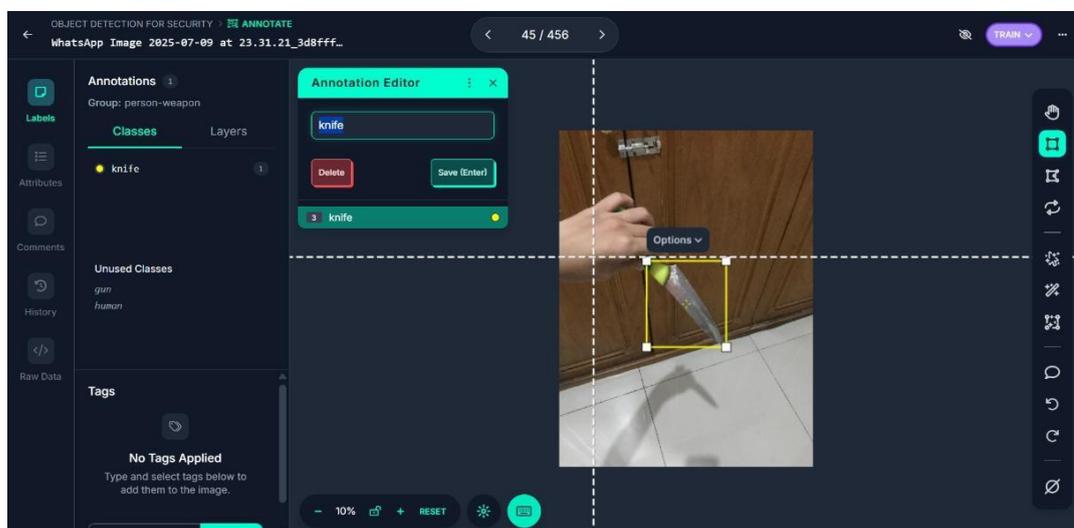
Pada Tabel 3.3, terdapat rincian jumlah total data yang setelah dikumpulkan semua data diunggah ke platform Roboflow untuk dilakukan proses *labeling* (pemberian anotasi) menggunakan *bounding box* serta penyesuaian format ke format YOLOv8.

3.3.2 Pre-processing Dataset

Dalam metode YOLO, tahapan ini bertujuan agar dataset yang akan dibuat membuat dataset yang akan digunakan untuk melatih model YOLOv8. Dataset terdiri dari tiga bagian utama yaitu *training*, *validation*, dan *testing*, dataset yang

telah diambil dari video yang sudah direkam sebelumnya. *Pre-processing* dilakukan menggunakan *Roboflow*, yang menyediakan fitur sebagai berikut :

1. Re-size gambar ke ukuran 640 x 640 piksel agar sesuai dengan standar input model YOLOv8.
2. Augmentasi data seperti blur, *noise*, *brightness adjustment*, dan sebagainya untuk meningkatkan variasi dari data dan juga ketahanan model tersebut terhadap berbagai kondisi pencahayaan serta sudut pandang yang berbeda.
3. Split data secara otomatis menjadi data latih (*training*), validasi (*valid*), dan pengujian (*testing*) dengan rasio 70 : 20 : 10.
4. Dataset yang telah dilatih kemudian diekspor ke dalam bentuk format YOLOv8. Dataset kemudian diunduh dan di-*extract* ke dalam direktori proyek untuk proses pelatihan.

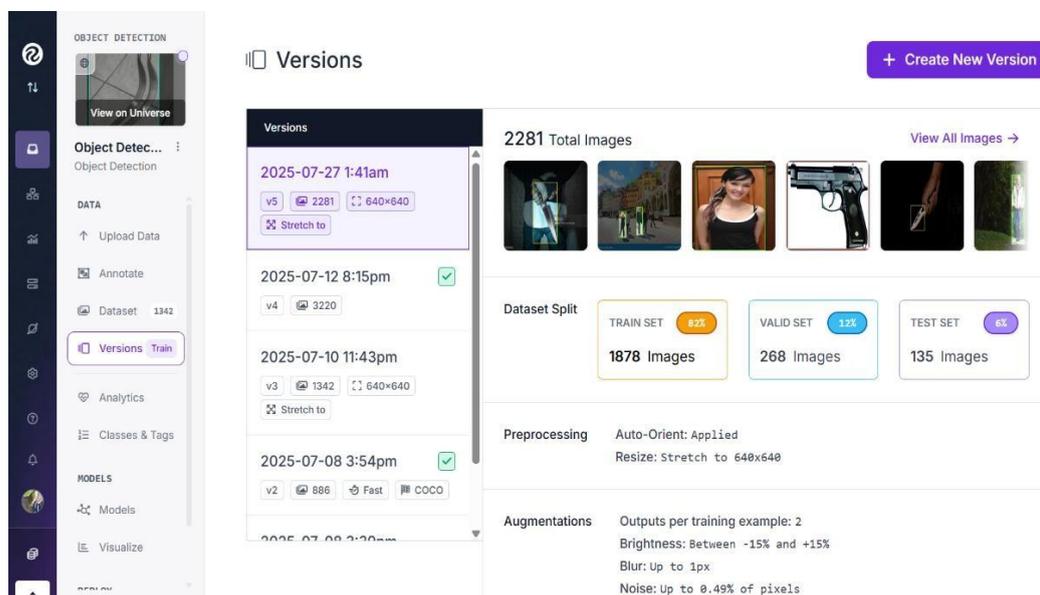


Gambar 3. 4 Proses *Labelling* Dataset

Pada tahapan ini, frame yang telah berhasil diekstrak kemudian disimpan dalam bentuk format file .jpg. Dari total 1.824 gambar yang diekstrak, sebanyak 1.324 gambar yang relevan telah dipilih dan dilakukan proses anotasi (*labeling*)

bounding box sesuai kelasnya, kemudian selebihnya ialah hasil data frames yang kurang atau tidak relevan. Selanjutnya gambar - gambar itu dilakukan proses agumentasikan dengan metode *brightness*, *noise*, dan *blur* sehingga menghasilkan jumlah dataset dua kali lipat sebanyak 2.281 *frames*. Keseluruhan dataset yang selesai diproses kemudian diekspor ke dalam format YOLOv8, yang terdiri dari file gambar (.jpg) dan file anotasi (.txt) yang berisi informasi *bounding box* dalam koordinat relatif terhadap ukuran gambar.

Berdasarkan hasil akhir pre-processing, diperoleh pembagian dataset sebanyak 1.878 *images* untuk *training* set, valid set yaitu 268 *images*, dan test set yaitu sebanyak 135 *images*. Struktur dataset ini kemudian digunakan dalam proses pelatihan model di platform Google Colab dengan memanfaatkan GPU untuk mempercepat proses pelatihan dan validasi.



Gambar 3. 5 Hasil Pre-processing Dataset

3.3.3 Pelatihan Dataset

Proses pelatihan dataset pada penelitian ini dimulai dengan mempersiapkan dataset yang telah dibuat sebelumnya. Dalam proses pelatihan model deteksi objek

dalam pengawasan keamanan pada penelitian ini, bertujuan agar model YOLOv8 mampu mengenali dan mengklasifikasikan objek – objek yang telah dianotasi pada dataset, yaitu objek manusia (human), senjata tajam (knife), dan senjata api (gun), dengan akurasi yang tinggi.

Proses dari pelatihan terhadap model dilakukan dengan menggunakan bantuan *Google Colab* dengan memanfaatkan GPU (*Graphics Processing Unit*) secara gratis. GPU digunakan untuk mempercepat proses pelatihan model karena pelatihan *deep learning* memerlukan daya komputasi tinggi. *Framework* yang digunakan untuk pelatihan adalah *ultralytics* YOLOv8, yang merupakan pengembangan dari YOLOv5 dan mendukung berbagai fitur seperti integrasi dengan Python, pelatihan custom dataset, dan monitoring hasil pelatihan secara langsung.

Dataset hasil dari proses pre-processing kemudian diunggah ke Google Colab dengan struktur direktori yang terdiri dari folder *images/train*, *images/val*, dan *images/test* untuk gambar, serta folder *labels/train*, *labels/val*, dan *label/test* untuk file anotasi label dalam format .txt . Jumlah gambar pada masing – masing bagian dataset adalah sebanyak 1.878 *image* sebagai data latih (*training*), 268 *image* untuk data validasi (*validation*), dan 135 gambar untuk data uji (*testing*).

Proses pelatihan dilakukan dengan mengatur parameter – parameter penting seperti jumlah epoch, ukuran batch, ukuran gambar (*image size*), dan *learning rate*. Berdasarkan pelatihan yang telah dilakukan selama 100 epoch dengan *batch size* 16 dan resolusi gambar 640 piksel. Model dilatih menggunakan perintah Python dari pustaka *Ultralytics*. File *config.yaml* berisi konfigurasi jalur dataset dan daftar nama kelas objek. Selama pelatihan berlangsung, sistem akan melakukan evaluasi

terhadap data validasi dan mencatat berbagai metrik performa, antara lain *precision*, *recall*, *f1-score*, mAP50, dan mAP50-95. Model terbaik yang diperoleh selama pelatihan akan disimpan dalam file *best.pt*, yaitu model dengan performa evaluasi tertinggi pada data validasi.

Setelah proses pelatihan selesai, model tersebut kemudian digunakan untuk pengujian dengan menggunakan data testing. Data testing yang digunakan tidak pernah dilibatkan dalam proses pelatihan agar hasil evaluasi benar – benar mencerminkan kemampuan generalisasi model terhadap data baru. Data testing ini memiliki karakteristik yang sama dengan data latih dan validasi, sehingga pengujian dianggap valid.

Ketika menguji seberapa baik program komputer dalam menemukan benda dalam gambar, dapat menggunakan alat khusus untuk melihat seberapa baik kinerjanya. Salah satu alat ini disebut "confusion matrix" yang membantu pengguna memahami kapan program dengan benar menemukan objek, melewatkan beberapa, atau membuat kesalahan. Dari matriks ini, pengguna dapat menentukan seberapa akurat cara menemukan semua objek kemudian pengguna dapat memutuskan apakah program tersebut cukup baik atau perlu memperbaikinya, misalnya dengan memeriksa gambar-gambar tersebut secara berkala.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$f_1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$AP_i = \int_0^1 Prec(Rec)d(Rec) + C$$

$$mAP = \frac{1}{N} \sum_i^N AP_i$$

Keterangan :

1. *Precision* diukur melalui proporsi prediksi positif yang sebenarnya benar mengingat total jumlah prediksinya positif dari modelnya.
2. *Recall* mengukur proporsi contoh positif yang sebenarnya diprediksi dengan benar mengingat total jumlah contoh positif.
3. *F1-score* menyeimbangkan presisi dan recall dengan mengambil rata-rata harmonik dari kedua nilai ini.
4. Rata-rata Presisi (*Average Precision*) ialah metrik yang dipergunakan untuk mengukur kualitas prediksi dalam mendeteksi objek. AP dihitung dengan menggabungkan presisi dan recall.
5. *True Positives* (TP) adalah jumlah titik data positif yang diidentifikasi dengan benar oleh sistem.
6. *True Negatives* (TN) adalah jumlah titik data negatif yang diidentifikasi dengan benar oleh sistem.
7. *False Positives* (FP) adalah jumlah titik data positif yang diidentifikasi salah oleh sistem.
8. *False Negatives* (FN) adalah jumlah titik data negatif yang diidentifikasi salah oleh sistem. mengukur proporsi prediksi positif dari model yang benar-benar akurat dibandingkan dari jumlah total prediksinya positif.

Hasil dari evaluasi tersebut berfungsi untuk menghitung seberapa baik kinerja model dalam melakukan deteksi objek berbahaya. Jika nilai metrik masih rendah, maka dilakukan perbaikan seperti revisi dataset, penambahan data latih, atau

pengaturan ulang parameter pelatihan agar performa model dapat ditingkatkan secara optimal.

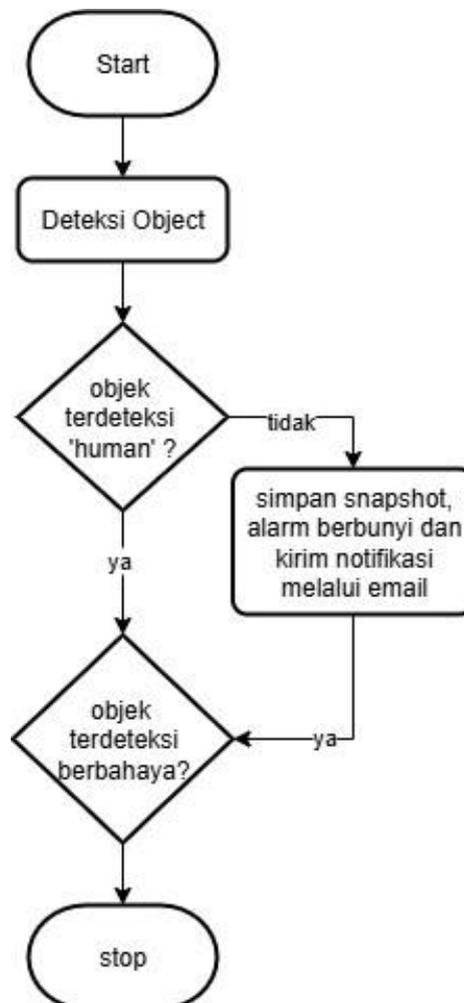
3.3.4 Perancangan Sistem untuk Deteksi Objek

Perancangan sistem deteksi objek pada penelitian ini bertujuan untuk mengembangkan sebuah sistem pengawasan keamanan berbasis kecerdasan buatan yang mampu mendeteksi objek berbahaya secara real-time menggunakan metode YOLOv8. Sistem ini dirancang agar dapat bekerja secara otomatis dan memberikan respons dalam bentuk alarm serta notifikasi kepada pengguna apabila objek yang terklasifikasi sebagai ancaman (seperti manusia, senjata tajam, dan senjata api) terdeteksi dalam area pengawasan. Sistem ini terdiri dari berbagai komponen utama, yaitu :

1. Kamera CCTV digunakan sebagai sumber input berupa video *streaming*.
2. Model deteksi objek YOLOv8 yang telah dilatih menggunakan dataset custom.
3. Antarmuka berbasis web (Flask) untuk menampilkan hasil deteksi secara langsung kepada pengguna.
4. Modul alarm suara (menggunakan *playsound*) sebagai peringatan lokal.
5. Modul notifikasi email sebagai peringatan jarak jauh kepada admin.



Gambar 3. 6 Ilustrasi Penerapan Sistem Deteksi Objek Berbahaya



Gambar 3. 7 Diagram Alir Sistem Deteksi Objek

Pada Gambar 3.6 dan Gambar 3.7, merupakan ilustrasi dan skema sistem ketika dilakukan pengujian secara keseluruhan. Sistem akan menerima input video dari kamera pengawas yang kemudia frame diambil dari video menggunakan OpenCV, selanjutnya frame dikirim ke model YOLOv8 untuk dilakukan proses deteksi. Hasil deteksi divisualisasikan ke dalam frame dengan *bounding box* dan label, ketika objek berbahaya terdeteksi maka frame disimpan dalam folder bukti kemudian fungsi ‘playsound()’ dijalankan untuk menyalakan alarm, dan dilakukan *alert* atau tanda peringatan berupa notifikasi email dengan fungsi ‘send_email_with_image()’ untuk mengirimkan notifikasi beserta bukti gambar hasil deteksi kepada *user* jika terdeteksi objek mencurigakan.

Proses deteksi dimulai dari kamera yang merekam secara real-time dan mengirimkan frame ke server. Frame tersebut akan diproses melalui model YOLOv8 yang sebelumnya telah dilatih dan dimuat melalui pustaka *ultralytics*. Deteksi dilakukan dengan fungsi *model.predict* (source=frame) untuk setiap frame yang masuk.

Tabel 3.4 Kode Pemanggilan Model YOLOv8

Source Code
<pre> from ultralytics import YOLO model = YOLO("best.pt") results = model.predict(source="0", show=True) </pre>

Setiap objek yang terdeteksi nanti ditampilkan didalam frame dengan *bounding box* serta label kelasnya. Jika objeknya berbahaya seperti “gun” atau “knife” terdeteksi, maka sistem secara otomatis menyimpan frame tersebut sebagai bukti

visual dan akan memicu dua respon penting, yaitu pemutaran alarm suara lokal melalui pustaka *playsound*, dan pengiriman email peringatan kepada admin yang disertai dengan gambar hasil tangkapan deteksi.

Tabel 3.5 Kode Fungsi Alarm

Source Code
<pre>def play_alarm(): playsound("alarm.wav")</pre>

Tabel 3.6 Kode Fungsi Pengiriman Email

Source Code
<pre>def send_email_with_image(image_path): msg = EmailMessage() msg['Subject'] = 'Peringatan Deteksi Objek Berbahaya!' msg['From'] = sender msg['To'] = receiver with open(image_path, 'rb') as f: img_data = f.read() msg.add_attachment(img_data, maintype='image', subtype='jpeg', filename=os.path.basename(image_path)) with smtplib.SMTP('smtp.gmail.com', 587) as smtp: smtp.starttls()</pre>

```
smtp.login(sender, password)

smtp.send_message(msg)
```

Bagian backend dari sistem ini dibangun dengan bahasa pemrograman Python menggunakan *framework* Flask. Proses pengambilan gambar dari kamera dilakukan menggunakan pustaka openCV yang menyediakan fungsionalitas untuk menangkap dan membaca video *streaming* secara langsung. Setiap frame dikirim melalui fungsi generator `gen_frames()` dan ditampilkan pada halaman HTML secara dinamis melalui Flask route `/video_feed`. Berikut adalah fungsi `gen_frame()` yang menjalankan proses deteksi dan pengiriman peringatan secara real-time :

Tabel 3.7 Kode Fungsi Deteksi Frame

Source Code
<pre>def gen_frames(): while True: success, frame = camera.read() if not success: break results = model(frame) for r in results: for c in r.bboxes.cls: label = model.names[int(c)] if label in ['knife', 'gun']: cv2.imwrite("deteksi.jpg", frame)</pre>

```

        threading.Thread(target=play_alarm).start()

        send_email_with_image("deteksi.jpg")

    ret, buffer = cv2.imencode('.jpg', frame)
    frame = buffer.tobytes()
    yield (b'--frame\\r\\n'
           b'Content-Type: image/jpeg\\r\\n\\r\\n' + frame + b'\\r\\n')

```

3.3.5 Pengujian Sistem dan Evaluasi

Pada tahap ini, setelah sistem deteksi objek berhasil dirancang dan dikembangkan, tahap selanjutnya adalah melakukan pengujian sistem secara menyeluruh untuk memastikan fungsionalitas dan akurasi dari sistem yang dibangun. Pengujian sistem bertujuan untuk mengetahui apakah sistem mampu mendeteksi objek berbahaya seperti manusia, senjata tajam (knife), dan senjata api (gun) secara real-time dan memberikan respons yang sesuai dalam bentuk alarm dan notifikasi email.

Pengujian dilakukan menggunakan data video yang belum pernah dilakukan pada saat pelatihan (data testing) sebanyak 135 gambar, untuk menghindari *overfitting* dan memberikan hasil pengujian yang objektif. Proses pengujian dilakukan dengan menjalankan file *app.py* yang terhubung langsung dengan model deteksi *best.pt*. Kamera digunakan sebagai input real-time, dan sistem akan mendeteksi keberadaan objek pada setiap frame yang ditangkap.

Pada proses ini, dilakukan pengamatan terhadap kemampuan sistem dalam menampilkan *bounding box* dan label klasifikasi secara akurat, kecepatan sistem

dalam mendeteksi objek dari input video, efektivitas pemicu alarm suara dan pengiriman email saat objek berbahaya terdeteksi, dan penyimpanan otomatis gambar yang disimpan. Selain itu, dilakukan pemantauan performa sistem dalam mengenali objek yang bergerak dan kondisi pencahayaan rendah seperti malam hari. Seluruh hasil pengujian tersebut dicatat dan dianalisis guna mengevaluasi kestabilan sistem dan keandalannya dalam penggunaan jangka panjang. Proses evaluasi ini memperkuat kesimpulan bahwa model deteksi berbasis YOLOv8 yang diimplementasikan telah sesuai untuk digunakan sebagai bagian dari sistem pengawasan keamanan berbasis komputer secara otomatis

Hal ini memungkinkan evaluasi dilakukan secara klasifikasi, yaitu :

1. *Precision*, untuk mengukur akurasi prediksi positif.
2. *Recall*, untuk mengukur seberapa banyak objek aktual yang berhasil dideteksi.
3. *F1-Score*, sebagai rata – rata harmonis dari *precision* dan *recall*.
4. *Mean Average Precision (mAP)*, sebagai indikator keseluruhan kinerja deteksi multi-kelas.

Evaluasi dilakukan terhadap ketiga kelas objek (human, knife, dan gun), dan nilai mAP diperoleh berdasarkan hasil penelitian model YOLOv8. Hasil evaluasi memperlihatkan bahwasannya model dapat mendeteksi objek yang berbahaya dengan tingkat akurasi yang tinggi, kemudian dihasilkan dengan nilai mAP sebesar 0.893 dan *f1-score* sebesar 0.89. Selain evaluasi kuantitatif, dilakukan juga evaluasi kualitatif terhadap tampilan visual deteksi pada antarmuka web dan kecepatan respons sistem saat objek berbahaya muncul.

Dengan hasil tersebut, sistem yang dibangun telah memenuhi tujuan penelitian, yaitu menghasilkan deteksi objek secara real-time yang responsif dan

akurat. Namun, untuk implementasi di dunia nyata, sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur seperti integrasi database, sistem kontrol otomatis, atau peningkatan kamera pengawas.

3.4 Jadwal Penelitian

Waktu pelaksanaan penelitian ini dilakukan dari bulan februari 2025 dapat dilihat pada tabel berikut.

Tabel 3.8 Jadwal Penelitian

NO	NAMA KEGIATAN	BULAN KE																											
		2				3				4				5				6				7							
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
1	Pengajuan Judul	■	■	■	■																								
2	Riset Awal					■	■	■	■																				
3	Pembuatan Proposal									■	■	■	■																
4	Bimbingan Proposal													■	■	■	■												
5	Seminar Proposal																	■											
6	Riset																	■	■	■	■	■	■	■	■				
7	Penyusunan Skripsi																					■	■	■	■	■	■	■	■

BAB IV

HASIL DAN PEMBAHASAN

4.1 Gambaran Umum Sistem

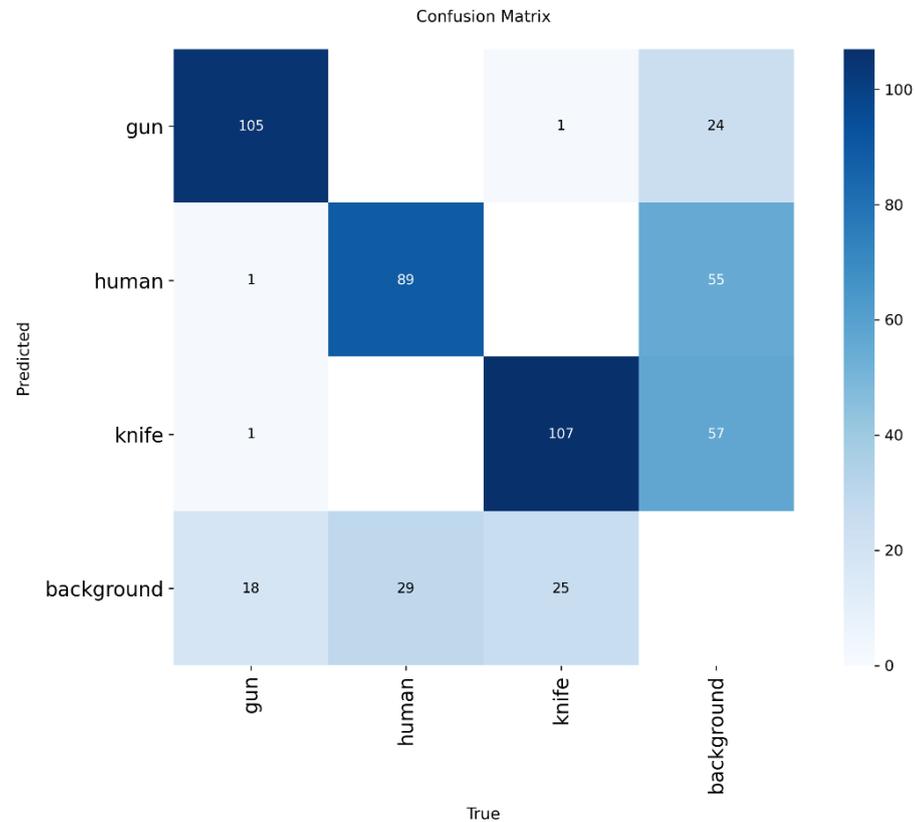
Sistem deteksi objek dalam observasi ini dirancang untuk mendeteksi keberadaan objek – objek yang berpotensi membahayakan seperti manusia (human) membawa senjata tajam (knife), dan senjata api (gun) secara real-time menggunakan metode YOLOv8. Sistem dibangun dengan pendekatan berbasis web menggunakan *framework* Flask, serta terintegrasi dengan kamera CCTV ataupun webcam sebagai sumber input video.

Antarmuka sistem dirancang agar sederhana dan informatif. Pengguna dapat melihat hasil deteksi secara langsung melalui tampilan web, dimana setiap objek yang terdeteksi akan ditampilkan dengan *bounding box* dan label klasifikasi. Jika sistem mendeteksi objek yang termasuk kategori berbahaya, maka secara otomatis sistem akan memicu dua jenis notifikasi yaitu pemutaran alarm serta pengiriman email peringatan kepada admin dilengkapi dengan lampiran gambar hasil deteksi, dimana lampiran tersebut dapat menjadi bukti untuk penyelidikan lebih lanjut.

4.2 Analisa Model Training YOLOv8

Tahapan dalam mengembangkan model YOLOv8 dapat disesuaikan dengan melakukan beberapa kali riset pada dataset tersebut yang dimana telah diproses sebelumnya. Hal ini difokuskan agar menghasilkan konfigurasi dengan hasil kinerjanya yang optimal. Pemantauan eksperimen dijalankan dengan menggunakan *Google Colab*, yang memfasilitasi *tracking* yang rinci dan terstruktur. Dihasilkan metrik evaluasi dan *confusion matrix* dari data validasi yang telah diterapkan pada

model *training*, di bawah ini terdapat gambar 4.1. yang merupakan hasil dari *confusion matrix*.



Gambar 4. 1 Confusion Matrix Hasil Model *Training*

Pada Gambar 4.1, *confusion matrix* menunjukkan model dengan tepat mengklasifikasikan ‘Gun’ yaitu sebanyak 105 kali dari jumlah total 130 anotasi dan ‘Knife’ yaitu 107 kali dari jumlah total 165 anotasi, hal ini memperlihatkan tingkat *true positives* yang tinggi terhadap kedua kelas tersebut. Di satu sisi, ‘Human’ diidentifikasinya dengan benar sebanyak 89 kali dari jumlah total 145 anotasi, merupakan lebih rendah dibandingkan dengan dua kelas lainnya. Sedangkan kelasnya ‘background’ merupakan area yang dihasilkan oleh model YOLOv8, yaitu area non-objek. Nilai ini penting dikarenakan berkorelasi dengan *False Positive* (FP) dan *False Negative* (FN) yang membantu mengevaluasi generalisasi modelnya

secara komprehensif. Kesalahan deteksi yang terlihat dalam *confusion matrix* terjadi beberapa kondisi berikut :

- a. Kelas ‘Gun’ diprediksinya sebagai ‘*background*’ yaitu sebanyak 24 kali.
- b. Kelas ‘Knife’ diprediksinya sebagai ‘*background*’ yaitu sebanyak 57 kali, dan sebagai kelas ‘Gun’ sebanyak 1 kali.
- c. Kelas ‘Human’ diprediksinya sebagai kelas ‘*background*’ yaitu berjumlah 55 kali, dan ‘Knife’ sebanyak 0 kali, namun ada 1 kali salah sebagai ‘Gun’.

Model ini juga identik mengklasifikasikan ‘*background*’ sebagai kelas objeknya, dengan total 18 kali untuk kelas ‘Gun’, sebanyak 29 kali diprediksi sebagai kelas ‘Human’, dan sebanyak 25 kali untuk kelas ‘Knife’. Hal ini menunjukkan bahwa meskipun model telah mencapai performa tinggi dalam mendeteksi objek berbahaya, masih terdapat tingkat kesalahan yang perlu diminimalkan, terutama dalam membedakan area objek dan non-objek (*background*) yang krusial dalam konteks sistem keamanan. Pada hasil analisis *confusion matrix* tersebut, dapat dibandingkan korelasinya dengan nilai metrik evaluasi seperti pada Tabel 4.1.

Tabel 4.1 Hasil Metrik Evaluasi Model *Training*

Kelas	<i>Precision</i>	<i>Recall</i>	mAP50	mAP50-95	<i>F1-score</i>
Gun	87%	81%	91%	68%	84%
Human	86%	61%	85%	54%	71%
Knife	80%	65%	87%	62%	71%
Semua Kelas	85%	69%	88%	61%	75%

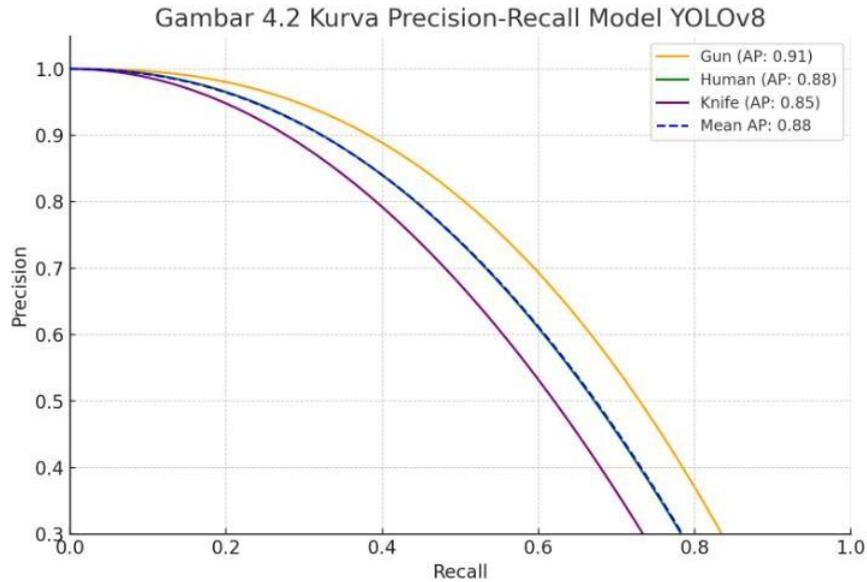
Pada Tabel 4.1, hasil evaluasi dari YOLOv8 menunjukkan *precision* sebesar 85% dengan *recall* sebesar 69% secara rata – rata untuk semua kelas. Hal ini mengindikasikan bahwa model memiliki performa yang baik dalam mengenali objek secara umum. Namun, nilai mAP50-95 sebesar 61% mengindikasikan bahwa

model masih menghadapi tantangan dalam mendeteksi objek secara presisi tinggi, terutama dalam hal penempatan bounding box yang tepat pada berbagai tingkat overlap (IoU 0.50 hingga 0.95). Deteksi 'Human' cukup stabil dengan nilai *precision* 86%, *recall* 61%, dan mAP50-95 sebesar 54%, meskipun *recall* nya lebih rendah dibandingkan dengan kelas 'Gun', tingkat *precision* yang tinggi menunjukkan bahwa model dapat menghindari *false positive* pada deteksi manusia, seperti terlihat pada *confusion matrix* yang menunjukkan 89 prediksi benar dan sedikit kesalahan silang antar kelas.

Sementara itu, deteksi 'Knife' memiliki *recall* 65% namun *precision* 80%, cenderung menghasilkan *false positive* yang lebih tinggi, terutama ketika objek berada di lingkungan kompleks. Hal ini sejalan dengan hasil *confusion matrix*, dimana banyak deteksi 'Knife' yang salah diklasifikasikan sebagai '*background*' sebanyak 57 kali dan 1 kali sebagai 'Gun'. Kesulitan model dalam mencapai akurasi tinggi pada mAP50-95 disebabkan oleh beberapa faktor seperti :

1. Bentuk objek senjata tajam (knife) yang kecil dan sering tertutup sebagian.
2. Variasi pencahayaan yang dapat mengganggu proses deteksi objek.
3. Gangguan latar belakang (*background clutter*) seperti ruangan berantakan atau tangan manusia yang membawa objek lain.
4. Keterbatasan jumlah dan variasi dataset *training*, terutama untuk objek knife yang memiliki bentuk beragam.

Dengan demikian, meskipun performa deteksi secara umum sangat baik, masih dibutuhkan optimalisasi khusus pada kondisi presisi tinggi dan deteksi objek kecil. Selain itu, kurva *precision-recall* seperti pada Gambar 4.2 memberikan pemahaman lebih rinci tentang performa model.



Gambar 4. 2 Kurva *Precision-Recall* Model Training

Dalam Gambar 4.2, kurva *precision-recall* melihat keseimbangan diantara *precision* dan *recall* dalam berbagai nilai *threshold* pada 3 kelas (Gun, Human, Knife) serta rata – rata keseluruhan kelas (Mean AP). Kelas ‘Gun’ memiliki *Average Precision* (AP) tertinggi sebesar 0.91, kemudian ‘Human’ sebesar 0.88, serta ‘Knife’ pada 0.85, dengan mAP terhadap semua kelasnya yaitu berjumlah 0.88. Di awal kurva tersebut menunjukkan nilai *precision* tinggi (1.0) dan *recall* rendah, disimpulkan bahwa modelnya sangat akurat dalam mengidentifikasi kasus positif yang jelas, tetapi *precision* sedikit turun seiring peningkatannya *recall* karena lebih banyak prediksi dipertimbangkan (termasuk yang lebih ambigu). Penurunan *precision* pada *recall* tinggi menunjukkan bahwa model mulai menghasilkan lebih banyak *false positives* saat mencoba menangkap semua objek secara agresif. Hal ini umum dalam model deteksi objek dan merupakan *trade-off* yang wajar. *Area Under Curve* (AUC) untuk masing – masing kelas mencerminkan kemampuan model membedakan objek dengan baik, bentuk

kurva yang mendekati sudut dari kanan atas memperlihatkan bahwasannya model tersebut memiliki performa deteksi yang sangat baik secara keseluruhan.

4.3 Analisis Testing Model Pada Video

Pengujian model YOLOv8 tidak hanya dilakukan dalam tahap *training* dan validasi, tetapi juga diuji langsung pada video pengujian untuk mengevaluasi performa deteksi dalam kondisi nyata. Pengujian ini penting untuk mengetahui keahlian dari generalisasi modelnya terhadap data baru yang bahkan tidak pernah dilihat sebelumnya.

Model *best.pt* yang telah dihasilkan dari proses training diintegrasikan ke dalam sistem real-time berbasis Flask dan OpenCV, seperti yang terlihat dalam file implementasi *app.py*. Sistem dirancang untuk membaca frame dari video (atau kamera CCTV) dan mendeteksi objek secara langsung dengan menggunakan model YOLOv8.

4.3.1 Hasil Deteksi Objek dalam Video

Pengujian dilakukan pada video berdurasi pendek kurang lebih 1 menit, yang memuat berbagai skenario seperti seseorang yang memegang senjata tajam (knife), anak – anak di dekat objek yang menyerupai senjata api (gun), orang dewasa berada dalam posisi normal (tanpa objek berbahaya). Dari hasil pengujian, ditentukan sebagai berikut :

1. Model berhasil mendeteksi objek ‘Gun’ dan ‘Knife’ secara konsisten di hampir semua frame.
2. *Bounding box* muncul dengan presisi posisi yang baik, tanpa loncatan atau error deteksi.
3. Objek ‘Human’ selalu terdeteksi, baik dalam posisi berdiri maupun duduk.

4. Ketika dua objek berbahaya terdeteksi bersamaan dalam satu frame, sistem secara otomatis menyimpan *screenshot* hasil deteksi ke dalam folder *snapshots/*, memutar suara alarm (*alarm.wav*), dan mengirimkan email notifikasi ke admin (fitur tersebut ada pada sistem *app.py*)

4.3.2 Dokumentasi Visual Hasil Uji Video

Dokumentasi visual pada hasil uji coba sistem deteksi objek secara real-time dapat dilihat pada tampilan gambar 4.3 dibawah ini .



Gambar 4. 3 Deteksi Objek ‘Knife’ dan ‘Gun’

Pada Gambar 4.3 menunjukkan hasil tangkapan layar sistem ketika mendeteksi dua objek berbahaya secara bersamaan dalam satu frame video. Dalam gambar tersebut, model berhasil mengenali objek ‘Knife’ dengan *confidence* 0.93 dan ‘Gun’ dengan *confidence* 0.74. Deteksi dilakukan secara real-time dari video input, dan hasil *bounding box* ditampilkan secara akurat disekitar objek yang relevan. Keberhasilan ini membuktikan bahwa model mampu bekerja dengan baik

dalam lingkungan dinamis, bahkan saat objek dikelilingi manusia atau latar kompleks. Ketika dua objek berbahaya terdeteksi dalam satu frame, sistem secara otomatis menjalankan serangkaian aksi yaitu alarm peringatan diputar, *screenshot* disimpan ke direktori, dan sistem siap untuk mengirim notifikasi jika fitur tersebut diaktifkan. Deteksi ini menjadi bukti bahwa sistem tidak hanya mengenali objek dengan baik, tetapi juga mampu mengeksekusi tindakan pengamanan secara otomatis sebagai respons terhadap potensi ancaman.

4.3.3 Evaluasi Respons Sistem

Dalam menilai sejauh mana efektivitas sistem dalam situasi nyata, dilakukan evaluasi terhadap kinerja sistem secara keseluruhan berdasarkan hasil pengujian pada video. Evaluasi ini mencakup kecepatan respons, keakuratan deteksi, serta kemampuan sistem dalam mengeksekusi aksi otomatis seperti pemutaran alarm, pengambilan *screenshot*, dan pengiriman notifikasi. Hasil pengujian ini dipaparkan pada tabel 4.2 berikut.

Tabel 4.2 Hasil Evaluasi Respons Sistem pada Pengujian Video

Aspek Evaluasi	Hasil
Kecepatan Deteksi	12-15 frame per detik (fps)
Waktu Respons Alarm	< 1 detik setelah deteksi ≥ 2 objek berbahaya
Keakuratan Bounding Box	Tepat di sekitar objek senjata dan manusia
Snapshot Otomatis	Berhasil disimpan pada setiap deteksi ganda
Email Notifikasi	Terkirim ke alamat admin dengan gambar terlampir

Pada Tabel 4.2 menunjukkan hasil evaluasi sistem saat dijalankan untuk mendeteksi objek berbahaya dalam video. Kecepatan deteksi berkisar antara 12 hingga 15 frame per detik (fps) menunjukkan bahwa sistem mampu beroperasi

dalam mode real-time. Hal ini sangat penting dalam konteks pengawasan keamanan, dimana respons cepat terhadap ancaman sangat dibutuhkan. Dalam hal lain, waktu respons sistem terhadap situasi kritis, seperti keberadaan dua objek berbahaya dalam satu frame, berada dibawah 1 detik. Hal ini menunjukkan bahwa mekanisme alarm dan pengembalian keputusan berjalan efektif. Akurasi penempatan *bounding box* juga sangat baik, dengan hasil yang tepat di sekitar objek senjata dan manusia. Proses penyimpanan snapshot dan pengiriman notifikasi email juga berhasil dilakukan dengan konsisten sesuai skenario pengujian, menandakan sistem telah berjalan stabil dan responsif.

4.4 Analisis Performa Sistem Deteksi Objek

Cara sistem menemukan orang dan senjata secara real-time sebagian besar bergantung pada jenis model yang digunakan dan seberapa banyak pekerjaan yang harus dilakukan sekaligus. Saat menguji sistem dalam berbagai kondisi cahaya, kami mengamati seberapa baik kinerjanya untuk setiap jenis objek. Kemudian, kami menghitung rata-rata untuk melihat kinerjanya secara keseluruhan dalam semua situasi pencahayaan yang berbeda, yang ditunjukkan pada Tabel 4.3 di bawah ini.

Tabel 4.3 Hasil Evaluasi Performa Sistem Deteksi Objek

Kelas	Precision	Recall	<i>F₁-score</i>
Gun	0.88	0.87	0.88
Human	0.91	0.93	0.92
Knife	0.84	0.82	0.83
Semua Kelas	0.88	0.87	0.87

Pada Tabel 4.3, terlihat bahwa secara keseluruhan, pada kasus pengujian sistem deteksi, kelas 'Human' memiliki nilai *F₁-score* yang tertinggi yaitu 0.92, menunjukkan bahwa model sangat baik dalam mengenali objek manusia.

Sebaliknya, kelas dengan kinerja terendah ada pada ‘Knife’ dengan F_1 -score 0.83. Perihal ini memperlihatkan bahwasannya model memiliki kesulitan dalam membedakan objek ‘Knife’ dan background, terlebih ketika objek kecil, tertutup sebagian, atau berada pada pencahayaan rendah.

Rendahnya performa deteksi pada kelas ‘Knife’ mempengaruhi rata – rata nilai evaluasi sistem, dengan nilai F_1 -score keseluruhan sebesar 0.87. Penyebab utamanya ialah banyak kasus dari *False Positive* dan *False Negative* yang terjadi, terutama dalam pengujian malam hari. Contoh kasus *false positive* pada sampel gambar pengujian langsung ditampilkan pada Gambar 4.4 dibawah ini.



Gambar 4.4 Contoh Kasus *False Positive*

Pada Gambar 4.4, merupakan contoh dari kasus *false positive* yang terjadi ketika pengujian sedang di uji langsung. Pada objek manusia yang sedang membawa senjata api, sering kali prediksi yang muncul bersamaan dengan terdeteksi objek *knife* pada *background* yang bukan merupakan objek dari senjata tajam. Kasus *false positive* pada kelas ‘Knife’ sering terjadi , yang bukan merupakan target penelitian sistem deteksi objek ini. Hal serupa juga terjadi pada

kelas 'Gun', dimana mencakup kesulitan membedakan antara mainan atau benda panjang dengan senjata tajam. Daripada itu, artefak lainnya di lingkungan *outdoor* atau publik, seperti benda memantul atau logam, mempengaruhi deteksi pada kelas 'Knife' dan 'Gun'.

Permasalahan ini dapat disebabkan oleh pencahayaan serta dataset yang kurang representatif, dan menunjukkan bahwa sistem masih dapat memberikan deteksi yang keliru, terutama ketika nilai confidence berada di bawah ambang batas optimal sehingga sistem memperkirakan objek tersebut sebagai kelas yang dideteksi. Sedangkan contoh *false negative* pada pengujian langsung terdapat pada Gambar 4.5.



Gambar 4.5 Contoh Kasus *False Negative*

Pada Gambar 4.5 merupakan contoh dari kasus *false positive* yang terjadi ketika pengujian sedang di uji langsung. Kasus *false negative* ini lebih sering melewatkan deteksi kelas 'Gun' dan 'Knife' dibandingkan dengan kelas 'Human'.

Pada pengujian, kondisi pencahayaan rendah dan sudut pengambilan gambar yang tidak optimal turut memengaruhi keberhasilan sistem dalam mendeteksi objek secara akurat. Pada gambar tersebut, objek 'Knife' di tangan orang dewasa tidak terdeteksi sama sekali meskipun terlihat jelas. Ini menunjukkan bahwa sistem terkadang melewatkan deteksi objek berbahaya, terutama jika bentuk atau posisi objek tidak dominan dalam frame.

4.5 Implementasi Antarmuka WEB dan Backend

Dalam implementasi sistem deteksi objek berbahaya berbasis YOLOv8 ini, pengembang tidak hanya berfokus pada akurasi model deteksi, tetapi juga memperhatikan kemudahan pengguna dalam mengakses dan memantau hasil deteksi melalui antarmuka web. Oleh karena itu, sistem ini dirancang dengan pendekatan full-stack sederhana, yakni backend Python berbasis Flask dan antarmuka frontend berbasis HTML dan CSS.

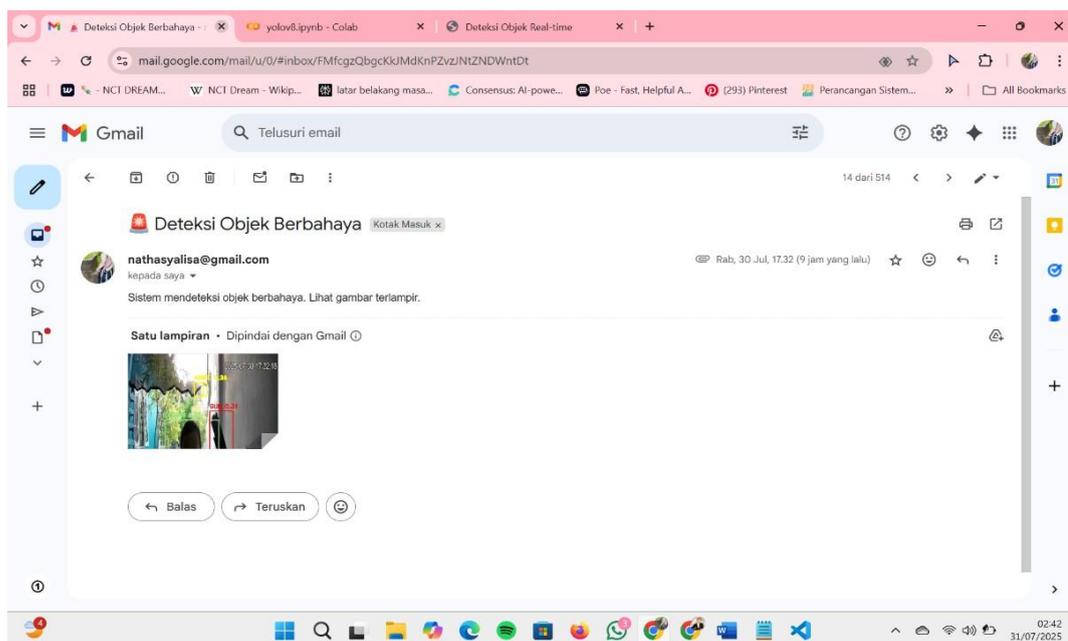
4.5.1 Integrasi Sistem

Integrasi *backend* dan *frontend* dilakukan melalui *framework* Flask sebagai server utama. Pada bagian *backend*, sistem mengatur alur kerja utama mulai dari akses kamera (webcam maupun RTSP CCTV), deteksi objek menggunakan model YOLOv8 yang telah dilatih, hingga eksekusi logika lanjutan seperti pengaktifan alarm, penyimpanan gambar snapshot, dan pengiriman notifikasi email.

Saat tombol pada antarmuka di klik, request akan diteruskan ke *backend* Flask untuk memproses input kamera. Frame yang masuk akan melalui fungsi `detect_object()` yang secara real-time menganalisis gambar menggunakan model YOLO dan menghasilkan *bounding box* beserta label objek. Jika objek berbahaya

seperti knife atau gun terdeteksi, *backend* langsung memicu fungsi `alarm_on()`, `save_snapshot()`, dan `send_email()` secara otomatis.

Implementasi ini juga memanfaatkan pustaka seperti OpenCV untuk mengakses kamera dan pengolahan gambar, serta *threading* untuk memastikan sistem tidak mengalami *bottleneck* saat proses deteksi berjalan bersamaan dengan streaming video. Setelah objek berbahaya seperti *gun* atau *knife* terdeteksi, sistem secara otomatis akan menyimpan tangkapan layar (*snapshot*) dan mengirimkan email notifikasi ke alamat yang telah ditentukan. Email tersebut dikirim dengan subjek "Deteksi Objek Berbahaya" dan dilengkapi dengan gambar tangkapan layar dari kamera sebagai lampiran.



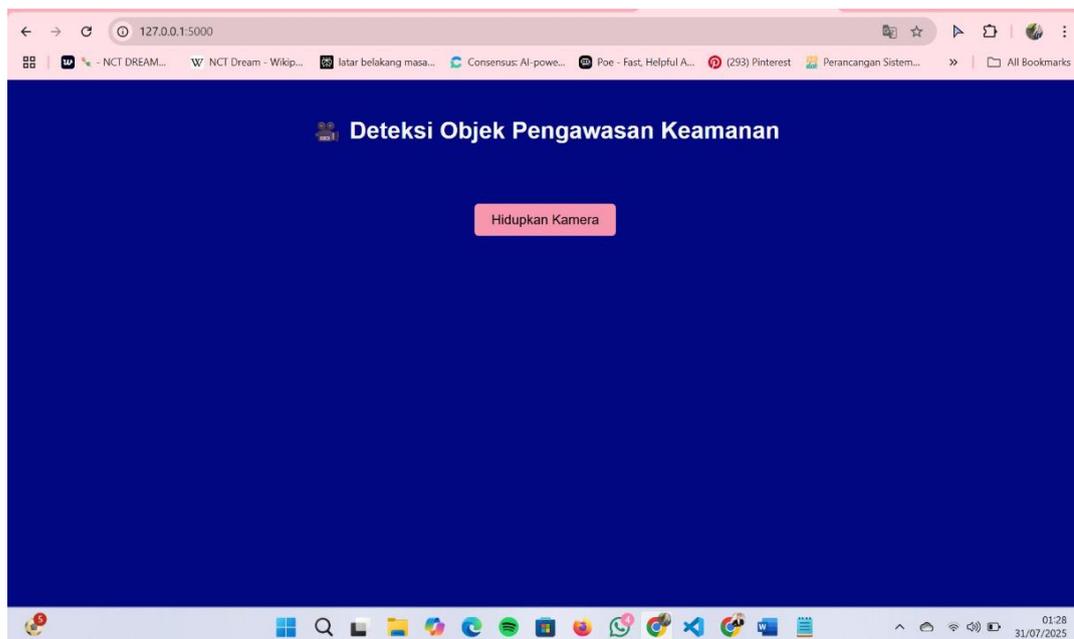
Gambar 4.6 Tampilan Notifikasi Email Admin

Pada Gambar 4.6 menunjukkan tampilan email yang diterima pengguna setelah sistem mendeteksi objek berbahaya. Tampilan ini membuktikan bahwa integrasi sistem tidak hanya bekerja secara lokal, tetapi juga mampu memberikan

notifikasi secara real-time ke jarak jauh melalui jaringan internet. Berdasarkan hasil uji coba, fitur pengiriman email ini berjalan dengan baik dan berhasil mengirimkan peringatan dalam waktu relatif cepat setelah deteksi dilakukan. Hal ini menunjukkan bahwa sistem dapat bekerja secara responsif dalam kondisi nyata. Dengan adanya notifikasi email mengenai situasi berbahaya ini dapat menjadi bukti ancaman yang terjadi di lingkungan tersebut, juga memudahkan penyelidikan lebih lanjut oleh pihak yang berwenang.

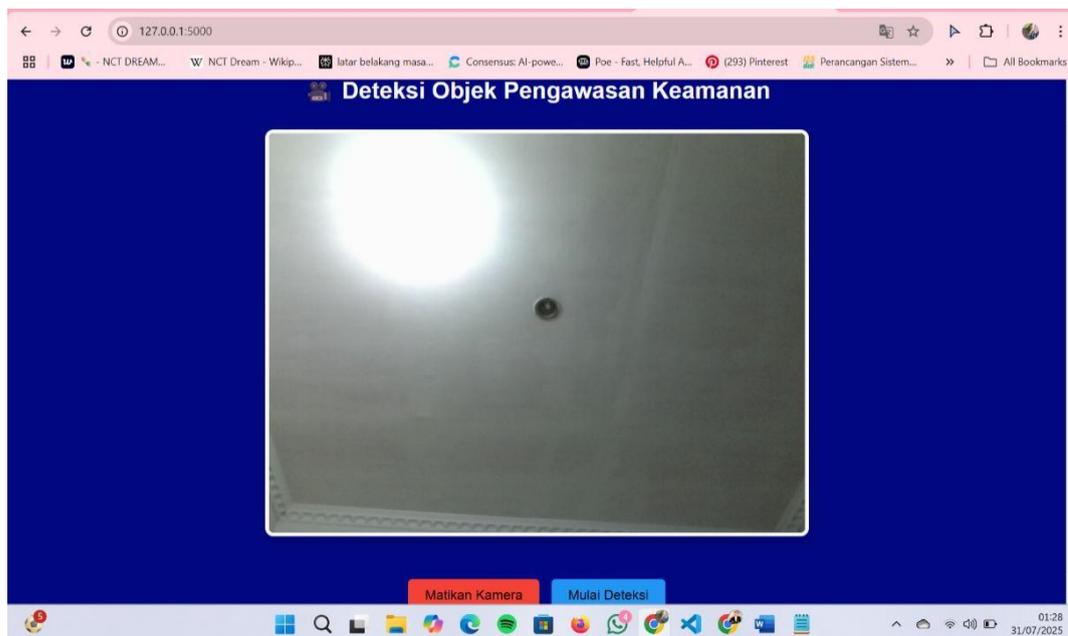
4.5.2 Tampilan Antarmuka WEB

Tampilan antarmuka web dirancang sederhana dan *user-friendly* menggunakan HTML dan CSS. Pengguna akan langsung diarahkan ke halaman utama dengan judul “Deteksi Objek Pengawasan Keamanan”, disertai ikon kamera yang relevan. Di halaman ini terdapat tombol “Hidupkan Kamera”, yang jika ditekan akan membuka *streaming* dari kamera pengawas. Berikut tampilan antarmuka sistem ketika belum dan sesudah kamera diaktifkan.



Gambar 4.7 Halaman Utama Web

Setelah kamera aktif, antarmuka menampilkan jendela video live dari kamera. Dibawahnya terdapat dua tombol tambahan yaitu “Matikan Kamera” dan “Mulai Deteksi”. Tombol tersebut berfungsi untuk mengontrol alur operasional sistem secara manual. Tombol – tombol ini dibuat dengan desain warna kontras (merah dan biru) agar mudah dikenali oleh pengguna. Tampilan antarmuka ini sangat membantu dalam pemantauan, terutama untuk pengguna non-teknis yang tidak familiar dengan jalur terminal atau *script*. Semua proses deteksi dapat dilakukan melalui klik tanpa perlu mengetik perintah apapun.



Gambar 4. 8 Tampilan Kamera Deteksi Web

Dengan adanya antarmuka ini, sistem menjadi lebih mudah digunakan dan dapat dijalankan di lingkungan lokal serta mampu melakukan deteksi secara real-time dengan visualisasi yang jelas. Hal ini membuktikan bahwa implementasi backend dan frontend berhasil terintegrasi dengan baik, mendukung tujuan sistem sebagai alat bantu pengawasan keamanan secara langsung.

4.6 Ujicoba Sistem dengan Kamera CCTV

Uji coba sistem menggunakan kamera CCTV dilakukan sebagai bentuk validasi akhir terhadap kinerja sistem secara nyata di lingkungan langsung (real-world scenario). Dalam implementasinya, sistem berbasis Flask diintegrasikan dengan sumber video streaming dari kamera CCTV menggunakan protokol RTSP. Pengujian dilakukan di lingkungan indoor dan outdoor, baik pada siang hari maupun malam hari, untuk mengevaluasi seberapa responsif dan akurat sistem dalam kondisi pencahayaan dan latensi jaringan yang bervariasi. Sistem mendeteksi objek secara langsung, dan jika teridentifikasi adanya objek berbahaya (knife atau gun), maka alarm berbunyi otomatis, sistem menyimpan tangkapan layar ke dalam direktori static/snapshot/ , dan fitur email dapat mengirimkan notifikasi.

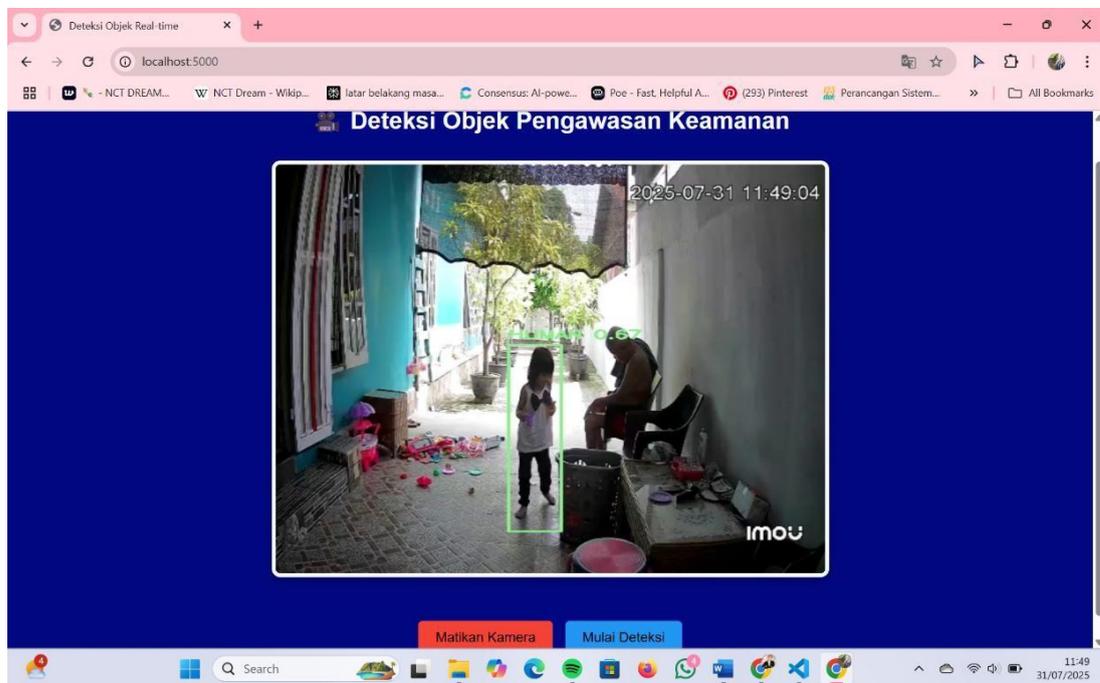
Pada siang hari, sistem menunjukkan performa yang optimal. Deteksi objek berjalan mulus dengan tajam dan bounding box sesuai dengan letak objek nyata. Tidak ditemukan keterlambatan berarti, dan semua tangkapan layar berhasil tersimpan secara otomatis. Namun pada malam hari, performa sistem menurun CCTV yang tidak dilengkapi pencahayaan tambahan mengakibatkan video menjadi gelap, dan hal ini menyebabkan confidence deteksi turun ke bawah 0.5 bahkan hingga tidak mendeteksi sama sekali. Dalam beberapa kasus, objek seperti knife tidak terdeteksi padahal terlihat samar oleh manusia.

Rata – rata waktu tanggap sistem dari deteksi hingga alarm berbunyi berada di kisaran 1.2 detik, yang masih dapat diterima dalam konteks sistem pengawasan cepat. Proses snapshot berjalan rata – rata 0.8 detik setelah deteksi berhasil, dan ukuran file hasil snapshot cukup kecil untuk dikirimkan melalui email. Berikut

rangkuman hasil pengujian sistem dengan kamera CCTV berdasarkan waktu dan kondisi pencahayaan yang terdapat pada tabel 4.4 berikut.

Tabel 4.4 Hasil Uji Sistem dengan Kamera CCTV

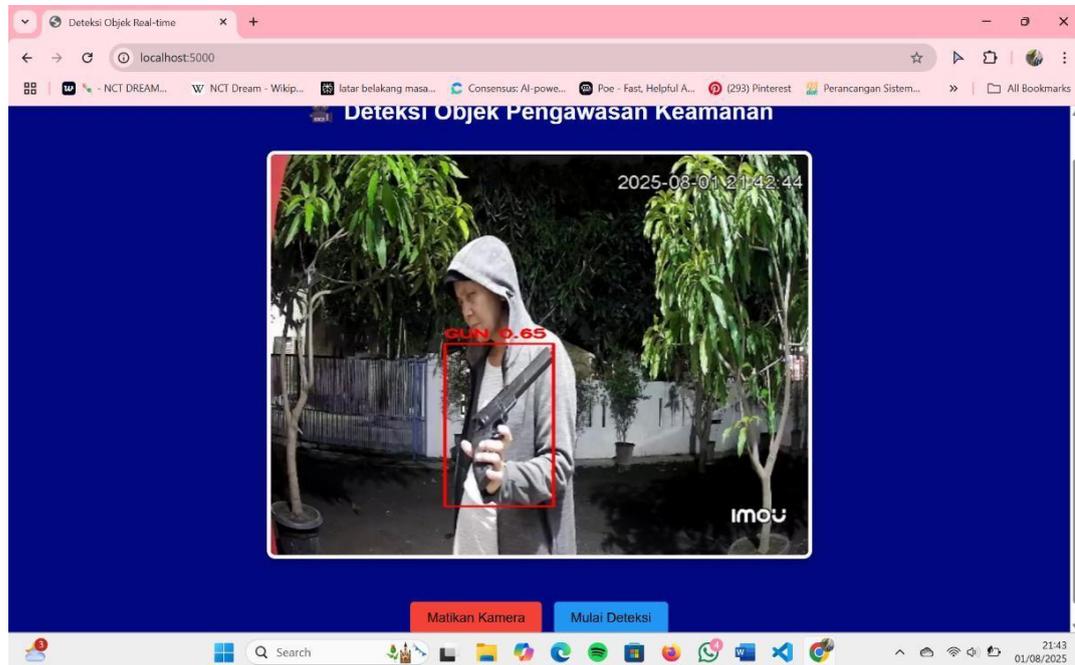
Waktu Pengujian	Lokasi	Pencahayaan	Deteksi Objek	Rata-Rata Confidence	Alarm	Snapshot
Siang Hari	Indoor	Terang	Terdeteksi	0.85 – 0.93	Aktif	Berhasil
Siang Hari	Outdoor	Cukup terang	Terdeteksi	0.82 – 0.89	Aktif	Berhasil
Malam Hari	Indoor	Redup	Terkadang	0.40 – 0.60	Lambat	Kadang
Malam Hari	Outdoor	Gelap	Tidak Terdeteksi	< 0.35	Tidak	Tidak



Gambar 4. 9 Hasil Ujicoba Sistem di Siang Hari

Secara keseluruhan, sistem telah berhasil diuji dengan baik menggunakan kamera CCTV sebagai sumber input langsung. Pengujian ini membuktikan bahwa sistem mampu bekerja dalam kondisi nyata, meskipun masih terdapat batasan pada pencahayaan malam hari. Performa sistem dapat ditingkatkan dengan

menambahkan pencahayaan tambahan dan kamera CCTV berkualitas lebih tinggi untuk menunjang sistem deteksi objek secara optimal dalam semua kondisi.



Gambar 4. 10 Hasil Ujicoba pada Malam Hari

Pada Gambar 4.10 merupakan hasil tangkapan layar dari antarmuka sistem deteksi objek *real-time*, meskipun pencahayaan di lingkungan luar (*outdoor*) cenderung redup, sistem masih mampu mendeteksi objek berbahaya secara cukup akurat. Hal ini menunjukkan bahwa model YOLOv8 yang digunakan masih bekerja cukup efektif meskipun dalam pencahayaan rendah. *Background* yang gelap dan kontras terhadap warna objek membantu sistem dalam membedakan objek dengan lebih baik. Namun, nilai *confidence* 0.65 juga menunjukkan bahwa pada kondisi malam hari, akurasi deteksi bisa sedikit menurun dibanding dengan siang hari, yang biasanya menghasilkan *confidence* di atas 0.80.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melihat semua langkah dalam penelitian sebelumnya, diperoleh beberapa hasil yang dapat disimpulkan yaitu sebagai berikut :

1. Sistem pendeteksi objek berbasis YOLOv8 dibangun dengan memiliki kemampuan mendeteksi objek berbahaya yang berupa senjata tajam (knife), senjata api (gun), dan manusia (human) secara real-time dari input kamera CCTV, serta memberikan respons berupa alarm dan pengiriman email kepada pengguna sebagai bentuk notifikasi otomatis.
2. Hasil evaluasi performa model memperlihatkan bahwasannya sistem memiliki tingkat akurasi yang baik, dengan nilai *Precision* 0.88, *Recall* 0.87, dan *F1-score* 0.87 secara rata-rata. Kinerja terbaik diperoleh pada kelas *Human* (*F1-score* 0.92), sedangkan kinerja terendah pada kelas *Knife* (*F1-score* 0.83), khususnya pada kondisi pencahayaan redup.
3. Pengujian sistem menunjukkan bahwa pada siang hari, deteksi berlangsung stabil dengan confidence tinggi (rata-rata ≥ 0.85), sedangkan pada malam hari, sistem tetap dapat mendeteksi objek namun dengan tingkat confidence yang lebih rendah dan waktu respons alarm yang lebih lambat.
4. Sistem ini telah berhasil diimplementasikan secara utuh dengan antarmuka berbasis web menggunakan Flask, mendukung pengawasan keamanan secara otomatis dan dapat dijadikan dasar pengembangan lebih lanjut untuk aplikasi keamanan real-time berbasis kecerdasan buatan.

5.2 Saran

Berdasarkan hasil pengujian dengan metode YOLO (*You Only Look Once*), terdapat beberapa saran yang dapat diimplementasikan agar kedepannya meningkatkan efektivitas program. Setelah diperoleh kesimpulan maka beberapa saran yang bisa dijadikan sebagai masukan yaitu :

1. Dataset dapat ditingkatkan dari segi jumlah dan keragaman gambar untuk mengurangi kemungkinan deteksi salah (*False Detection*), terutama pada kondisi pencahayaan rendah atau objek tertutup sebagian.
2. Pengembangan fitur notifikasi tambahan seperti SMS, atau integrasi ke dashboard pemantauan dapat menambah fungsionalitas dan kemudahan dalam pengawasan *real-time*.
3. Penggunaan perangkat keras yang lebih baik seperti GPU lokal dapat meningkatkan kecepatan inferensi sistem dan mengurangi *delay* saat proses deteksi.
4. Pengembangan lanjutan pada sistem dapat dikombinasikan dengan teknologi pelacakan objek (*object tracking*), atau analisis perilaku (*behavior analysis*) untuk meningkatkan kapabilitas pengawasan.

DAFTAR PUSTAKA

- Andreas Restu Priatama. (2023). Deteksi Objek Pisau Menggunakan Yolo Machine Learning. *PROKASDADIK: Prosiding Kecerdasan Artifisial, Sains Data, Dan Pendidikan Masa Depan*, 1(1), 56–61.
- Batubara, Z. H., Hamonangan, Y., Arfan, M., & Hidayatno, A. (2024). Perancangan Sistem Deteksi Pelanggaran Penggunaan Helm Dengan Metode Deep Learning Menggunakan Yolov5 Ultralytic. *Transient: Jurnal Ilmiah Teknik Elektro*, 13(1), 11–20. <https://doi.org/10.14710/transient.v13i1.11-20>
- Dahlan, I. A., Faculty, M. E., Versantariqh, M. A., Faculty, M. E., Ariateja, D., Faculty, M. E., David, M., Faculty, M. E., Arghanie, M. A., Faculty, M. E., Fatmawati, U. D., & Faculty, M. E. (2021). *Sistem Deteksi Senjata Otomatis Menggunakan Deep Learning Berbasis CCTV Cerdas*. 04(02), 126–141.
- He, J., & Eddie Law, K. L. (2024). Deep Learning Models for Rotated Object Detection in Aerial Images: Survey and Performance Comparisons. *IEEE Access*, 12(November), 180436–180457. <https://doi.org/10.1109/ACCESS.2024.3509745>
- Hidayatulloh, M. S. (2021). Sistem Pengenalan Wajah Menggunakan Metode Yolo (You Only Look Once). *Universitas Medika*, i–43.
- Manurung, D. G., Pinasthika, M. R., Vasya, M. A. O., Putri, R. A. D. S., Tampubolon, A. P., Prayata, R. F., Nisa, S. K., & Yudistira, N. (2024). Deteksi Dan Klasifikasi Hama Potato Beetle Pada Tanaman Kentang Menggunakan YOLOV8. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 11(4), 723–734. <https://doi.org/10.25126/jtiik.1148092>

- Marpaung, F., Aulia, F., & Nabila, R. C. (2022). *Computer Vision Dan Pengolahan Citra Digital*. www.pustakaaksara.co.id
- Muhamad Zulkarnaen, F. (2024). *Sistem Deteksi Objek Manusia Menggunakan Algoritma Yolov8 Berbasis Kamera Depth Sensor (Studi Kasus: Cv. Ateri Global Teknologi)*.
- Nur, T., Huzaeni, & Khadafi, M. (2023). Implementasi Metode Object Detection Dengan Algoritma Yolo (You Only Look Once) Untuk. *Jurnal Teknologi Rekayasa Informasi Dan Komputer*, 6(2), 28–33.
- Penulisan, S., & Pustaka, L. T. (2023). *Deteksi Kematangan Buah Tomat Berbasis Two-Layers You Only Look Once (YOLOv8) Pada Green-House Monitoring Robots Ahmad Bayu Pamungkas, Jans Hendry, S.T., M.Eng.*
- Prabowo, D. A., & Abdullah, D. (2018). Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan Color Object Tracking. *Pseudocode*, 5(2), 85–91. <https://doi.org/10.33369/pseudocode.5.2.85-91>
- Pratiwi, N. (2025). *Deteksi manusia dengan algoritma yolo untuk pemutaran audio otomatis di area tertentu*. 10(1), 667–674.
- Rahayu, M. I., Rizaludin, M., & Jayusman, Y. (2025). *Sistem Presensi menggunakan Deteksi Objek Wajah Mahasiswa Berbasis YOLO-V5*. October 2024. <https://doi.org/10.52771/bangkitindonesia.v13i2.310>
- Robots, M., Pamungkas, A. B., & Hendry, J. (2023). *Deteksi Kematangan Buah Tomat Berbasis Two-Layers You Only Look Once (YOLOv8) Pada Green-House Monitoring Robots Ahmad Bayu Pamungkas, Jans Hendry, S.T., M.Eng.*
- Sunarto, A. A., Informatika, T., Sukabumi, U. M., Kematangan, K., Learning, D.,

- & Super, K. (2024). *PENERAPAN YOU ONLY LOOK ONCE (YOLO) V8 UNTUK*. 8(5), 10566–10571.
- Surya, M. A., Susanto, M., Setyawan, A., Fitriawan, H., & Mardiana. (2024). Sistem Keamanan Ruang Dengan Human Detection Menggunakan Sensor Kamera Berbasis Deep Learning. *Jurnal Teknoinfo*, 18(1), 182–192.
<https://ejournal.teknokrat.ac.id/index.php/teknoinfo/index>
- Triyanto, D., Zidan, M., Wahyudi, M., & Pujiastuti, L. (2024). *Pengembangan Sistem Deteksi Objek Botol Real-Time dengan YOLOv8 untuk Aplikasi Vision*. 3(1).
- Umar, R., Riadi, I., & Purwono, P. (2020). Klasifikasi Kinerja Programmer pada Aktivitas Media Sosial dengan Metode Support Vector Machines. *Cybernetics*, 4(01), 32. <https://doi.org/10.29406/cbn.v4i01.2042>
- Virgiawan, I., Maulana, F., Putra, M. A., Kurnia, D. D., & Sinduningrum, E. (2024). Deteksi dan tracking objek secara real-time berbasis computer vision menggunakan metode YOLO V3. *Jurnal Ilmiah Multidisiplin Indonesia*, 3(3). <https://journal.ikopin.ac.id/index.php/humantech>
- Zophie, J., & Triharminto, H. H. (2020). *Implemetasi Algoritma You Only Look Once (YOLO) menggunakan Web Camera untuk Mendeteksi Objek Statis dan Dinamis Implementation of You Only Look Once (YOLO) Algorithm using Web Camera for Static dan Dinamic Object Detection*. 1(1), 98–109.

LAMPIRAN

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size	640: 100%	235/235	[00:45<00:00, 5.13it/s]	all
1/100	1.09G	1.254	2.599	1.539	24	640: 100%	MAP50	MAP50-95): 100%	235/235 [00:45<00:00, 5.13it/s]	all
	Class	Images	Instances	Box(P	R	MAP50	MAP50-95): 100%	17/17	[00:03<00:00, 4.76it/s]	
2/100	1.34G	1.388	2.411	1.641	10	640: 100%	MAP50	MAP50-95): 100%	235/235 [00:40<00:00, 5.78it/s]	all
	Class	Images	Instances	Box(P	R	MAP50	MAP50-95): 100%	17/17	[00:02<00:00, 7.54it/s]	
3/100	1.35G	1.536	2.473	1.758	18	640: 100%	MAP50	MAP50-95): 100%	235/235 [00:40<00:00, 5.81it/s]	all
	Class	Images	Instances	Box(P	R	MAP50	MAP50-95): 100%	17/17	[00:02<00:00, 6.08it/s]	
	all	268	363	0.373	0.255	0.236	0.106			
4/100	1.37G	1.524	2.402	1.76	31	640: 100%	MAP50	MAP50-95): 100%	235/235 [00:40<00:00, 5.86it/s]	all
	Class	Images	Instances	Box(P	R	MAP50	MAP50-95): 100%	17/17	[00:02<00:00, 6.71it/s]	
5/100	1.39G	1.473	2.272	1.717	13	640: 100%	MAP50	MAP50-95): 100%	235/235 [00:40<00:00, 5.86it/s]	all
	Class	Images	Instances	Box(P	R	MAP50	MAP50-95): 100%	17/17	[00:02<00:00, 7.64it/s]	
6/100	1.4G	1.433	2.18	1.688	9	640: 100%	MAP50	MAP50-95): 100%	235/235 [00:39<00:00, 5.95it/s]	all
	Class	Images	Instances	Box(P	R	MAP50	MAP50-95): 100%	17/17	[00:02<00:00, 7.32it/s]	
7/100	1.42G	1.392	2.077	1.654	16	640: 100%	MAP50	MAP50-95): 100%	235/235 [00:39<00:00, 5.98it/s]	all
	Class	Images	Instances	Box(P	R	MAP50	MAP50-95): 100%	17/17	[00:02<00:00, 7.44it/s]	
	all	268	363	0.39	0.448	0.334	0.159			
8/100	1.44G	1.369	1.989	1.641	19	640: 100%	MAP50	MAP50-95): 100%	235/235 [00:40<00:00, 5.87it/s]	all
	Class	Images	Instances	Box(P	R	MAP50	MAP50-95): 100%	17/17	[00:02<00:00, 5.79it/s]	
9/100	1.46G	1.339	1.963	1.608	13	640: 100%	MAP50	MAP50-95): 100%	235/235 [00:39<00:00, 5.93it/s]	all
	Class	Images	Instances	Box(P	R	MAP50	MAP50-95): 100%	17/17	[00:02<00:00, 7.66it/s]	

```

1 from flask import Flask, render_template, Response, request
2 import cv2
3 from ultralytics import YOLO
4 import smtplib, ssl
5 from email.mime.text import MIMEText
6 from email.mime.multipart import MIMEMultipart
7 from email.mime.base import MIMEBase
8 from email import encoders
9 from datetime import datetime
10 import os
11 import threading
12 from playsound import playsound
13 import time
14
15 app = Flask(__name__)
16
17 # ===== LOAD YOLOv8 MODEL =====
18 model = YOLO("best.pt") # pastikan best.pt ada di Folder
19 class_names = model.names
20
21 # ===== LABEL COLOR MAPPING =====
22 colors = {
23     'gun': (0, 0, 255), # red
24     'human': (144, 238, 144), # light green
25     'knife': (0, 255, 255) # yellow
26 }
27
28 # ===== EMAIL CONFIGURATION =====
29 EMAIL_SENDER = 'nathasyalisa@gmail.com'
30 EMAIL_RECEIVER = 'nathasyalisa@gmail.com'
31 EMAIL_PASSWORD = 'fvptozzmlcebwzh'
32
33 # ===== ALARM & SNAPSHOT CONFIG =====
34 ALARM_PATH = 'alarm.wav'

```

The screenshot shows a VS Code editor window titled 'yolo_flask_project'. The Explorer sidebar on the left shows a project structure with folders like '.venv', 'Include', 'Lib', 'Scripts', and files like 'app.py', 'app2.py', 'best.pt', 'cek_model.py', 'deteksi_log.txt', 'README.txt', and 'requirements.txt'. The main editor area displays the following Python code:

```

33 # ===== ALARM & SNAPSHOT CONFIG =====
34 ALARM_PATH = 'alarm.wav'
35 SNAPSHOT_DIR = 'snapshots'
36 os.makedirs(SNAPSHOT_DIR, exist_ok=True)
37
38 last_alert_time = None
39 COOLDOWN = 30 # detik
40
41 # ===== ALARM FUNCTION =====
42 def play_alarm():
43     playsound(ALARM_PATH)
44
45 # ===== EMAIL SENDER FUNCTION =====
46 def send_email_with_image(image_path):
47     msg = MIMEText('')
48     msg['From'] = EMAIL_SENDER
49     msg['To'] = EMAIL_RECEIVER
50     msg['Subject'] = " 🚨 Deteksi Objek Berbahaya"
51
52     body = "Sistem mendeteksi objek berbahaya. Lihat gambar terlampir."
53     msg.attach(MIMEText(body, 'plain'))
54
55     with open(image_path, 'rb') as f:
56         part = MIMEBase('application', 'octet-stream')
57         part.set_payload(f.read())
58         encoders.encode_base64(part)
59         part.add_header('Content-Disposition', f'attachment; filename={os.path.basename(image_path)}')
60         msg.attach(part)
61
62     try:
63         server = smtplib.SMTP_SSL('smtp.gmail.com', 465, context=ssl.create_default_context())
64         server.login(EMAIL_SENDER, EMAIL_PASSWORD)
65         server.send_message(msg)
66

```

The status bar at the bottom indicates 'Ln 1, Col 60, Spaces: 4, UTF-8, LF, Python, Python 3.11 (64-bit)' and the system clock shows '02:31 31/07/2025'.

The screenshot shows a VS Code editor window titled 'yolo_flask_project'. The Explorer sidebar on the left shows the same project structure as the previous screenshot. The main editor area displays the following Python code:

```

46 def send_email_with_image(image_path):
47     server.quit()
48     print(" ✅ Email dikirim.")
49 except Exception as e:
50     print(" ❌ Gagal kirim email:", e)
51
52 # ===== VIDEO STREAM FUNCTION =====
53 def gen_frames():
54     global last_alert_time
55
56     rtsp_url = "rtsp://admin:Nathasya18_@192.168.43.219:554/cam/realmonitor?channel=1&subtype=1"
57     cap = cv2.VideoCapture(rtsp_url, cv2.CAP_FFMPEG)
58     cap.set(cv2.CAP_PROP_BUFFERSIZE, 0)
59     cap.grab()
60     succes, frame = cap.read()
61     .....
62
63     if not cap.isOpened():
64         print(" ❌ Gagal membuka kamera.")
65         return
66
67     prev_time = time.time()
68
69     while True:
70         cap.grab()
71         succes, frame = cap.read()
72         if not succes or frame is None or frame.size == 0:
73             continue
74
75         frame = cv2.resize(frame, (480, 640))
76
77         now = time.time()
78         fps = 1 / (now - prev_time)
79         prev_time = now
80

```

The status bar at the bottom indicates 'Ln 1, Col 60, Spaces: 4, UTF-8, LF, Python, Python 3.11 (64-bit)' and the system clock shows '02:32 31/07/2025'.

```

72 def gen_frames():
99     results = model(frame)[0]
100
101     alert_triggered = False
102     detection_count = {'gun': 0, 'knife': 0}
103
104     for box in results.boxes:
105         cls_id = int(box.cls[0])
106         conf = box.conf[0].item()
107         label = class_names[cls_id]
108         x1, y1, x2, y2 = map(int, box.xyxy[0])
109         color = colors.get(label, (255, 255, 255))
110
111         cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
112         cv2.putText(frame, f'{label.upper()} {conf:.2f}', (x1, y1 - 10),
113                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, color, 2)
114
115         if label in ['gun', 'knife']:
116             detection_count[label] += 1
117
118     # VALIDASI JUMLAH DETEKSI: jika lebih dari 1 deteksi senjata
119     if (detection_count['gun'] + detection_count['knife'] >= 2) and \
120         (last_alert_time is None or now - last_alert_time > COOLDOWN):
121
122         last_alert_time = now
123         timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
124         snapshot_path = os.path.join(SNAPSHOT_DIR, f'{timestamp}.jpg')
125         cv2.imwrite(snapshot_path, frame)
126
127         threading.Thread(target=play_alarm).start()
128         threading.Thread(target=send_email_with_image, args=(snapshot_path,)).start()
129
130     ret, buffer = cv2.imencode('.jpg', frame)

```

```

130     ret, buffer = cv2.imencode('.jpg', frame)
131     frame_bytes = buffer.tobytes()
132
133     yield (b'--frame\r\n'
134           | b'Content-Type: image/jpeg\r\n\r\n' + frame_bytes + b'\r\n')
135
136     # ===== ROUTING =====
137 @app.route('/')
138 def index():
139     return render_template('index.html')
140
141 @app.route('/video')
142 def video():
143     return Response(gen_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')
144
145 @app.route('/start_camera', methods=['POST'])
146 def start_camera():
147     global camera_on
148     camera_on = True
149     return '', 204
150
151 @app.route('/stop_camera', methods=['POST'])
152 def stop_camera():
153     global cap, camera_on, deteksi_on
154     camera_on = False
155     deteksi_on = False
156     if cap:
157         cap.release()
158     return '', 204
159
160 @app.route('/start_detection', methods=['POST'])
161 def start_detection():

```

The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure for 'YOLO_FLASK_PROJECT'. The main editor window shows the code in 'app.py' with the following content:

```

159 @app.route('/start_detection', methods=['POST'])
160 def start_detection():
161     global deteksi_on
162     deteksi_on = True
163     return '', 204
164
165
166 if __name__ == '__main__':
167     app.run(debug=False)
168

```

The status bar at the bottom indicates 'Ln 1, Col 60', 'Spaces: 4', 'UTF-8', 'LF', 'Python', and 'Python 3.11 (64-bit)'.

The screenshot shows the VS Code editor with the file explorer on the left. The main editor window shows the code in 'index.html' with the following content:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Deteksi Objek Real-time</title>
5 <style>
6   body {
7     font-family: Arial, sans-serif;
8     text-align: center;
9     background-color: #010780;
10    color: #333;
11  }
12
13  h2 {
14    margin-top: 40px;
15    font-size: 28px;
16    color: #f6f7f9;
17  }
18
19  button {
20    padding: 10px 20px;
21    font-size: 16px;
22    margin: 10px 5px;
23    border: none;
24    border-radius: 5px;
25    cursor: pointer;
26    color: rgb(0, 0, 0);
27  }
28
29  #startBtn {
30    background-color: #f796ae; /* Green */
31  }
32
33  #stopBtn {
34    background-color: #e44336; /* Red */

```

The status bar at the bottom indicates 'Ln 50, Col 6', 'Spaces: 2', 'UTF-8', 'LF', 'HTML', and 'Python 3.11 (64-bit)'.

The screenshot shows the VS Code editor with the 'index.html' file open. The Explorer sidebar on the left shows the project structure, including files like 'app.py', 'app2.py', 'best.pt', and 'index.html'. The main editor area displays the following code:

```

1  templates > index.html > html > head > style > img
2  <html>
3  <head>
4  <title>
5  </title>
6  <style>
7
8  #stopBtn {
9  |   background-color: #f44336; /* Red */
10 | }
11
12 #detectBtn {
13 |   background-color: #2196f3; /* Blue */
14 | }
15
16 #videoContainer {
17 |   margin-top: 30px;
18 |   display: none;
19 | }
20
21 img {
22 |   border: 5px solid #f8f8f8;
23 |   border-radius: 10px;
24 |   box-shadow: 0px 4px 12px rgba(0, 0, 0, 0.2);
25 | }
26
27 </style>
28 </head>
29 <body>
30 <h2> Deteksi Objek Pengawasan Keamanan </h2>
31
32 <div id="videoContainer">
33 |   
34 </div>
35
36 <br><br>
37 <button id="startBtn" onclick="startCamera()">Hidupkan Kamera</button>

```

The status bar at the bottom indicates the cursor is at line 50, column 6, with 2 spaces, in UTF-8 encoding, editing an HTML file.

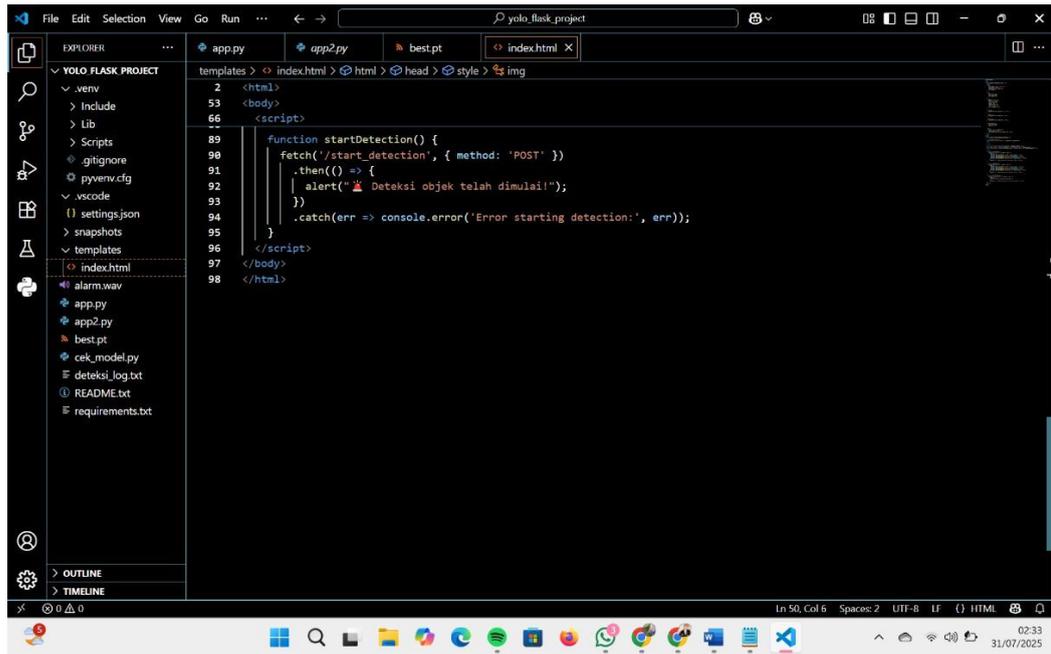
The screenshot shows the VS Code editor with the 'index.html' file open, displaying the JavaScript code. The Explorer sidebar on the left shows the project structure. The main editor area displays the following code:

```

53 <body>
54
55 <button id="startBtn" onclick="startCamera()">Hidupkan Kamera</button>
56 <button id="stopBtn" onclick="stopCamera()" style="display: none;">Matikan Kamera</button>
57 <button id="detectBtn" onclick="startDetection()" style="display: none;">Mulai Deteksi</button>
58
59 <script>
60   function startCamera() {
61     fetch('/start_camera', { method: 'POST' })
62       .then(() => {
63         document.getElementById("startBtn").style.display = "none";
64         document.getElementById("stopBtn").style.display = "inline";
65         document.getElementById("detectBtn").style.display = "inline";
66         document.getElementById("videoContainer").style.display = "block";
67       })
68     .catch(err => console.error('Error starting camera:', err));
69   }
70
71   function stopCamera() {
72     fetch('/stop_camera', { method: 'POST' })
73       .then(() => {
74         document.getElementById("startBtn").style.display = "inline";
75         document.getElementById("stopBtn").style.display = "none";
76         document.getElementById("detectBtn").style.display = "none";
77         document.getElementById("videoContainer").style.display = "none";
78       })
79     .catch(err => console.error('Error stopping camera:', err));
80   }
81
82   function startDetection() {
83     fetch('/start_detection', { method: 'POST' })
84       .then(() => {
85         alert("Deteksi objek telah dimulai!");
86       })
87     .catch(err => console.error('Error starting detection:', err));
88   }
89
90 </script>
91
92 </body>

```

The status bar at the bottom indicates the cursor is at line 50, column 6, with 2 spaces, in UTF-8 encoding, editing an HTML file.



The image shows a screenshot of the Visual Studio Code (VS Code) editor interface. The window title is "yolo_flask_project". The Explorer sidebar on the left shows the project structure for "YOLO_FLASK_PROJECT", including folders like ".venv", "Include", "Lib", "Scripts", ".gitignore", "pyvenv.cfg", ".vscode", "settings.json", "snapshots", "templates", and "alarm.wav". The "templates" folder is expanded, showing "index.html".

The main editor area displays the content of "index.html". The code is as follows:

```
1 <html>
2 <body>
3 <script>
4
5     function startDetection() {
6
7         fetch('/start_detection', { method: 'POST' })
8         .then(() => {
9             alert("🚨 Deteksi objek telah dimulai!");
10        })
11        .catch(err => console.error("Error starting detection:", err));
12    }
13
14 </script>
15 </body>
16 </html>
```

The status bar at the bottom indicates the current cursor position: "Ln 50, Col 6", "Spaces: 2", "UTF-8", "LF", and "HTML". The system tray at the bottom right shows the time "02:33" and date "31/07/2025".

ORIGINALITY REPORT

20%

SIMILARITY INDEX

19%

INTERNET SOURCES

6%

PUBLICATIONS

11%

STUDENT PAPERS

PRIMARY SOURCES

1	eprints.untirta.ac.id Internet Source	3%
2	Submitted to Universitas Muhammadiyah Sumatera Utara Student Paper	2%
3	repository.umsu.ac.id Internet Source	1%
4	repository.uin-suska.ac.id Internet Source	1%
5	repository.usbypkp.ac.id Internet Source	1%
6	Submitted to unimal Student Paper	1%
7	eprints.uty.ac.id Internet Source	1%
8	repositori.umrah.ac.id Internet Source	< 1%
9	repository.its.ac.id Internet Source	< 1%
10	Submitted to Sekolah Teknik Elektro & Informatika Student Paper	< 1%

Submitted to Marmara University

11	Student Paper	< 1 %
12	repository.dinamika.ac.id Internet Source	< 1 %
13	123dok.com Internet Source	< 1 %
14	eprints.uad.ac.id Internet Source	< 1 %
15	Nuri Cahyono, Anggista Oktavia Praneswara. "Analisis Sentimen Ulasan Aplikasi TikTok Shop Seller Center di Google Playstore Menggunakan Algoritma Naive Bayes", Indonesian Journal of Computer Science, 2023 Publication	< 1 %
16	repository.univ-tridinanti.ac.id Internet Source	< 1 %
17	stackoverflow.com Internet Source	< 1 %
18	Submitted to Universitas Islam Riau Student Paper	< 1 %
19	eprints.uny.ac.id Internet Source	< 1 %
20	etd.repository.ugm.ac.id Internet Source	< 1 %
21	docplayer.info Internet Source	< 1 %
22	repository.unpar.ac.id Internet Source	< 1 %

23 Argi Nur Faturrohman, Sayekti Harits Suryawan, Abdul Rahim. "Pengembangan Model Klasifikasi Kendaraan Keluar Masuk Area Parkir Dengan Algoritma YOLOv8", Teknika, 2024

Publication

< 1 %

24 Robby Kamil Siregar, Anton Anton, Widiastuti Widiastuti. "Perancangan Aplikasi Bahasa Isyarat "Isyaratku" Dengan Deep Learning Serta Google Cloud Platform", Simpatik: Jurnal Sistem Informasi dan Informatika, 2021

Publication

< 1 %

25 e-jurnal.pnl.ac.id

Internet Source

< 1 %

26 ejournal3.undip.ac.id

Internet Source

< 1 %

27 repository.adzkia.ac.id

Internet Source

< 1 %

28 text-id.123dok.com

Internet Source

< 1 %

29 Submitted to Universitas Muhammadiyah Purwokerto

Student Paper

< 1 %

30 repository.uinjkt.ac.id

Internet Source

< 1 %

31 Nazmul Siddique, Mohammad Shamsul Arefin, Ahmed Wasif Reza, Aminul Haque. "Data-Driven Applications for Emerging Technologies", CRC Press, 2025

Publication

< 1 %

32	Submitted to University of Essex Student Paper	< 1 %
33	Submitted to Arab Open University Student Paper	< 1 %
34	Submitted to CSU, Dominguez Hills Student Paper	< 1 %
35	Submitted to LL DIKTI IX Turnitin Consortium Part III Student Paper	< 1 %
36	Submitted to Universitas Sumatera Utara Student Paper	< 1 %
37	etheses.uingusdur.ac.id Internet Source	< 1 %
38	Submitted to Universitas Kristen Wira Wacana Sumba Student Paper	< 1 %
39	eprints.undip.ac.id Internet Source	< 1 %
40	en.brilio.net Internet Source	< 1 %
41	repository.unpkediri.ac.id Internet Source	< 1 %
42	repository.ustjogja.ac.id Internet Source	< 1 %
43	www.jptam.org Internet Source	< 1 %
44	Submitted to Universitas Diponegoro Student Paper	< 1 %

45	jurnal.stkipppgritulungagung.ac.id Internet Source	< 1 %
46	namephologyid.blogspot.com Internet Source	< 1 %
47	www.domainesia.com Internet Source	< 1 %
48	www.jurnalet.com Internet Source	< 1 %
49	www.openjournal.unpam.ac.id Internet Source	< 1 %
50	Nurhayati Nurhayati, Aditya Kusuma. "PERANCANGAN SISTEM DETEKSI KEMATANGAN BUAH NAGA MERAH BERBASIS WEB MENGGUNAKAN METODE ALGORITMA YOLO (YOU ONLY LOOK ONCE)", JUTECH : Journal Education and Technology, 2025 Publication	< 1 %
51	Submitted to UNIVERSITAS BUDI LUHUR Student Paper	< 1 %
52	digilib.unila.ac.id Internet Source	< 1 %
53	dspace.uii.ac.id Internet Source	< 1 %
54	eprints.radenfatah.ac.id Internet Source	< 1 %
55	ifrelresearch.org Internet Source	< 1 %
56	Imam Fathurrahman, Muhammad Djamaluddin, Zaenul Amri, M. Nurul Wathani.	< 1 %

"Klasifikasi Motif Batik Nusantara Menggunakan Vision Transformer (ViT) Berbasis Deep Learning", Infotek: Jurnal Informatika dan Teknologi, 2025

Publication

57 Ni Luh Putu Ning Septyarini Putri Astawa, Putu Trisna Hady Permana. "Media Pembelajaran dengan Kecerdasan Buatan dalam Pembelajaran Bahasa Inggris Generasi-Z", Jurnal Sains Sosio Humaniora, 2020

Publication

58 Submitted to Texas Tech University Main Campus

Student Paper

59 data.sekolah-kita.net

Internet Source

60 digilib.uinsby.ac.id

Internet Source

61 eprints.unm.ac.id

Internet Source

62 jurnalsecurity.com

Internet Source

63 repository.ittelkom-pwt.ac.id

Internet Source

64 repository.unmuhjember.ac.id

Internet Source

65 repository.unugiri.ac.id

Internet Source

66 Elisa Tikasni, Ema Utami, Dhani Ariatmanto. "Analisis Akurasi Object Detection

Menggunakan Tensorflow Untuk Pengenalan Bahasa Isyarat Tangan Menggunakan Metode SSD", JURNAL FASILKOM, 2024

Publication

67	Submitted to LPPM Student Paper	< 1 %
68	Luhur Pambudi, Sefrika Sefrika. "Klasifikasi Dampak dan Kondisi Pasien Hepatitis Menggunakan Metode Decision Tree", RIGGS: Journal of Artificial Intelligence and Digital Business, 2025 Publication	< 1 %
69	Submitted to Politeknik Negeri Lampung Student Paper	< 1 %
70	digidahua.com Internet Source	< 1 %
71	repositories.buddhidharma.ac.id Internet Source	< 1 %
72	repository.itskesicme.ac.id Internet Source	< 1 %
73	repository.unej.ac.id Internet Source	< 1 %
74	www.bhinneka.com Internet Source	< 1 %
75	Submitted to Universitas Muhammadiyah Palembang Student Paper	< 1 %
76	digilib.uinsa.ac.id Internet Source	< 1 %

77	eprints.universitaspotrabangsa.ac.id Internet Source	< 1 %
78	eprints3.upgris.ac.id Internet Source	< 1 %
79	erepository.uwks.ac.id Internet Source	< 1 %
80	es.scribd.com Internet Source	< 1 %
81	eskripsi.usm.ac.id Internet Source	< 1 %
82	fr.slideserve.com Internet Source	< 1 %
83	journal.sttindonesia.ac.id Internet Source	< 1 %
84	ms.eferrit.com Internet Source	< 1 %
85	nehegallery.blogspot.com Internet Source	< 1 %
86	openjournal.unpam.ac.id Internet Source	< 1 %
87	repository.untar.ac.id Internet Source	< 1 %
88	www.mdpi.com Internet Source	< 1 %
89	www.scribd.com Internet Source	< 1 %
90	Prasenjit Dey, Sudip Kumar Adhikari, Sourav De, Indrajit Kar. "Internet of Things-Based	< 1 %

Machine Learning in Healthcare – Technology and Applications", CRC Press, 2024

Publication

91 Dadang Iskandar Mulyana, M Ainur Rofik. < 1%
"Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5", Jurnal Pendidikan Tambusai, 2022

Publication

92 eprints.walisongo.ac.id < 1%
Internet Source

Exclude quotes On

Exclude matches Off

Exclude bibliography On