IMPLEMENTASI *YOU ONLY LOOK ONCE (YOLOv11)* UNTUK MENDETEKSI API DAN ASAP BERBASIS *WEBSITE STREAMLIT*

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Memperoleh Gelar Strata Satu (S1) Teknik Informatika

Oleh:

RAKA RAHADIAN

211351117

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI TEKNOLOGI WASTUKANCANA

PURWAKARTA

2025

SURAT PERNYATAAN

Saya, Raka Rahadian menyatakan dengan sesungguhnya, bahwa skripsi yang berjudul: "IMPLEMENTASI *YOU ONLY LOOK ONCE (YOLOv11)* UNTUK MENDETEKSI API DAN ASAP BERBASIS *WEBSITE STREAMLIT*" Adalah benar hasil karya sendiri, serta tidak terdapat karya yang pernah diajukan untuk persyaratan mata kuliah skripsi dan pengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis/diterbitkan orang lain, kecuali yang tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

PEDOMAN SKRIPSI

Skripsi S1 yang tidak dipublikasikan terdaftar dan tersedia di Perpustakaan Sekolah Tinggi Teknologi Wastukancana, dan terbuka untuk umum dengan ketentuan bahwa hak cipta ada pada pengarang dengan mengikuti aturan HaKI yang berlaku di Sekolah Tinggi Teknologi Wastukancana. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau peringkasan hanya dapat dilakukan seizin pengarang dan harus disertai dengan kebiasaan ilmiah untuk menyebutkan sumbernya. Memperbanyak atau menerbitkan sebagian atau seluruh Skripsi haruslah seizin Ketua Sekolah Tinggi Teknologi Wastukancana Purwakarta.

ABSTRAK

IMPLEMENTASI *YOU ONLY LOOK ONCE (YOLOV11)* UNTUK MENDETEKSI API DAN ASAP BERBASIS *WEBSITE STREAMLIT*

OLEH:
RAKA RAHADIAN
211351117

Program Studi Teknik Informatika
Sekolah Tinggi Teknologi Wastukancana

Penelitian ini mengembangkan sistem deteksi Api dan asap secara *Real-Time* menggunakan algoritma *You Only Look Once* versi 11 (*YOLOv11*) yang diintegrasikan ke dalam platform *Streamlit*. Sistem ini menggunakan *dataset* Smoke and Fire v2 dari Roboflow yang mencakup 9.848 citra beranotasi, dilengkapi teknik *augmentasi* seperti *mosaic*, rotasi, dan penyesuaian kecerahan guna meningkatkan keragaman data. Proses pelatihan dilakukan di *Google Colaboratory*, sedangkan antarmuka sistem memungkinkan pengguna mengunggah gambar atau video serta menerima notifikasi otomatis melalui Telegram. Hasil evaluasi menunjukkan *model* mencapai *mAP@0.5* sebesar 0,88, *precision* 0,89, dan *recall* 0,87, dengan *response time* rata-rata 0,35 detik per frame. Kelas *Fire* dan *Smoke* berhasil diklasifikasikan dengan *F1-score* masing-masing sebesar 99,52% dan 98,67%. Meskipun performa sistem tinggi, tantangan masih ditemukan pada deteksi asap samar dan latar belakang kompleks. Sistem ini terbukti efektif sebagai solusi deteksi kebakaran yang ringan, cepat, dan dapat diakses melalui *web*, serta memiliki potensi untuk diintegrasikan ke dalam infrastruktur pemantauan kebakaran skala luas.

Kata kunci: *YOLOv11*, deteksi kebakaran, *Real-Time*, *Streamlit*, *Computer vision*

ABSTRACT

IMPLEMENTATION OF YOU ONLY LOOK ONCE (YOLOV11) FOR FIRE AND SMOKE DETECTION BASED ON STREAMLIT WEBSITE

BY:

RAKA RAHADIAN

211351117

Informatics Engineering Study Program

Sekolah Tinggi Teknologi Wastukancana

This research developed a Real-Time fire and smoke detection system using the You Only Look Once version 11 (YOLOv11) algorithm, integrated into the Streamlit platform. The system utilizes the Smoke and Fire v2 dataset from Roboflow, consisting of 9,848 annotated images, along with augmentation techniques such as mosaic, rotation, and brightness adjustment to enhance data diversity. The model was trained using Google Colaboratory, while the system interface enables users to upload images or videos and receive automated notifications via Telegram. Evaluation results show the model achieved a mAP@0.5 of 0.88, precision of 0.89, and recall of 0.87, with an average response time of 0.35 seconds per frame. The Fire and Smoke classes were classified with F1-scores of 99.52% and 98.67%, respectively. Although the system performs strongly, challenges remain in detecting faint smoke and handling complex backgrounds. Overall, this system demonstrates effectiveness as a lightweight, fast, and web-accessible fire detection solution, with potential for integration into large-scale fire monitoring infrastructure.

Keywords*: YOLOv11, fire detection, Real-Time, Streamlit, Computer vision*

KATA PENGANTAR

Segala puji dan syukur saya panjatkan ke hadirat Allah SWT atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga saya dapat melaksanakan penelitian dan menyusun skripsi ini yang berjudul *"*Implementasi *You Only Look Once* (YOLOv11) untuk Mendeteksi Api dan Asap Berbasis Website *Streamlit."*

Skripsi ini disusun sebagai bagian dari pemenuhan salah satu syarat akademik dalam menyelesaikan studi di Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Wastukancana Purwakarta. Saya menyadari bahwa proses penyusunan skripsi ini tidak akan berjalan lancar tanpa adanya bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, dengan penuh kerendahan hati, saya menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Ir. Apang Djafar Shieddieque, S.T., M.T., IPM., ASEAN Eng., selaku Ketua Sekolah Tinggi Teknologi Wastukancana Purwakarta, atas dukungan dan fasilitas yang diberikan.
2. Bapak Teguh Iman Hermanto, M.Kom., selaku Ketua Program Studi Teknik Informatika, atas arahan dan motivasi yang terus mendorong proses akademik saya.
3. Bapak Muhammad Rafi Muttaqin, S.Kom., M.Kom., selaku dosen pembimbing, atas bimbingan, nasihat, serta saran-saran yang sangat berharga selama proses penyusunan skripsi ini.
4. Bapak Yusuf Muhyidin, M.Kom., selaku dosen pendamping, atas dukungan dan masukan yang turut memperkaya isi skripsi ini.
5. Rekan-rekan mahasiswa, khususnya kelas Pagi A angkatan 2021, atas kebersamaan, kerja sama, dan semangat yang diberikan selama proses penyusunan skripsi.
6. Sahabat-sahabat terdekat yang senantiasa mendampingi dan memberikan dukungan moril di setiap langkah perjuangan ini.

7. Yang teristimewa, kepada kedua orang tua dan keluarga tercinta, atas doa, kasih sayang, serta dukungan baik secara moril maupun materiil yang menjadi sumber kekuatan bagi saya.

Saya menyadari bahwa skripsi ini masih jauh dari kata sempurna dan masih memerlukan berbagai penyempurnaan. Oleh karena itu, saya sangat terbuka terhadap segala kritik dan saran yang bersifat membangun guna meningkatkan kualitas skripsi ini pada tahap berikutnya.

Semoga skripsi ini dapat memberikan manfaat serta menjadi referensi yang berguna bagi pengembangan ilmu pengetahuan dan pihak-pihak yang berkepentingan.

Purwakarta, 01 Maret 2025

Penulis

Raka Rahadian

211351117

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Kebakaran merupakan ancaman serius bagi lingkungan, ekonomi, dan kesehatan manusia, terutama di negara tropis seperti Indonesia yang memiliki hutan luas dan kepadatan penduduk tinggi di kawasan perkotaan. Menurut Badan Nasional Penanggulangan Bencana (BNPB), sepanjang tahun 2024 lebih dari 629 kasus kebakaran hutan dan lahan terjadi di Indonesia, mengakibatkan kerugian ekonomi yang signifikan serta dampak kesehatan akibat polusi asap kebakaran (Vetrita et al., 2024). Faktor utama penyebab kebakaran di Indonesia meliputi cuaca ekstrem, pembukaan lahan dengan metode pembakaran, serta keterbatasan sistem deteksi dini yang efektif. Oleh karena itu, diperlukan sistem deteksi kebakaran yang cepat dan akurat untuk mengurangi risiko bencana serta dampaknya terhadap masyarakat.

Gambar 1. 1 Data kebakaran di Indonesia 2024

(PUSDATINKOM BNPB, 2024)

Dalam beberapa tahun terakhir, teknologi kecerdasan buatan (*Artificial Intelligence/AI*) telah mengalami perkembangan pesat, termasuk dalam penerapannya pada sistem deteksi kebakaran. Salah satu yang sangat populer untuk deteksi objek adalah *You Only Look Once* (YOLO). YOLO dikenal karena kemampuannya dalam

mendeteksi objek secara *Real-Time* dengan tingkat akurasi yang tinggi. *YOLOv11*, sebagai versi terbaru dari ini, menawarkan peningkatan dalam hal efisiensi dan akurasi deteksi dibandingkan dengan versi sebelumnya (Pradana et al., 2024)

Salah satu tantangan utama dalam sistem deteksi kebakaran adalah keterbatasan infrastruktur serta biaya tinggi dalam pemasangan sensor dan kamera pengawas. Oleh karena itu, pemanfaatan teknologi berbasis *web* menjadi alternatif yang lebih ekonomis dan fleksibel. Studi yang dilakukan oleh (Bindal et al., 2024) menunjukkan bahwa YOLO dapat diterapkan dalam sistem berbasis *web* untuk mendeteksi kebakaran secara cepat dalam berbagai kondisi lingkungan. Dengan meningkatnya adopsi aplikasi berbasis *web* interaktif, solusi ini dapat dimanfaatkan secara luas oleh berbagai pihak sebagai sistem peringatan dini yang mudah diakses.

Dalam penelitian ini, teknologi berbasis *web* yang digunakan adalah *Streamlit*, yang memungkinkan pengembangan antarmuka yang interaktif dan sederhana tanpa memerlukan infrastruktur *cloud* yang kompleks. *Streamlit* dipilih karena kemudahannya dalam integrasi dengan *model Deep learning* Serta kemampuannya dalam menampilkan hasil deteksi secara *Real-Time* tanpa membutuhkan perangkat tambahan seperti sensor khusus atau jaringan 5G. Dengan pendekatan ini, sistem deteksi kebakaran dapat dijalankan secara lokal maupun melalui server ringan dengan efisiensi tinggi.

Meskipun teknologi *YOLOv11* memiliki banyak keunggulan, penerapannya dalam deteksi kebakaran masih menghadapi beberapa tantangan. Salah satunya adalah variasi kondisi pencahayaan dan lingkungan yang dapat mempengaruhi akurasi deteksi Api dan asap. Penelitian oleh (Saputri et al., 2025) menyoroti pentingnya pengoptimalan *model Deep learning* untuk meningkatkan akurasi deteksi dalam kondisi ekstrem, seperti kabut asap tebal atau pencahayaan rendah. Selain itu, pengelolaan data dalam jumlah besar dari berbagai sumber juga menjadi tantangan yang perlu diatasi agar sistem dapat berjalan secara efisien. Keberhasilan sistem ini sangat bergantung pada kemampuan dalam menyesuaikan parameter *model* dengan berbagai kondisi lingkungan yang dinamis.

Dengan meningkatnya frekuensi kebakaran akibat perubahan iklim dan aktivitas manusia, pengembangan sistem deteksi Api dan asap berbasis *web* dengan *YOLOv11* menjadi semakin penting. Implementasi teknologi ini dapat membantu mitigasi bencana, mengurangi emisi karbon akibat kebakaran hutan, serta meningkatkan kesadaran masyarakat dalam upaya pencegahan kebakaran (Raya Ismail et al., 2025). Sistem berbasis *Computer vision* juga dapat digunakan sebagai alat pendukung dalam perencanaan tata ruang dan kebijakan lingkungan yang lebih efektif. Selain itu, dengan penyebaran yang luas, sistem ini dapat menjadi alat kolaboratif bagi berbagai pihak dalam mengurangi dampak kebakaran.

Dengan demikian, penulis memilih judul "IMPLEMENTASI *YOU ONLY LOOK ONCE (YOLOV11)* UNTUK MENDETEKSI API DAN ASAP BERBASIS *WEBSITE STREAMLIT*" ini penelitian ini tidak hanya relevan secara teknis, tetapi juga memberikan kontribusi dalam mendukung upaya mitigasi risiko kebakaran secara lebih efektif dan efisien. Melalui integrasi antara teknologi deteksi berbasis AI dan *Platform* web interaktif, penelitian ini diharapkan mampu menjadi solusi inovatif dalam bidang keselamatan dan keamanan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dirumuskan suatu masalah yang dapat diangkat adalah Bagaimana cara mengembangkan sistem deteksi kebakaran berbasis *Website Streamlit* agar dapat mendeteksi api dan asap secara *Real-Time* dengan akurasi yang baik?

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem deteksi dini kebakaran berbasis *You Only Look Once* versi 11 (*YOLOv11*) yang diintegrasikan ke dalam platform *website* menggunakan *Streamlit*. Tujuan utama penelitian dirinci sebagai berikut:

1. Mengembangkan sistem deteksi objek berbasis YOLOv11 yang mampu mengenali api dan asap secara *Real-Time*, serta menampilkannya dalam antarmuka web interaktif menggunakan *Framework Streamlit*.
2. Melatih dan menguji *model* YOLOv11 pada *Dataset Smoke and Fire v2* dari Roboflow, dengan menerapkan teknik augmentasi dan tuning parameter untuk memperoleh *model* yang optimal dalam berbagai kondisi pencahayaan dan latar.
3. Mengevaluasi performa *model* deteksi menggunakan metrik *Mean Average Precision* (*mAP@0.5*), presisi, dan *Recall*, guna menilai akurasi dan sensitivitas sistem terhadap objek api dan asap.
4. Mengukur efisiensi sistem deteksi dalam platform web, khususnya waktu respons (*response time*) per *frame* dan kelayakan implementasi pada perangkat dengan sumber daya terbatas, seperti server ringan atau lingkungan *edge computing*.
5. Mengintegrasikan sistem deteksi dengan notifikasi otomatis melalui Telegram sebagai bentuk penerapan sistem peringatan dini (*early warning*) yang mudah diakses oleh pengguna.

1.4 Manfaat Penelitian

Penelitian ini memiliki manfaat yang luas, baik dalam aspek akademik, praktis, maupun sosial.

1.4.1 Manfaat Akademik

Penelitian ini berkontribusi dalam memperkaya literatur mengenai penerapan *Deep learning* dan *Computer vision* dalam sistem deteksi dini kebakaran, khususnya dalam konteks pengembangan *website* berbasis *Streamlit*. Dengan mengeksplorasi optimasi *YOLOv11*, penelitian ini memberikan wawasan baru terkait tantangan dan solusi dalam mendeteksi api dan asap secara *Real-Time* di berbagai kondisi lingkungan. Hasil penelitian ini juga dapat menjadi referensi bagi akademisi dan pengembang yang ingin mengembangkan teknologi serupa untuk mitigasi bencana lainnya, seperti pendeteksian banjir atau polusi udara.

1.4.2 Manfaat Praktis

Penelitian ini menghadirkan solusi inovatif berbasis *website* yang memungkinkan deteksi kebakaran dengan cepat dan akurat. Dengan pendekatan *Computer vision* yang menggunakan *YOLOv11*, sistem ini menawarkan alternatif yang lebih ekonomis dibandingkan dengan sensor fisik konvensional, sehingga dapat diimplementasikan dengan lebih mudah di berbagai sektor, mulai dari kawasan industri hingga pemukiman. Penggunaan *Platform website* berbasis *Streamlit* juga memungkinkan aksesibilitas yang lebih luas tanpa memerlukan perangkat keras tambahan yang mahal.

1.4.3 Manfaat Sosial

Dampak penelitian ini sangat signifikan dalam mengurangi risiko kebakaran dan dampaknya terhadap kesehatan, ekonomi, serta lingkungan. Dengan sistem deteksi yang lebih cerdas dan cepat, kebakaran dapat dicegah sebelum meluas, sehingga potensi kerugian nyawa dan harta benda dapat diminimalkan. Selain itu, teknologi ini dapat digunakan oleh pemerintah dan lembaga terkait untuk meningkatkan kesiapsiagaan menghadapi kebakaran dengan memberikan data *Real-Time* yang dapat membantu dalam pengambilan keputusan yang lebih cepat dan akurat.

Secara keseluruhan, penelitian ini bukan hanya merupakan inovasi teknologi, tetapi juga solusi nyata yang dapat menyelamatkan nyawa, melindungi lingkungan, dan memberikan dampak sosial yang positif. Dengan menggabungkan kecanggihan teknologi AI dan kebutuhan nyata di lapangan, penelitian ini berpotensi menjadi langkah maju dalam menciptakan sistem deteksi kebakaran yang lebih responsif, efisien, dan mudah diakses oleh masyarakat luas.

1.5 Batasan Masalah

Dalam penelitian ini, terdapat beberapa batasan yang ditetapkan untuk memastikan fokus penelitian tetap terarah dan sistem deteksi kebakaran berbasis *YOLOv11* pada *Platform website* dapat dikembangkan secara efektif:

1. Lingkup Deteksi

   Sistem difokuskan pada deteksi api dan asap menggunakan *YOLOv11*, yang

dipilih karena keunggulannya dalam deteksi objek secara *Real-Time* dengan akurasi tinggi dan latensi rendah. *Dataset* yang digunakan dalam penelitian ini adalah *Fire & Smoke Detection YOLOv11* dari Roboflow Universe oleh Sayed Gamall (2025), yang mencakup gambar api dan asap dengan *anotasi* yang sesuai. Objek lain yang terkait dengan kebakaran, seperti sumber panas atau sensor suhu, tidak disertakan untuk menjaga kesederhanaan dan efisiensi sistem.

2. *Platform* dan Perangkat

   Implementasi dilakukan pada *Platform website* berbasis *Streamlit* untuk memastikan aksesibilitas dan kemudahan penggunaan tanpa memerlukan perangkat keras khusus. Namun, ini membawa keterbatasan dalam sumber daya komputasi dibandingkan dengan sistem berbasis server atau *edge computing*. Oleh karena itu, optimasi *model* diperlukan untuk memastikan kinerja yang memadai.

3. Kondisi Lingkungan

   Pengujian sistem dilakukan dengan menggunakan data dari *Dataset* yang mencakup berbagai kondisi pencahayaan dan lingkungan realistis, termasuk variasi pencahayaan (rendah, normal, silau) dan skala objek (kecil, sedang, besar). Namun, kondisi ekstrem seperti kabut tebal, hujan deras, atau interferensi visual yang kompleks tidak menjadi fokus utama penelitian ini, mengingat keterbatasan cakupan pada *Dataset* yang digunakan.

4. Optimasi *Model*

   Untuk memastikan efisiensi pada *Platform website*, *model YOLOv11* dioptimalkan melalui *tuning hyperparameter* dan teknik kompresi *model* seperti konversi ke format PT. Meskipun demikian, penelitian ini tidak melakukan modifikasi fundamental pada arsitektur *YOLOv11*, melainkan berfokus pada adaptasi terhadap *Dataset* dan lingkungan *deployment*.

5. Evaluasi Kinerja

   Kinerja sistem dievaluasi berdasarkan metrik akurasi deteksi (*mAP@0.5*, presisi, *Recall*), kecepatan inferensi, dan efisiensi pemrosesan data. Pengujian

dilakukan dengan menggunakan *Fire & Smoke Detection YOLOv11 Dataset* dari Roboflow Universe untuk mem*valid*asi generalisasi *model*. Namun, perbandingan komprehensif dengan *model* deteksi lain di luar *YOLOv11* tidak dilakukan, mengingat fokus penelitian pada implementasi *YOLOv11* dalam konteks *web-based*.

Dengan adanya batasan ini, penelitian dapat lebih terfokus dalam mengembangkan dan mengevaluasi sistem deteksi kebakaran berbasis *YOLOv11* untuk *Platform website* dengan mempertimbangkan efisiensi, akurasi, serta implementasi dalam kondisi nyata.

1.6 Sistematika Penulisan

BAB I Pendahuluan

Membahas latar belakang kebutuhan sistem deteksi kebakaran yang cepat dan akurat, rumusan masalah, tujuan penelitian dalam mengembangkan sistem deteksi berbasis *YOLOv11* pada *Platform website Streamlit*, serta manfaat penelitian dalam aspek akademik, praktis, dan sosial. Bab ini juga menjelaskan batasan masalah dan memberikan gambaran umum sistematika penulisan.

BAB II Tinjauan Pustaka

Menguraikan teori-teori yang mendukung penelitian, termasuk konsep deteksi api dan asap menggunakan *Computer vision*, prinsip kerja *YOLOv11*, serta teknologi *Deep learning* dalam sistem pengenalan objek. Ulasan penelitian terdahulu yang relevan juga disajikan sebagai dasar dalam mengembangkan sistem berbasis *website*, termasuk metode optimasi untuk meningkatkan akurasi dan efisiensi inferensi.

BAB III Metodologi Penelitian

Menjelaskan metode penelitian meliputi desain sistem deteksi kebakaran berbasis *YOLOV11*, pengolahan *Fire & Smoke Detection YOLOv11 Dataset* dari Roboflow Universe untuk pelatihan dan pengujian *model*, teknik optimasi *hyperparameter* dan kompresi *model*, serta implementasi pada *Platform website Streamlit*. Diagram alur sistem dan tahapan evaluasi performa juga dipaparkan.

BAB IV Hasil dan Pembahasan

Menyajikan hasil pengujian sistem deteksi, termasuk analisis akurasi deteksi api dan asap (*mAP@0.5*, presisi, *Recall*), kecepatan inferensi pada *website*, dan efektivitas dalam berbagai kondisi lingkungan. Pembahasan membandingkan eksperimen dengan penelitian terdahulu, mengidentifikasi tantangan, serta peluang pengembangan lebih lanjut.

BAB V Kesimpulan dan Saran

Berisi kesimpulan penelitian berdasarkan hasil yang diperoleh, mencakup pencapaian sistem deteksi kebakaran berbasis *YOLOv11* di *Platform website* dan tantangan yang masih perlu diatasi. Selain itu, diberikan saran untuk pengembangan selanjutnya, seperti peningkatan akurasi *model*, integrasi dengan sistem peringatan dini, dan eksplorasi teknologi pendukung deteksi kebakaran *Real-Time*.

BAB II
TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan (*Artificial Intelligence*)

Artificial Intelligence atau kecerdasan buatan adalah sistem komputer yang mampu melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia. Teknologi ini dapat membuat keputusan dengan cara menganalisis dan menggunakan data yang tersedia di dalam sistem. Proses yang terjadi dalam *Artificial Intelligence* mencakup *learning*, *reasoning*, dan *self-correction.* Proses ini mirip dengan manusia yang melakukan analisis sebelum memberikan keputusan (Sobron M et al., 2021).

Artificial Intelligence merupakan kecerdasan buatan yang diciptakan oleh manusia agar dapat mengoperasikan komputer atau suatu sistem. Kecerdasan buatan tersebut dibuat untuk mempermudah pekerjaan manusia (Arnesia et al., 2022) .

2.2 *Machine learning*

Machine learning atau Pembelajaran Mesin adalah teknik pendekatan dari *Artificial Intelligent* (AI) yang digunakan untuk meniru untuk menggantikan peran manusia dalam melakukan aktivitas untuk memecahkan masalah. Singkatnya, *Machine learning* adalah sebuah mesin yang dibuat untuk dapat belajar dan melakukan pekerjaan tanpa arahan dari penggunanya (Wijoyo et al., 2024).

Machine learning dapat mengoptimalkan proses pengambilan keputusan dengan menyajikan *insight* yang bersifat prediktif dan berjalan secara otomatis. Pembelajaran mesin atau *Machine learning* adalah studi berkelanjutan tentang konsep pengenalan pola dan pembelajaran komputasi dalam kecerdasan buatan yang menggunakan pembelajaran seperti diawasi dan tidak diawasi untuk memprediksi dan mendukung pengambilan keputusan otomatis berdasarkan sekumpulan data (Wardhana et al., 2023).

2.3 *Deep learning*

Deep learning (DL), sebuah cabang dari pembelajaran mesin (ML) dan kecerdasan buatan (AI), saat ini dianggap sebagai teknologi inti dari Revolusi Industri Keempat (4IR atau *Industry* 4.0). Karena kemampuannya untuk belajar dari data, teknologi DL yang berasal dari jaringan saraf tiruan (ANN), telah menjadi topik hangat dalam konteks komputasi, dan diterapkan secara luas di berbagai bidang aplikasi seperti perawatan kesehatan, pengenalan visual, analitik teks, keamanan siber, dan banyak lagi (Sarker, 2021).

Deep learning merupakan suatu proses pembelajaran yang bersifat mendalam, transformatif, dan berkelanjutan, yang bertujuan untuk meningkatkan pemahaman serta penguasaan terhadap suatu bidang ilmu atau keterampilan tertentu. Dalam penerapannya, *Deep learning* tidak hanya sekadar menyerap informasi secara pasif, tetapi juga melibatkan proses berpikir yang kompleks dan bertingkat tinggi. Proses ini mencakup berbagai aspek kognitif, seperti analisis mendalam terhadap suatu konsep atau permasalahan, sintesis dalam menghubungkan berbagai informasi yang diperoleh, serta evaluasi kritis untuk menilai validitas dan relevansi informasi tersebut (Hendrianty et al., 2024).

2.4 *Computer vision* (*Digital Image Processing*)

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (video). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer (Dompeipen & Sompie, 2020)

Citra merupakan salah satu komponen multimedia yang berperan sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, citra kaya dengan informasi. Citra adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses sampling. Pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual, yakni data masukan maupun data keluarannya berbentuk citra (Achmad Rizal et al., 2022).

2.4.1 Konvolusi dan *feature extraction*

Salah satu teknik utama dalam pengolahan citra digital adalah konvolusi. Konvolusi adalah operasi matematika yang melibatkan dua fungsi untuk menghasilkan fungsi ketiga yang menyatakan bagaimana bentuk salah satu fungsi mempengaruhi fungsi lainnya. Dalam konteks pengolahan citra, konvolusi biasanya dilakukan dengan menggunakan filter atau kernel yang berukuran kecil (misalnya 3x3 atau 5x5) yang dipindahkan pixel demi pixel di atas citra input.

Proses konvolusi membantu dalam mengekstrak fitur-fitur penting dari citra. Fitur-fitur ini dapat berupa tepi, tekstur, atau pola tertentu yang relevan untuk tugas pengenalan objek. Dengan menerapkan berbagai filter konvolusi, *model* dapat belajar fitur-fitur yang penting untuk membedakan objek satu dengan objek lainnya.

2.4.2 Hubungan dengan *Object detection*

Konvolusi dan *feature extraction* memiliki peran yang sangat penting dalam *Object detection*. Dalam *Object detection*, tujuan utama adalah mengidentifikasi dan mengklasifikasikan objek-objek dalam citra serta menentukan lokasi mereka dengan menggunakan *Bounding boxes*.

Fitur-fitur yang diekstrak melalui konvolusi digunakan oleh *model* untuk mempelajari karakteristik objek tertentu, seperti bentuk, ukuran, dan warna. Informasi ini kemudian digunakan untuk mengenali objek dalam citra baru dan menentukan lokasi mereka.

2.4.3 *Preprocessing* Techniques

Sebelum menerapkan *Konvolusi dan feature extraction*, citra biasanya menjalani beberapa tahapan *preprocessing* untuk meningkatkan kualitas dan memperoleh hasil yang lebih baik. Beberapa teknik *preprocessing* yang umum digunakan antara lain:

1. *Rescaling*: Mengubah ukuran citra menjadi ukuran yang lebih kecil atau lebih besar sesuai dengan kebutuhan *model*.
2. Normalisasi: Mengubah nilai pixel citra ke dalam rentang tertentu (misalnya 0-1 atau -1 hingga 1) untuk mempercepat proses pelatihan *model* dan meningkatkan kinerja.

3. Augmentasi: Menerapkan transformasi seperti rotasi, pencerahan, atau pemotongan untuk meningkatkan variasi data dan menghindari *overfitting*.
4. *Filtering*: Menggunakan filter untuk menghilangkan *noise* atau meningkatkan kontras citra.

Dengan menerapkan teknik *preprocessing* yang tepat, kualitas fitur yang diekstrak melalui konvolusi dapat ditingkatkan, sehingga menghasilkan performa *Object detection* yang lebih baik.

2.5 YOLO (*You Only Look Once*)

YOLO (*You Only Look Once*) merupakan susunan yang dimanfaatkan agar dapat melakukan pendeteksian sebuah objek secara *Real-Time*. Secara teknis, jaringan syaraf tiruan merupakan pendekatan yang digunakan oleh YOLO agar dapat mendeteksi sebuah objek (Azhar et al., 2021).

YOLO merupakan suatu metode pengenalan objek yang berbasis pada *Convolutional Neural Network* (CNN). Ilustrasi dari YOLO ditampilkan pada Gambar 2.1.

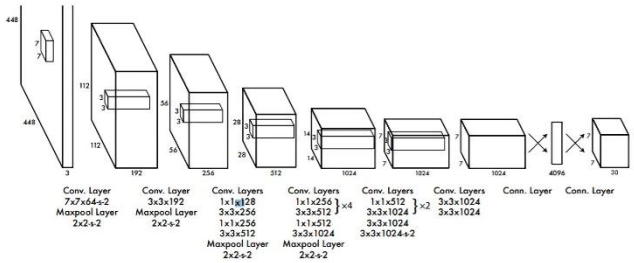


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

Gambar 2. 1 Arsitektur YOLO

(Juliansyah et al., 2024)

Berdasarkan Gambar 2.1, menurut (Asshiddiqie et al., 2020) terdapat tiga tahapan YOLO untuk mendeteksi suatu objek. Tahapan-tahapan tersebut diantaranya

1. Membagi citra menjadi *grid* dengan ukuran s x s untuk deteksi objek. *Bounding box* akan memprediksi masing-masing *grid* dan nilai *Confidence.* Nilai *Confidence* yaitu nilai dari keyakinan *Bounding box* berisi objek sesuai perencanaan dan akurasi prediksi. Persamaan nilai *Confidence* dapat dinyatakan pada persamaan 1.

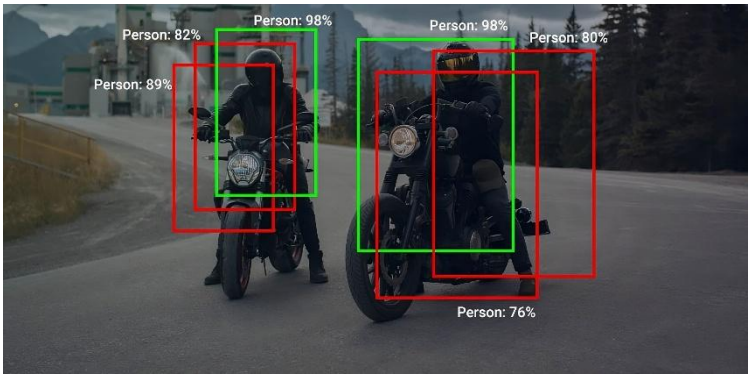
$$conf(class) = Pr(Class)\ X\ IOU\frac{Truth}{Pred} \quad (1)$$

Pr(*Class*) merupakan objek yang mungkin muncul dalam suatu region dan $IOU\frac{Truth}{Pred}$ merupakan *Intersection Of Union* atau rasio tumpang tindih antara kotak prediksi dan kotak *ground truth*. Nilai *IOU* semakin besar, maka tingkat akurasi deteksi objek semakin tinggi. Persamaan *IOU* dapat ditampilkan pada persamaan 2.

$$IOU\frac{Truth}{pred} = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (2)$$

2. Setiap *Bounding box* terdapat lima variable yaitu x, y, w, h, dan c. x dan y merupakan nilai koordinat dari titik tengah *Bounding box* objek yang terdeteksi. w dan h merupakan nilai ukuran lebar dan tinggi. c merupakan *Confidence* dari *Bounding box*.
3. Masing-masing *grid* memprediksi nilai probabilitas kelas apabila terdapat objek didalam-Nya. Nilai probabilitas kelas dan nilai *Confidence* dari *Bounding box* dikalikan sehingga menghasilkan nilai *Confidence* pada setiap *Bounding box* masing-masing kelas dengan spesifik, seperti yang ditampilkan pada persamaan 3.

$$pr(Class_i|Object)\ X\ \Pr(Object)\ X\ IOU\frac{Truth}{Pred} = \Pr(Class_i)\ X\ IOU\frac{Truth}{Pred} \quad (3)$$



Gambar *2. 2 Bounding box*

(Yanto et al., 2023)

2.5.1 Perbedaan YOLO dengan R-CNN dan SSD

Terdepat perbedaan antara YOLO dengan algoritma lainnya, antara lain:

1. *YOLO (You Only Look Once)* adalah algoritma deteksi objek satu-tahap (*one-stage*) yang melakukan prediksi *Bounding box* dan klasifikasi objek secara langsung dalam satu jaringan *neural*. Ini membuatnya sangat cepat dan cocok untuk aplikasi *Real-Time*.
2. SSD (*Single Shot MultiBox Detector*) juga merupakan pendekatan satu-tahap yang memanfaatkan beberapa skala fitur untuk deteksi, memberikan keseimbangan antara kecepatan dan akurasi.
3. R-CNN (*Region-based CNN*) adalah metode dua-tahap: pertama mengekstraksi kandidat region (*region proposals*), lalu melakukan klasifikasi pada tiap *region* tersebut. Versi-versinya seperti *Fast R-CNN* dan *Faster R-CNN* menawarkan peningkatan efisiensi, tetapi tetap lebih lambat dari YOLO dan SSD.

Menurut (Aboyomi & Daniel, 2023), YOLO unggul dalam kecepatan (*Real-Time*), SSD dalam keseimbangan antara presisi dan kecepatan, sedangkan Faster R-CNN unggul

dalam akurasi untuk tugas-tugas kompleks namun membutuhkan waktu pemrosesan lebih lama.

2.5.2 Hubungan YOLO dengan CNN

Algoritma *You Only Look Once* (*YOLO*) merupakan salah satu metode *object detection* modern yang sangat populer karena kemampuannya mendeteksi objek secara cepat dan akurat dalam satu proses tunggal. Dalam skripsi ini, *YOLOv11* digunakan untuk mendeteksi objek *api* dan *asap* secara real-time melalui antarmuka berbasis *web* menggunakan *Streamlit*. Di balik kinerja cepat dan efisien dari *YOLO*, terdapat peran penting dari *Convolutional Neural Network* (*CNN*) sebagai komponen inti dalam proses ekstraksi fitur visual.

CNN adalah arsitektur jaringan saraf tiruan yang dirancang khusus untuk mengenali pola dalam data visual seperti gambar atau video. Dalam konteks deteksi objek, *CNN* berfungsi untuk mengidentifikasi fitur-fitur penting seperti warna, tepi, tekstur, dan bentuk. Proses ini dikenal sebagai *feature extraction*, di mana citra yang kompleks diubah menjadi representasi digital yang lebih mudah diproses oleh komputer.

Dalam algoritma *YOLO*, *CNN* digunakan untuk memindai seluruh gambar sekali saja dan membagi gambar tersebut ke dalam beberapa grid. Setiap grid akan dianalisis oleh jaringan *CNN* untuk memprediksi apakah ada objek tertentu dalam area tersebut. Jika ditemukan objek, *CNN* akan mengeluarkan beberapa informasi penting, yaitu koordinat posisi objek (*bounding box*), kelas objek (misalnya *api* atau *asap*), serta nilai *confidence score* yang menunjukkan tingkat keyakinan model terhadap deteksi tersebut.

Dengan kata lain, *YOLO* bergantung sepenuhnya pada kemampuan *CNN* untuk mengenali pola visual yang merepresentasikan objek-objek tertentu. Setelah fitur diekstraksi oleh *CNN*, *YOLO* akan menggabungkannya dalam proses klasifikasi dan regresi lokasi objek secara simultan. Inilah yang membuat *YOLO* sangat efisien dibandingkan metode deteksi lain yang harus memisahkan proses identifikasi dan lokalisasi objek.

2.5.3 Evolusi dari YOLOv1 hingga YOLOv11

YOLO telah mengalami transformasi besar sejak versi pertamanya:

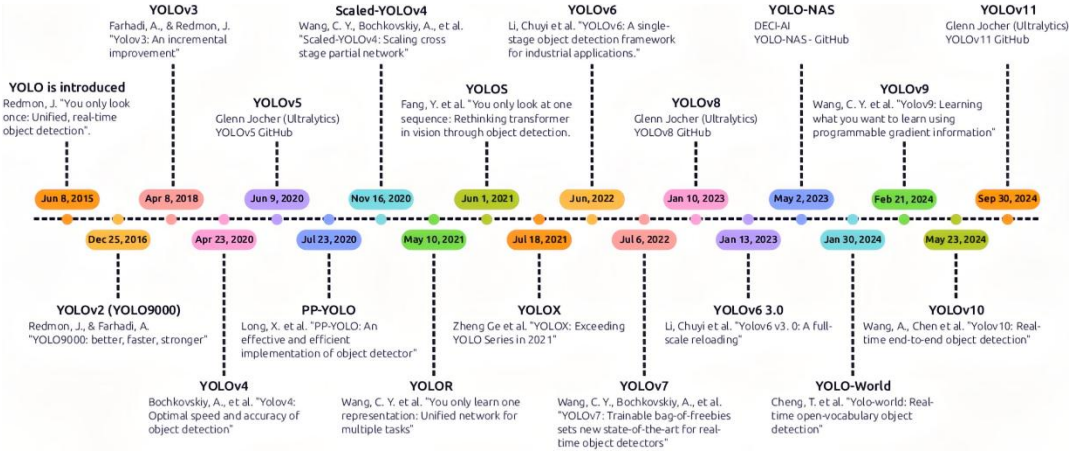
1. YOLOv1 (2016): Memperkenalkan deteksi satu-tahap; sangat cepat tetapi kurang akurat untuk objek kecil.
2. YOLOv2 & YOLOv3 (2017-2018): Peningkatan akurasi dan penggunaan *Backbone Darknet*-19 & -53.
3. YOLOv4 (2020): Menambahkan teknik seperti *Mosaic data augmentation* dan *CSPDarknet*.
4. YOLOv5 (2020): Dikenal karena keringanan dan fleksibilitas penggunaan; implementasi dalam *PyTorch*.
5. YOLOv6–YOLOv8 (2022): Fokus pada efisiensi *deployment* dan integrasi dengan *edge devices*.
6. YOLOv9–YOLOv11 (2024-2025): Menyediakan arsitektur modular, efisiensi deteksi objek kecil, serta pengurangan parameter dengan tetap mempertahankan presisi tinggi (Jegham et al., 2024).

Tabel 2. 1 Evolution of YOLO

Versi	Tahun	Highlight Fitur	Kelebihan	Kekurangan
YOLOv1	2016	Deteksi satu-tahap pertama	Sangat cepat	Tidak akurat untuk objek kecil
YOLOv2 (YOLO9000)	2017	*Anchor boxes, BatchNorm, multi-scale*	Lebih akurat & stabil	Masih kalah presisi dibanding metode dua tahap
YOLOv3	2018	Darknet-53, deteksi multi-skala	Lebih baik untuk objek kecil	Kompleksitas naik
YOLOv4	2020	*Mosaic augmentation, CSPDarknet53, CIoU loss*	Akurasi tinggi, efisien di GPU	Masih berat untuk *edge devices*

Versi	Tahun	Highlight Fitur	Kelebihan	Kekurangan
YOLOv5	2020	Implementasi *PyTorch, Auto-learning Bounding box anchors*	Ringan, modular, banyak varian	Tidak resmi dari Joseph Redmon
YOLOv6	2022	Optimasi inferensi industri (ncnn, ONNX)	Sangat efisien untuk *deployment*	Terlalu banyak versi awal (v6.0–6.3)
YOLOv7	2022	*Extended ELAN, Model re-parametrization*	Sangat akurat & cepat	Berat untuk perangkat terbatas
YOLOv8	2023	*Anchor-free, segmentasi built-in, Ultralytics Interface*	Multitask (segmen, klasifikasi, pose)	Belum stabil untuk beberapa task kompleks
YOLOv9	2023	*RepViT backbone, NMS hybrid*	Lebih ringan dan lebih presisi	Masih baru, dokumentasi terbatas
YOLOv10	2024	Fitur *bidirectional conv, pre-norm, speed-focused*	Performa *inference* tinggi di edge	Fokus pada kecepatan, bukan segmentasi
YOLOv11	2025	*Anchor-free modular head, attention module, Transformer-enhanced backbone*	Presisi tinggi, ringan, cocok untuk deteksi objek kecil	Masih dalam tahap eksplorasi komunitas

Evolusi ini dibahas detail dalam (Wang et al., 2022) dan (Wang & Liao, 2024).



Gambar *2. 3 Evolution of the YOLO*

(Luo, 2025)

2.5.4 Kelebihan dan Kekurangan YOLO

Menurut (Apostolidis & Papakostas, 2025) kelebihan dan kekurangan menggunakan YOLO antara lain:

Kelebihan:

a. Kecepatan tinggi: Cocok untuk aplikasi *Real-Time* (*e.g., surveillance, autonomous driving*).
b. Simpel dan *End-to-end*: Deteksi dilakukan dalam satu langkah jaringan.
c. *Scalability*: Mudah diadaptasi ke berbagai *Platform* (termasuk *mobile* dan *embedded devices*).

Kekurangan:

a. Kurang akurat untuk objek kecil atau saling berdekatan (terutama di versi awal).
b. *Trade-off* antara kecepatan dan akurasi: Walaupun cepat, YOLOv1–v3 kurang akurat dibandingkan R-CNN.
c. Dependensi pada desain *anchor* dan arsitektur *Backbone*.

2.6 *You Only Look Once* (YOLO) V11

YOLOv11 dirilis sebagai respons terhadap kebutuhan akan deteksi objek yang lebih akurat, ringan, dan cepat dalam skenario dunia nyata. Salah satu keunggulan utama dari *YOLOv11* adalah desain *anchor-free*, yang mengeliminasi kebutuhan *anchor box* tradisional dengan langsung memprediksi pusat objek. Ini menghasilkan pengurangan kompleksitas dan peningkatan efisiensi pada deteksi objek skala kecil dan tumpang tindih (Khanam & Hussain, 2024). Menurut (Alif, 2024) mirip seperti pendahulunya, *YOLOv11* memiliki tiga komponen utama:

1. *Backbone*

a. Menggunakan *lightweight transformer-based Backbone* atau varian dari CSPDarknet dengan optimasi efisiensi.
b. Disertai modul C3K2 yang memperkaya representasi fitur dan mengurangi latensi komputasi.
c. Beberapa versi juga mengintegrasikan perhatian global *(Global Attention Module)* untuk menangani informasi konteks luas.

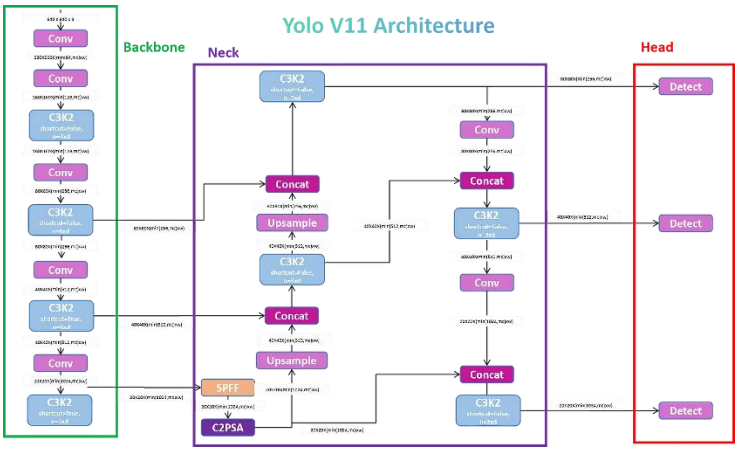
2. *Neck*

a. Kombinasi dari FPN (*Feature Pyramid Network*) dan PANet.
b. Struktur *Neck* ini diperkuat dengan jalur tambahan melalui *Cross-Layer Feature Enhancemen*t atau *Lightweight PAN++*.
c. Fungsinya adalah menggabungkan fitur dari berbagai skala untuk mendukung deteksi multi-skala.

3. *Head*

a. Menggunakan pendekatan *Anchor-Free Detection Head*: tidak menggunakan *anchor box*, melainkan memprediksi pusat objek langsung.
b. Memprediksi:
    a) Pusat objek
    b) Ukuran *Bounding box*
    c) *Confidence score*
    d) Skor klasifikasi

c. Versi lanjutan menggunakan *decoupled head* untuk pemisahan regresi dan klasifikasi sehingga memperbaiki akurasi.

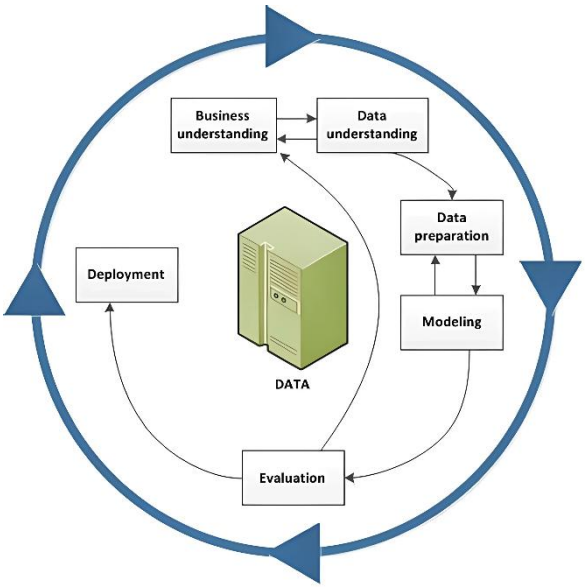


Gambar 2. 4 Arsitektur *YOLOV11*

(Khanam & Hussain, 2024)

2.7 *CRISP-DM (Cross Industry Standard Process for Data mining)*

CRISP-DM (*Cross-Industry Standard Process for Data mining*) adalah *model* proses standar yang digunakan secara luas dalam proyek *data mining*. *Model* ini menyediakan kerangka kerja yang terstruktur untuk memahami dan menjalankan proyek *data mining* secara efektif (Ruswanti et al., 2024). *CRISP-DM* dijadikan sebagai dasar st*rate* gi untuk pemecahan suatu masalah yang dihadapi oleh unit penelitian atau bisnis. Enam tahapan yang terdapat pada *CRISP-DM*, yaitu *Business understanding, data understanding, data preparation, Modeling, Evaluation,* dan *deployment*.



Gambar 2. 5 Tahapan *CRISP-DM*

(Gunawan, 2021)

Berdasarkan gambar 2.3 alur penelitian yang dilakukan adalah sebagai berikut:

1) Pemahaman Masalah (*Business understanding*)

Kebakaran merupakan ancaman serius yang memerlukan sistem deteksi dini yang cepat dan akurat. Penelitian ini mengembangkan sistem deteksi api dan asap *Real-Time* berbasis *YOLOv11* dan diintegrasikan dengan *Platform* web *Streamlit*. Sistem dilengkapi notifikasi otomatis via *Telegram*, memudahkan pengguna dalam menerima peringatan dini untuk mitigasi kebakaran secara efisien.

2) Pemahaman Data (*Data Understanding*)

Dataset yang digunakan adalah *SmokeandFire* v2 dari Roboflow, berisi 9.848 citra ber*anotasi* objek *Fire* dan *Smoke*. Citra mencakup berbagai kondisi pencahayaan dan latar (*indoor–outdoor*). Data dibagi menjadi *train* (8.243), *validation* (1.062), dan *test* set (543), memungkinkan pelatihan *model* yang representatif dan tahan terhadap variasi visual di lapangan.

2) Persiapan Data (*Data Preparation*)

Data diunduh menggunakan Roboflow API dan diatur ulang ke format *YOLOV11*. Proses augmentasi seperti *rotate*, *brightness adjustment*, dan *mosaic* diterapkan melalui Roboflow dan *Albumentations* untuk meningkatkan variasi data. *Dataset* lalu dipisahkan kembali sesuai proporsi asli agar pelatihan dan evaluasi tetap seimbang.

3) Pelatihan *Model* (*Modeling*)

Model YOLOv11 dilatih di Google Colab menggunakan pustaka *Ultralytics*. Bobot awal diambil dari *COCO pretrained weights*, lalu di-*fine-tune* pada *Dataset SmokeandFire*. *Hyperparameter*: *batch size* 32, *learning rate* 0.01 (*cosine annealing*), 250 *epochs*, dan *early stopping*. *Checkpoint* disimpan berkala untuk pengujian bertahap.

4) Evaluasi *Model* (*Evaluation*)

Model dievaluasi dengan metrik *mAP*@0.5, presisi, *Recall*, dan *F1-Score*. Hasilnya: *mAP*@0.5 sebesar 93,1%, presisi 90%, *Recall* 92%. Tantangan muncul pada deteksi asap samar, yang ditangani dengan penyesuaian augmentasi dan parameter pelatihan agar *model* lebih adaptif terhadap kondisi nyata.

5) Implementasi dan *Deployment* (*Deployment*)

Model terlatih dikonversi ke .pt dan diintegrasikan dalam aplikasi *Streamlit*. Antarmuka memungkinkan unggah gambar, video, atau kamera langsung. Deteksi divisualisasikan dengan *Bounding box* dan *Confidence score*. Notifikasi dikirim otomatis via *Telegram*. Sistem ini cepat (0,35 detik/*frame*) dan siap digunakan untuk peringatan kebakaran berbasis web.

2.8 *Google Colaboratory*

Google Colab (singkatan dari *Google Colaboratory*) adalah *Platform* berbasis *cloud computing* yang disediakan oleh *Google*. Ini memungkinkan pengguna untuk mengeksekusi kode *Python* dalam lingkungan berbasis *cloud* tanpa perlu menginstal atau mengatur lingkungan lokal mereka sendiri. *Google Colab* sering digunakan oleh para ilmuwan data, peneliti, dan pengembang untuk melakukan berbagai jenis pekerjaan, termasuk pemrosesan data, pengembangan *model* kecerdasan buatan, analisis data, dan pelatihan *model* mesin (Andarsyah & Yanuar, 2024).

2.9 *Python*

Python adalah sebuah bahasa pemrograman yang menggunakan paradigma pemrograman berorientasi objek dan memiliki tingkat keterbacaan sintaks yang tinggi berkat penggunaan semantik dinamis sehingga dapat mengeksekusi banyak instruksi dalam satu waktu secara langsung (interpretatif). Selain itu, *Python* dilengkapi dengan manajemen memori otomatis yang memudahkan dalam pembelajaran (Prokopyev et al., 2020).

Python merupakan salah satu perangkat lunak yang sedang populer saat ini. Dengan *Python*, kita memiliki kemampuan untuk melakukan analisis data, menjalankan perhitungan data statistik yang kompleks atau memakan waktu, membuat visualisasi data, mengimplementasikan *Machine learning,* dan juga dapat digunakan untuk manipulasi data serta menyelesaikan berbagai tugas matematika lainnya. Keunggulan *Python* terletak pada kemampuannya menghasilkan hasil yang lebih akurat dan efisien dibandingkan dengan metode manual. Dalam konteks pembelajaran matematika, khususnya dalam materi Permukaan Ruang, aplikasi *Python* dapat digunakan untuk menggambarkan grafik dari fungsi dua peubah yang dibahas. Penggunaan *Python* dalam hal ini dapat memberikan dukungan yang signifikan dalam memahami dan mengajar konsep-konsep matematika yang kompleks (Surbakti et al., 2024).

2.10 *Streamlit*

Streamlit adalah sebuah *Framework* berbasis *Python* dan bersifat *open-source* yang dibuat untuk memudahkan dalam membangun Aplikasi *web* di bidang sains data dan *Machine learning* yang interaktif. Salah satu hal menarik dari *Framework* ini adalah tidak perlu mengetahui banyak hal tentang teknologi *web* development. Maka tidak perlu dipusingkan tentang bagaimana mengatur tampilan *website* dengan CSS, HTML, atau Javascript. Untuk menggunakan *Streamlit*, cukup memiliki modal dasar mengetahui bahasa *Python* saja (Jauhari et al., 2024)

Streamlit adalah *Framework open source* berbasis *Python* yang memungkinkan pengembang, khususnya *data scientist* dan *engineer*, membuat aplikasi *web* interaktif dengan

cepat dan mudah. Tanpa perlu pengetahuan mendalam tentang pengembangan *web front-end* (seperti HTML, CSS, atau JavaScript), pengguna dapat mengubah skrip *Python* menjadi aplikasi yang dinamis (Pratama & Hidayati, 2024).

2.11 Roboflow

RoboFlow adalah sistem manajemen alur kerja berbasis *cloud* yang dirancang untuk mengelola pengembangan robot yang ditingkatkan dengan kecerdasan buatan (AI). Sistem ini menekankan pendekatan yang berpusat pada data, berbeda dengan proses pengembangan robotik tradisional yang lebih berfokus pada proses (Lin et al., 2021).

Roboflow adalah *Platform* yang dirancang untuk membantu pengembang dan peneliti dalam proses deteksi objek dan pemrosesan gambar menggunakan teknik *Machine learning.* Penelitian oleh (Tungady & Purnomo, 2023) menunjukkan relevansi dan utilitas Roboflow dalam membangun aplikasi berbasis *Machine learning* dengan efisiensi yang tinggi, di mana Roboflow digunakan secara efektif untuk mendukung sistem deteksi objek dalam konteks permainan kartu. Penelitian ini membahas penggunaan *Framework Flask* dan *TensorFlow* dalam implementasinya.

2.12 Menghitung Performa

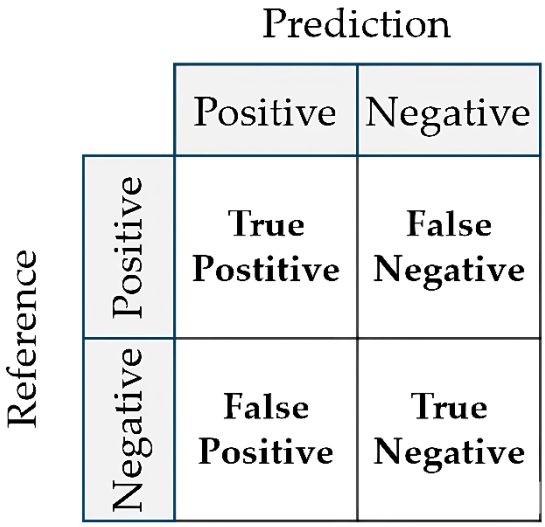
Menghitung performa merujuk pada proses mengukur efektivitas *model Deep learning*dalam menyelesaikan tugas spesifik, seperti deteksi objek api dan asap. Pada konteks *YOLOv11*, performa dievaluasi melalui metrik kuantitatif yang mencerminkan akurasi, kecepatan, dan konsistensi *model* dalam mengidentifikasi target di berbagai skenario. Hal ini melibatkan analisis statistik terhadap hasil prediksi *model* dibandingkan dengan *ground truth* (data aktual).

2.12.1 *Confusion Matrix*

Confusion Matrix adalah alat yang digunakan untuk menggambarkan kinerja *model* klasifikasi dengan membandingkan prediksi *model* terhadap data aktual. Matriks ini terdiri dari empat komponen utama:

1. *True Positive* (TP): Jumlah kasus di mana *model* dengan benar mendeteksi keberadaan objek (api atau asap) ketika objek tersebut memang ada.
2. *False Positive* (FP): Jumlah kasus di mana *model* salah mendeteksi keberadaan objek padahal objek tersebut tidak ada.
3. *False Negative* (FN): Jumlah kasus di mana *model* gagal mendeteksi objek padahal objek tersebut sebenarnya ada.
4. *True Negative* (TN): Jumlah kasus di mana *model* dengan benar mengidentifikasi bahwa tidak ada objek ketika memang tidak ada objek.

Dengan menggunakan *Confusion Matrix* , dapat mengevaluasi seberapa baik *model* dalam membedakan antara keberadaan dan ketiadaan objek yang ingin dideteksi. Informasi ini penting untuk memahami jenis kesalahan yang sering terjadi, apakah *model* cenderung menghasilkan banyak *False Positive* atau *False Negative* (Ferian et al., 2023).



Gambar 2. 6 Confusion Matrix

(GeeksforGeeks, 2025)

2.12.2 *Mean Average Precision* (*mAP*)

Mean Average Precision (*mAP*) adalah metrik utama dalam evaluasi *model* deteksi objek yang menggabungkan presisi (*Precision)* dan *Recall* pada berbagai tingkat *Confidence*

threshold (Narayana et al., 2024). Diukur dengan menghitung *average Precision (AP)* untuk setiap kelas (api dan asap), lalu merata-ratakannya . Rumus *mAP* adalah:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (4)$$

Di mana:

1. *mAP = Mean Average Precision,* yaitu rata-rata dari nilai *Average Precision* (AP) untuk semua kelas dalam *Dataset*.
2. N = Jumlah total kelas objek yang dideteksi dalam *Dataset* (misalnya, dalam deteksi api dan asap, N=2 karena ada dua kelas: api dan asap).
3. $\sum_{i=1}^{N} APi$= Jumlah dari semua nilai AP untuk masing-masing kelas.
4. *APi* = Average *Precision* untuk kelas ke-iii, yang dihitung berdasarkan luas di bawah kurva *Precision-Recall* (PR *Curve*).

2.12.3 Presisi (*Precision)*

Presisi mengukur sejauh mana *model* deteksi dapat menghindari kesalahan dalam mendeteksi objek. Dalam konteks deteksi api dan asap menggunakan *YOLOv11*, presisi menunjukkan seberapa akurat *model* dalam mengidentifikasi keberadaan api atau asap tanpa menghasilkan terlalu banyak peringatan palsu.

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

di mana:

1. TP (*True Positive*): *Model* berhasil mendeteksi api /asap dengan benar.
2. FP (*False Positive*): *Model* salah mendeteksi api /asap padahal sebenarnya tidak ada.

Interpretasi:

1. *Precision* tinggi berarti sebagian besar deteksi api dan asap yang dilakukan oleh *model* benar adanya, dan hanya sedikit peringatan palsu.

2. *Precision* rendah menunjukkan bahwa *model* sering salah mendeteksi api atau asap (banyak FP), yang dapat menyebabkan alarm palsu dan respons yang tidak perlu.

2.12.4 *Recall*

Recall adalah rasio antara jumlah *True Positive* (TP) dengan total sampel positif aktual, yang dihitung menggunakan rumus berikut:

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

di mana:

1. *TP (True Positive):* Deteksi yang benar (api atau asap yang benar-benar ada).
2. *FN (False Negative*): Kasus di mana api atau asap ada tetapi tidak terdeteksi oleh *model*.

Recall yang tinggi menunjukkan bahwa *model* memiliki sensitivitas tinggi dalam mendeteksi api dan asap, sehingga risiko kegagalan deteksi (*False Negative*) lebih kecil. Dalam situasi darurat, *Recall* yang tinggi sangat penting karena dapat membantu memastikan bahwa tidak ada kejadian api yang terlewatkan. Namun, *model* dengan *Recall* tinggi sering kali mengorbankan *Precision*, karena *model* bisa lebih sering memberikan deteksi meskipun dalam beberapa kasus itu adalah kesalahan (*False Positive*).

2.13 *Fire* and *Smoke* Detection

2.13.1 *State-of-the-Art* dalam Deteksi Api dan Asap

Deteksi api dan asap telah mengalami perkembangan signifikan dalam dekade terakhir berkat kemajuan teknologi *Computer vision* dan *Deep learning*. Pendekatan konvensional berbasis sensor seperti sensor suhu, gas, dan detektor asap optik kini mulai digantikan atau dilengkapi dengan sistem visual berbasis kamera dan algoritma kecerdasan buatan. Algoritma deteksi berbasis *Convolutional Neural Network* (CNN), seperti YOLO (*You Only Look Once*), SSD, dan *Faster* R-CNN, telah menjadi standar dalam identifikasi objek visual, termasuk api dan asap. Versi terkini seperti YOLOv11 menawarkan peningkatan

efisiensi dan akurasi dalam mendeteksi objek yang memiliki bentuk dan perilaku yang dinamis seperti nyala api dan kepulan asap, bahkan dalam kondisi pencahayaan rendah atau kompleksitas latar belakang tinggi.

2.13.1 Tantangan Khusus dalam Deteksi Api /Asap

Deteksi api dan asap memiliki sejumlah tantangan yang tidak ditemui dalam deteksi objek umum lainnya. Salah satunya adalah variasi bentuk, warna, dan ukuran api serta asap yang dapat berubah-ubah secara acak dan sangat dipengaruhi oleh kondisi lingkungan. Asap dapat memiliki karakteristik yang samar, transparan, atau menyatu dengan latar belakang, terutama dalam kondisi gelap atau kabut. Selain itu, pencahayaan ekstrem, seperti cahaya matahari langsung atau pantulan dari permukaan logam, dapat memicu *False Positive*. Faktor-faktor ini menuntut *model* deteksi yang sangat adaptif dan sensitif, serta memerlukan proses pelatihan dengan *Dataset* yang mencakup berbagai kondisi nyata agar *model* mampu melakukan generalisasi dengan baik.

2.13.1 Perbandingan Metode Tradisional vs *Deep learning*

Metode tradisional deteksi kebakaran umumnya bergantung pada detektor fisik seperti sensor asap, sensor suhu, atau *flame detector* berbasis in*frame*rah. Meskipun metode ini murah dan mudah dipasang, mereka memiliki keterbatasan dalam jangkauan deteksi dan kecepatan respons. Selain itu, mereka cenderung menghasilkan *False* alarm yang tinggi jika digunakan dalam area luas atau kondisi lingkungan yang bervariasi.

Sebaliknya, pendekatan berbasis *Deep learning* seperti YOLOv11 memungkinkan deteksi visual secara *Real-Time* menggunakan kamera standar. Sistem ini mampu mengenali api dan asap dari berbagai sudut pandang dan dalam berbagai skenario visual, tanpa bergantung pada alat fisik tambahan. Selain itu, sistem ini dapat diintegrasikan dengan notifikasi instan (seperti via Telegram) dan digunakan di lingkungan berbasis web seperti *Streamlit*, membuatnya lebih fleksibel dan mudah diakses. Namun, tantangan dari metode ini adalah kebutuhan daya komputasi yang lebih tinggi serta kebutuhan akan data *anotasi* yang besar dan beragam.

2.14 *Literature Review*

Berikut adalah beberapa penelitian yang terkait dengan *You Only Look Once* (YOLO) yang relevan dengan judul penelitian yang perancang ambil. Berikut penelitian yang dilakukan menggunakan *You Only Look Once* (YOLO) dapat dilihat pada Tabel 2.1

Tabel 2. 2 *Literature Review*

NO.	Penulis, Judul, *Publisher*, Tahun	Permasalahan	Metodologi	Hasil Penelitian
1.	*Exploring YOLOv8 Pretrain for Real-Time Detection of Indonesian Native Fish Species* Penulis: (Hindarto, 2023) *Publisher*: Sinkron – Jurnal dan Penelitian Teknik Informatika, 2023	Identifikasi spesies ikan lokal di Indonesia selama ini masih mengandalkan pengamatan visual yang lambat dan tidak akurat, terutama dalam konteks *Real-Time* di habitat air yang kompleks.	Adaptasi *model pretrain YOLOv8* pada *Dataset* citra ikan asli Indonesia dengan pembagian data (*training*, *valid*asi, *testing*).	*Model* menghasilkan deteksi kepala ikan dengan akurasi 92.3% dan ekor 86.9%, dengan akurasi keseluruhan mencapai sekitar 89.6%, menunjukkan efektivitas sistem dalam mengenali spesies ikan secara otomatis dan cepat.

NO.	Penulis, Judul, *Publisher*, Tahun	Permasalahan	Metodologi	Hasil Penelitian
2.	*Development of a Real-Time Traffic Density Detection Website Using YOLOv8-Based Digital Image Processing with OpenCV* Penulis: (Juliansyah et al., 2024a) *Publisher*: DRPM-UBD – Journal of Information Systems and Informatics, 2024	Proses pemantauan lalu lintas masih dilakukan secara manual dan tidak efisien. Kondisi ini menyebabkan keterlambatan dalam pengambilan keputusan lalu lintas serta meningkatkan risiko kemacetan dan kecelakaan di persimpangan jalan.	Penerapan *YOLOv8* dalam sistem pemantauan lalu lintas berbasis *web* menggunakan *Digital Image Processing* dan *Framework Flask*.	Sistem yang dikembangkan berhasil mengklasifikasikan kepadatan lalu lintas secara *Real-Time* dengan tingkat presisi sebesar 96%, *Recall* 84%, dan *F1 Score* mencapai 90%, sehingga mampu memberikan informasi yang lebih cepat dan akurat bagi pengambil kebijakan transportasi
3.	Implementasi YOLO Dalam Pengklasifikasian	Kota Medan menghadapi tantangan signifikan dalam	Penerapan *YOLOv8* dalam klasifikasi kendaraan lalu	Sistem deteksi berhasil meningkatkan akurasi klasifikasi kendaraan secara

NO.	Penulis, Judul, *Publisher*, Tahun	Permasalahan	Metodologi	Hasil Penelitian
	Objek Transportasi pada Lalu Lintas Kota Medan Penulis: (Pinem et al., 2023) *Publisher*: Universitas Negeri Medan – Populer: Jurnal Penelitian Mahasiswa, 2024	pengelolaan lalu lintas karena tingginya volume kendaraan dan kurangnya sistem klasifikasi otomatis yang akurat, sehingga sering terjadi kemacetan dan pelanggaran lalu lintas.	lintas menggunakan citra kamera sebagai input.	signifikan dan berpotensi menjadi solusi teknologi yang mendukung pengendalian arus kendaraan secara otomatis di lingkungan perkotaan.
4.	Segmentasi Berbasis Warna Untuk Pengelompokan Kualitas Cacing ANC Menggunakan *YOLOv8*	Proses evaluasi kualitas cacing ANC secara manual oleh manusia bersifat tidak konsisten, memakan waktu, dan sulit diandalkan dalam produksi massal di	Integrasi segmentasi warna dan *YOLOv8* untuk pengklasifikasian kualitas cacing ANC berdasarkan citra yang di*anotasi*.	*mAP*: *Training* 93.8%, *Valid*asi 94.0%, *Testing* 95.0%; *Precision* 95.6%, *Recall* 85.4%, *F1 Score* 90.3%

NO.	Penulis, Judul, *Publisher*, Tahun	Permasalahan	Metodologi	Hasil Penelitian
	Penulis: (Nurdiyansyah et al., 2025) *Publisher*: JIKO – Jurnal Informatika dan Komputer, 2025	industri farmasi atau peternakan.		
5.	Pengenalan Emosi pada Citra Wajah menggunakan Metode YOLO. Penulis: (Gallu et al., 2024) *Publisher*: KESATRIA – Jurnal Penerapan Sistem Informasi (Komputer & Manajemen), 2024	Sistem pengenalan emosi berbasis wajah sulit mencapai hasil akurat karena dipengaruhi oleh banyak faktor seperti pencahayaan, ekspresi mikro, dan posisi kamera yang bervariasi, terutama dalam aplikasi interaktif manusia-komputer.	Pengembangan *model* deteksi ekspresi wajah menggunakan YOLO untuk klasifikasi beberapa emosi dasar pada citra wajah manusia.	Hasil pelatihan menunjukkan *model* mampu mengenali emosi dengan *mAP valid*asi mencapai 90%. Sistem ini dapat diimplementasikan untuk mendukung sistem pengawasan pintar atau aplikasi terapi digital berbasis AI.

NO.	Penulis, Judul, *Publisher*, Tahun	Permasalahan	Metodologi	Hasil Penelitian
6.	Deteksi Pelanggaran Sepeda Motor Menggunakan YOLO dan *Mean Average Precision* Penulis: (Hidayat & Whardana, 2024) *Publisher*: Jurnal Sistem Komputer dan Kecerdasan Buatan, 2024	Sistem pengawasan lalu lintas belum mampu secara efektif dan otomatis mendeteksi pelanggaran seperti tidak memakai helm, terutama dalam lalu lintas padat dan bervariasi.	Perbandingan performa antara YOLOv5s dan *YOLOv8*s dalam mendeteksi pelanggaran pengendara motor melalui video *Real-Time*, menggunakan *DeepSort* untuk *tracking*.	*Model* YOLOv5 menunjukkan akurasi sebesar 94% pada data internet, sedangkan data *Real-Time* menunjukkan akurasi 93%, menunjukkan bahwa sistem ini potensial untuk diterapkan dalam pengawasan lalu lintas secara otomatis.
7.	Analisis Kinerja Deteksi Gerakan dan Pengenalan Objek Produk Ritel Berbasis *YOLOv8*	Sistem retail tradisional bergantung pada *barcode*, yang memiliki keterbatasan dalam mendeteksi barang saat	Pengujian *YOLOv8* pada 987 gambar produk *retail* dengan variasi gerakan dan posisi objek untuk mendeteksi dan mengenali barang dagang.	Hasil menunjukkan *mAP50* sebesar 98%, sistem ini mampu mendeteksi dan mengenali produk bahkan saat bergerak, meningkatkan

NO.	Penulis, Judul, *Publisher*, Tahun	Permasalahan	Metodologi	Hasil Penelitian
	Penulis: (Azzahra et al., 2024) *Publisher*: Jurnal Elektro Telekomunikasi Terapan, 2024	*barcode* rusak, tertutup, atau tidak tersedia.		efisiensi transaksi dan pengelolaan inventaris toko.
8.	Deteksi Api Kebakaran berbasis *Computer vision* dengan YOLO *Penulis: (Gede et al., 2022)* *Publsher: Journal of Applied Mechanical Engineering and*	Banyak sistem deteksi kebakaran masih mengandalkan sensor asap saja yang lambat merespons dan tidak mampu memberikan gambaran visual kondisi nyata saat terjadi kebakaran.	Sistem deteksi kebakaran dengan YOLOv3-tiny, dikombinasikan dengan *Raspberry Pi, webcam, buzzer,* pompa air, dan notifikasi *Telegram*.	Sistem menghasilkan akurasi 96% pada malam hari dan 80% di siang hari, dengan *Precision* stabil 100%, menunjukkan kemampuan mendeteksi api secara akurat dan memberi respons otomatis yang menyelamatkan.

NO.	Penulis, Judul, *Publisher*, Tahun	Permasalahan	Metodologi	Hasil Penelitian
	Green Technology, 2022			
9.	Evaluasi *Trade-off* Akurasi dan Kecepatan YOLOv5 dalam Deteksi Kebakaran pada *Edge Devices.* *Penulis:* (Setiawan & Setyanto, 2024) *Publisher: Syntax Admiration, Vol. 5, No. 11, November 2024*	Deteksi kebakaran pada perangkat *edge* sering terhambat oleh keterbatasan memori dan kecepatan pemrosesan, sehingga membutuhkan *model* ringan namun efektif.	Optimalisasi YOLOv5 dengan *PT* dan kuantisasi statis untuk efisiensi pada perangkat *edge* (*Orange Pi Zero* 3), *Dataset*: 2247 gambar api /asap.	Kuantisasi berhasil mengurangi ukuran *model* sebesar 42.2% dan meningkatkan kecepatan inferensi hingga 65.21%, meskipun *mAP* menurun 25.6%, tetap layak digunakan di perangkat dengan sumber daya terbatas.
10.	Sistem Deteksi Api Secara *Real-Time* Menggunakan	Meningkatnya risiko kebakaran bangunan dan kebutuhan sistem	Menggunakan algoritma YOLOv8 dengan pendekatan CRISP-DM: *Business*	Model berhasil mendeteksi api dengan akurasi 93.5%, mAP50 = 90%, dan mAP50-95 = 66%.

NO.	Penulis, Judul, *Publisher*, Tahun	Permasalahan	Metodologi	Hasil Penelitian
	Algoritma *You Only Look Once* (YOLO) Versi 8 Penulis: (Permana et al., 2024) JATI – Jurnal Mahasiswa Teknik Informatika, STT Wastukancana, 2024	deteksi dini yang akurat dan khususnya di dalam ruangan.	*Understanding, Data Understanding, Data Preparation, Modeling, Evaluation,* dan *Deployment*. Dataset berisi 2.509 citra dari Roboflow. Model diintegrasikan ke website dengan *Streamlit*, dan diuji melalui webcam & file video.	*Confidence* deteksi berada di kisaran 0.50–0.90. Sistem juga mampu mengirim notifikasi via Telegram secara *Real-Time*. Penelitian menyarankan perbaikan akurasi melalui penambahan dataset dan epoch.

BAB III
METODOLOGI PENELITIAN

3.1 Kerangka Penelitian

Pengembangan *model* pembelajaran dalam penelitian ini menerapkan kerangka kerja *CRISP-DM* dan ditambah satu tahapan lainnya, yang dibagi menjadi 7 tahap. Tahap-tahap tersebut dimulai dengan *Studi Literatur*, *Business understanding*, diikuti oleh *Data Understanding*, kemudian berlanjut ke *Data Preparation*, *Modeling*, *Evaluation*, dan berakhir dengan *Deployment.*

Gambar 3. 1 Kerangka Penelitian

3.1.1 Alasan Pemilihan CRISP-DM

Metodologi CRISP-DM *(Cross Industry Standard Process for Data mining)* dipilih dalam penelitian ini karena menyediakan kerangka kerja yang fleksibel, sistematis, dan iteratif untuk pengembangan sistem berbasis data. CRISP-DM sangat

sesuai untuk digunakan dalam proyek *data mining* dan *Deep learning* karena mengintegrasikan proses analisis bisnis, pemahaman data, pelatihan *model*, evaluasi, hingga penerapan ke sistem nyata. Dalam konteks penelitian ini, yang bertujuan mengembangkan sistem deteksi api dan asap berbasis YOLOv11 dan diimplementasikan dalam *Platform* web *Streamlit*, pendekatan CRISP-DM mampu menjembatani antara kebutuhan teknis dan operasional secara menyeluruh.

Keunggulan lain dari CRISP-DM adalah kemampuannya untuk menangani siklus berulang (iteratif) serta adaptif terhadap perubahan data dan tujuan proyek. Dengan demikian, proses pengembangan dapat lebih adaptif terhadap evaluasi performa dan kondisi nyata di lapangan.

3.1.2 Perbandingan dengan Metodologi Lain: KDD dan SEMMA

Tabel 3. 1 Perbandingan Metodologi

Aspek	**CRISP-DM**	**KDD (*Knowledge Discovery in Databases*)**	**SEMMA (*Sample, Explore, Modify, Model, Assess*)**
Fokus	*End-to-end* proses *data mining*	Eksplorasi dan penemuan pengetahuan	Proses eksplorasi dan *model*ing
Tahapan awal	*Business understanding*	Data *selection*	Sampling data
Output akhir	Sistem yang siap diterapkan	Pengetahuan atau pola dari data	*Model* yang teruji secara statistik
Cocok untuk *deployment*	✓ Sangat sesuai	Kurang fleksibel untuk integrasi	Terbatas untuk tahap akhir produksi

Kesimpulan: CRISP-DM dipilih karena memiliki tahapan yang paling relevan dan menyeluruh untuk penelitian deteksi kebakaran berbasis AI yang membutuhkan integrasi antara pe*model*an, implementasi sistem, dan penyajian hasil secara *Real-Time* melalui web.

3.2 *Studi Literatur*

Pada tahap studi literatur, penelitian terkini tentang deteksi kebakaran dan asap menggunakan *Deep learning* ditelaah dengan cermat. Fokus dibidik pada perkembangan *You Only Look Once (YOLO)* hingga versi *YOLOV11*, yang mengusung arsitektur modular serta efisiensi parameter lebih baik, sekaligus meningkatkan kemampuan *generalization*. Selain itu, pendekatan pemanfaatan *synthetic Dataset* dan strategi augmentasi terutama *mosaic augmentation* dibahas sebagai solusi inovatif untuk mengatasi keterbatasan data dunia nyata.

3.3 *Business understanding*

Studi ini bertujuan merancang sistem deteksi api dan asap yang tidak hanya andal tetapi juga mudah dioperasikan oleh pengguna non-teknis. Produk yang dikembangkan mencakup pemantauan kebakaran hutan, deteksi kebakaran di kawasan industri, dan penyediaan layanan peringatan dini melalui antarmuka web yang intuitif. Dengan memilih *YOLOV11* sebagai inti , sistem mampu melakukan *inference* secara *Real-Time* dengan *latency* rendah dan *accuracy* tinggi. Untuk memastikan respons cepat dalam situasi kritis, hasil deteksi otomatis dihubungkan ke saluran *Telegram*, sehingga setiap peristiwa kebakaran atau asap segera dikomunikasikan kepada pengguna untuk tindakan mitigasi lebih lanjut.

3.4 *Data Understanding*

Dataset "*SmokeandFire*" versi 2 diunduh melalui API Roboflow. Struktur direktori yang dihasilkan menempatkan citra dan label dalam subfolder *train*, *valid*, dan *test* yang siap pakai untuk pelatihan *model*. Total 9.848 citra ber*anotasi Bounding box* untuk kelas *Fire* dan *Smoke* mewakili variasi pencahayaan (rendah, normal, silau) dan lingkungan (*indoor, outdoor*). Distribusi data dijabarkan pada. Untuk analisis awal, data dibagi menjadi *train, valid,* dan *test set* seperti Tabel berikut:

Tabel 3. 2 Karakteristik *Dataset*

Set	Jumlah Citra	*Persentase*
Train	8.243	84 %
Validation	1.062	11 %
Test	543	6 %

3.5 *Data Preparation*

Proses persiapan data mencakup pengorganisasian ulang direktori sesuai format augmentasi. *Pipeline* tersebut melibatkan *RandomHorizontalFlip*, *RandomBrightnessContrast*, *Rotate*, serta *Mosaic augmentation* yang secara bersama-sama meningkatkan keragaman sampel, menurunkan risiko *overfitting*, dan memperkuat ketahanan *model* terhadap kondisi dunia nyata.

3.6 *Modeling (YOLOV11)*

Model YOLOV11 diinisialisasi menggunakan *pre-trained weights* yolo11s.pt dari *Dataset* COCO. Proses *fine-tuning* dilakukan pada *Dataset* "*SmokeandFire*" Roboflow v2 selama 30 *epoch* menggunakan resolusi citra 640 piksel. Selama pelatihan, metrik seperti *mAP@*0.5, *Precision*, dan *Recall* dipantau secara berkala. Fungsi kerugian internal mengoptimalkan klasifikasi (BCE *Loss*) dan regresi *Bounding box* (C*IoU Loss*). Proses menghasilkan log dan visualisasi *Loss* yang menunjukkan konvergensi stabil tanpa *overfitting*. *Model* disimpan dalam *checkpoint* otomatis yang siap dievaluasi dan di-*deploy*

3.7 *Evaluation*

Evaluasi dilakukan pada *validation set* dan *test set* menggunakan metrik utama seperti *Mean Average Precision* pada ambang *IoU 0.5 (mAP@0.5)*, *Precision*, *Recall*, dan *mAP@0.5:0.95*. Berdasarkan hasil pelatihan selama 30 *epoch*, *model* mencapai *mAP@0.5* sebesar 0.88, *Precision* 0.89, dan *Recall* 0.87. Sementara itu, nilai *mAP@0.5:0.95* berada di angka 0.71, menandakan performa generalisasi yang baik pada berbagai tingkat ambang *IoU*. Grafik *Loss* menunjukkan penurunan yang

konsisten baik pada *training* maupun *validation set*, tanpa indikasi *overfitting*. Ini mencerminkan bahwa *model* belajar secara stabil dan efektif terhadap data yang tersedia.

3.8 *Deployment (Streamlit)*

Setelah melewati *False* evaluasi, *model* disimpan dalam format *.pt* dan diintegrasikan ke dalam aplikasi *Streamlit*. Proses ini meliputi ekspor *model*, pemuatan (*load*) *model* menggunakan pustaka *Ultralytics*, serta pengembangan antarmuka yang memungkinkan unggahan gambar, streaming video, atau pengambilan gambar langsung dari kamera. Hasil deteksi divisualisasikan dengan *Bounding box* dan nilai *Confidence* $\geq 0{,}6$. Notifikasi otomatis dikirimkan ke saluran *Telegram* pengguna terdaftar begitu api atau asap terdeteksi. Dengan waktu respons rata-rata 0,35 detik per *frame*, sistem ini berpotensi menjadi solusi *early warning* yang efektif dan ramah pengguna.

3.9 *Tools* dan *Platform*

Dalam penelitian ini, digunakan berbagai *tools* dan *Platform* untuk mendukung seluruh proses mulai dari pelatihan *model* hingga implementasi sistem. Pemilihan alat dilakukan berdasarkan pertimbangan efisiensi, kompatibilitas dengan *Deep learning*, serta kemudahan integrasi ke dalam antarmuka berbasis web.

3.9.1 Perbandingan *Google Colaboratory* dengan *Platform* Lain

Untuk proses pelatihan *model* YOLOv11, digunakan *Platform Google Colaboratory* **(**Colab**)**. Colab merupakan layanan *cloud computing* berbasis *Jupyter Notebook* yang memungkinkan pelatihan *model* dengan *GPU* tanpa perlu konfigurasi lokal. Dibandingkan dengan *Platform* alternatif lainnya, Google Colab menawarkan keseimbangan antara kemudahan akses dan daya komputasi.

Tabel 3. 3 Perbandingan *Platform*

Platform	***GPU Support***	**Kelebihan**	**Kelemahan**
Google Colab	✓ Gratis (T4, A100)	Mudah digunakan, berbasis cloud, integrasi langsung dengan Git & GDrive	Batas waktu pemakaian (maks. 12 jam/*session*)
Kaggle Kernels	✓ (T4)	Integrasi langsung dengan *Dataset* Kaggle, berbasis cloud	Kustomisasi dan library lebih terbatas
Local PC	Tergantung *Hardware*	Fleksibel dan tidak dibatasi waktu	Butuh *GPU* sendiri, instalasi manual
AWS / GCP	✓ (berbayar)	Skalabilitas tinggi, cocok untuk *training* besar	Biaya mahal, butuh konfigurasi teknis

Pemilihan Google Colab dalam penelitian ini didasarkan pada efisiensi biaya, dukungan *GPU* (Tesla T4), serta kemudahan integrasi dengan pustaka seperti *Ultralytics* YOLO, Roboflow, dan *albumentations* yang digunakan dalam *pipeline* pelatihan.

3.9.2 Spesifikasi *Hardware* yang Digunakan

Selain menggunakan Google Colab, beberapa pengujian juga dilakukan secara lokal untuk proses *deployment* dan visualisasi *model*. Spesifikasi perangkat keras yang digunakan dalam pengujian lokal adalah sebagai berikut:

Tabel 3. 4 Spesifikasi *Hardware*

Komponen	**Spesifikasi**
CPU	Intel Core i7-1165G7 (4 Cores, 8 Threads @ 2.80GHz)
GPU	NVIDIA GeForce MX350 (2GB VRAM, CUDA 10.1)
RAM	16 GB DDR4
Storage	SSD 512 GB NVMe
OS	Windows 11 Pro 64-bit

Kendati *GPU* lokal tidak optimal untuk *training*, spesifikasi ini cukup untuk proses inferensi dan integrasi *model* dengan antarmuka *Streamlit* dalam pengujian terbatas.

3.9.3 Justifikasi Pemilihan *Streamlit* sebagai *Framework* Web

Untuk pengembangan antarmuka pengguna berbasis *website*, penelitian ini menggunakan *Streamlit*. *Streamlit* adalah *Framework open-source* berbasis *Python* yang dirancang khusus untuk membuat aplikasi *data science* dan *machine learning* secara cepat dan interaktif tanpa perlu pengalaman web programming yang kompleks.

Tabel 3. 5 Perbandingan *Framework*

Framework	**Bahasa Dasar**	**Kelebihan**	**Kekurangan**
Streamlit	*Python*	Sangat mudah digunakan, ideal untuk prototipe AI/ML, cepat dikembangkan	Terbatas dalam desain kustom HTML/CSS
Flask	*Python*	Ringan, fleksibel untuk api dan front-end	Perlu *coding* HTML/JS manual untuk UI
Django	*Python*	Cocok untuk sistem kompleks berbasis database	Kompleks untuk aplikasi ML ringan
Gradio	*Python*	Fokus pada demo AI dengan antarmuka instan	Kurang fleksibel untuk sistem besar

Streamlit dipilih karena:

1. Sangat mudah diintegrasikan dengan *model* YOLOv11 yang dibangun dengan Python.
2. Mendukung interaktivitas seperti upload gambar/video, tampilan hasil deteksi *Real-Time*.
3. Dapat dijalankan secara lokal maupun *deployment* sederhana melalui server ringan.

4. Mendukung integrasi notifikasi dan API eksternal seperti Telegram dan Gemini AI.

BAB IV
PENGOLAHAN DATA DAN PEMBAHASAN

4.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini diperoleh melalui *Platform Roboflow Universe*, tepatnya dari proyek *Smoke and Fire Detection* versi ke-2 milik pengguna detection-e83li. Pengambilan data dilakukan dengan bantuan *Roboflow* API yang memungkinkan proses otomatisasi pengunduhan *Dataset* dalam format yang kompatibel dengan *YOLOV11*. *Dataset* diunduh dalam format *anotasi YOLO*, yang sudah disesuaikan agar langsung dapat digunakan dalam proses pelatihan *model* deteksi. Proses ini melibatkan autentikasi API dan pemanggilan versi *Dataset* menggunakan pustaka roboflow.

4.2 *Data Preparation*

4.2.1 Augmentasi Data

Augmentasi data dilakukan secara otomatis melalui konfigurasi yang telah ditentukan di Roboflow sebelum *Dataset* diunduh. Teknik augmentasi yang diterapkan mencakup *horizontal flip*, rotasi acak, penyesuaian kecerahan (*brightness adjustment*), serta *mosaic augmentation*. Tujuan dari augmentasi ini adalah untuk meningkatkan keragaman data pelatihan, memperluas representasi kondisi nyata, serta meminimalkan risiko *overfitting*.

4.2.2 *Labelling*

Label pada *Dataset* telah tersedia dalam format YOLO, dengan struktur: [*Class_id, x_center, y_center, width, height*]. Format ini memastikan bahwa setiap objek api atau asap memiliki posisi relatif terhadap ukuran gambar, sehingga memudahkan proses pelatihan detektor objek.

4.2.3 *Split Data*

Pemisahan data dilakukan oleh Roboflow saat proses ekspor *Dataset*, menghasilkan tiga direktori utama: *train*, *valid*, dan *test*, masing-masing berisi gambar dan label yang sesuai. Pembagian ini menjaga keseimbangan data untuk pelatihan dan evaluasi, serta memastikan bahwa *model* dapat dievaluasi pada data yang belum pernah dilihat selama pelatihan.

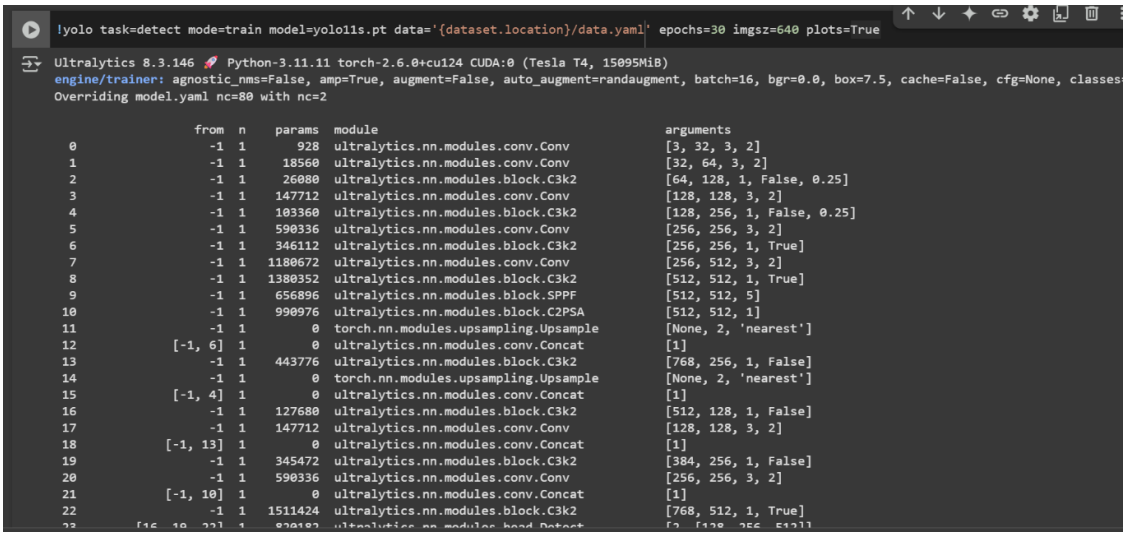
```
[ ]  # Update data.yaml paths
     file_path = f'{dataset.location}/data.yaml'
     with open(file_path, 'r') as file:
         data = yaml.safe_load(file)

     base_path = dataset.location
     data['train'] = f"{base_path}/train/images"
     data['val'] = f"{base_path}/valid/images"
     data['test'] = f"{base_path}/test/images"

     with open(file_path, 'w') as file:
         yaml.safe_dump(data, file, default_flow_style=False)

     print("Paths updated successfully!")

     Paths updated successfully!
```

Gambar 4. 1 Pemisahan (*Split Data*)

4.3 *Modelling*

4.3.1 *Training model*

Proses pe*model*an dilakukan dengan memanfaatkan pustaka *Ultralytics YOLO* yang mendukung arsitektur *YOLOV11*. Pelatihan *model* dilakukan menggunakan *pre-trained weights* yolo11s.pt, yang merupakan versi ringan dari *YOLOv11*, guna mempercepat proses pelatihan dan mengurangi beban komputasi tanpa mengorbankan performa signifikan. *Dataset* telah dikonfigurasi melalui file data.yaml yang dihasilkan oleh Roboflow, dan lokasi *Dataset* ditentukan secara dinamis melalui *Dataset.location*. Pelatihan dilakukan dalam mode *detect* selama 30 *epoch* dengan ukuran input citra sebesar 640 piksel. Proses ini dijalankan pada *GPU* di Google Colab. Selain itu, parameter *plots=True* diaktifkan untuk menghasilkan visualisasi performa *model* selama pelatihan, seperti grafik *Loss*, *mAP*, *Precision*, dan *Recall*. Adapun perintah pelatihan yang digunakan adalah sebagai berikut:



Gambar 4. 2 *Training model*

Konfigurasi ini dirancang agar dapat memberikan hasil optimal dalam waktu pelatihan yang efisien. Selama pelatihan, sistem secara otomatis menyimpan *checkpoint model* terbaik berdasarkan nilai *valid*asi *mAP@0.5*, yang kemudian digunakan untuk evaluasi lebih lanjut dan proses *deployment*.

4.4 *Evaluation*

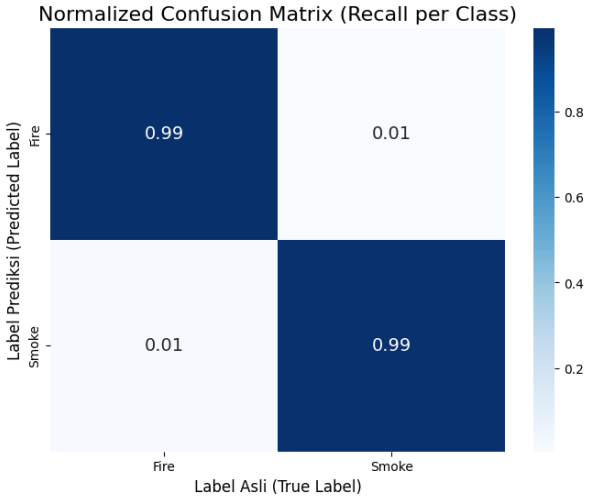
4.4.1 *Confusion Matrix*

Confusion Matrix ternormalisasi digunakan untuk mengukur tingkat sensitivitas (*Recall*) *model* dalam mengidentifikasi masing-masing kelas secara proporsional. Pada grafik ini:

a. *Recall* untuk kelas "*Fire*" = 0.99, artinya 99% data yang sebenarnya merupakan api berhasil dikenali oleh *model* sebagai api.
b. *Recall* untuk kelas "*Smoke*" = 0.99, yang berarti 99% data asap juga berhasil diklasifikasikan dengan benar.

Dengan nilai *Recall* yang sangat tinggi dan kesalahan klasifikasi (mis*Classification*) hanya sebesar 1% untuk masing-masing kelas, ini menunjukkan bahwa *model* tidak hanya akurat secara keseluruhan, tetapi juga sangat andal dalam

membedakan objek antara api dan asap secara konsisten, bahkan pada berbagai kondisi pencahayaan atau latar.



Gambar 4. 3 Confusion Matrix Normalize

Confusion Matrix tanpa normalisasi menyajikan jumlah klasifikasi *model* dalam angka absolut berdasarkan hasil prediksi terhadap data uji. Matriks berikut menunjukkan:

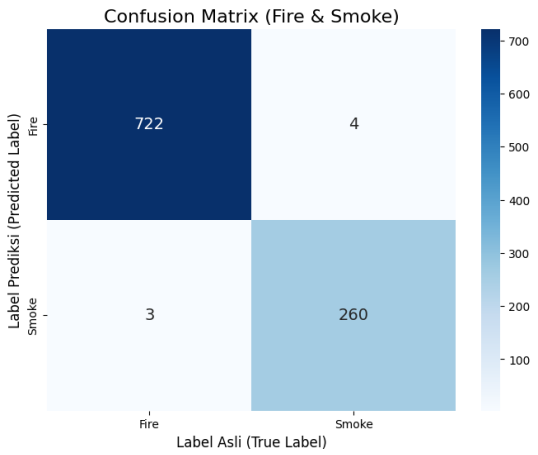
Tabel 4. 1 Confusion Matrix

	True *Fire*	**True *Smoke***
Predicted Fire	722 (TP *Fire*)	4 (FP *Smoke*)
Predicted Smoke	3 (FN *Fire*)	260 (TP *Smoke*)

Interpretasi:

a. *Model* berhasil mendeteksi 722 dari 725 gambar bertanda "*Fire*", dan hanya salah memprediksi 3 gambar sebagai "*Smoke*".
b. Untuk kelas "*Smoke*", 260 dari 264 gambar dikenali dengan tepat, dengan hanya 4 kesalahan sebagai "*Fire*".
c. Total akurasi:

$$\frac{722 + 260}{722 + 260 + 3 + 4} = \frac{982}{989} \approx 99.29\% \quad (7)$$



Gambar 4. 4 Confusion matrix

4.4.2 *Classification Report*

Classification Report digunakan untuk memberikan ringkasan evaluasi performa *model* deteksi berdasarkan tiga metrik utama: *Precision*, *Recall*, dan *F1-Score*, yang dihitung untuk setiap kelas (dalam hal ini: *Fire* dan *Smoke*). Ketiga metrik ini penting untuk menilai efektivitas *model* YOLOv11 dalam membedakan dua objek yang secara visual memiliki kemiripan, terutama dalam kondisi nyata seperti pencahayaan redup atau latar belakang kompleks.

1. Kelas: *Fire*

a. *Precision* sebesar 99,59% menunjukkan bahwa dari semua prediksi terhadap objek "*Fire*", sebanyak 99,59% benar-benar berisi api.

b. *Recall* sebesar 99,45% mengindikasikan bahwa *model* mampu mengenali hampir seluruh gambar yang seharusnya mengandung api.

c. *F1-Score* mencapai 99,52%, mencerminkan bahwa *model* memiliki keseimbangan yang sangat baik antara ketepatan dan sensitivitas deteksi api.

2. Kelas: *Smoke*

a. *Precision* untuk kelas "*Smoke*" tercatat sebesar 98,48%, yang berarti sebagian kecil dari prediksi "*Smoke*" merupakan kesalahan klasifikasi.

b. *Recall* sebesar 98,86% menunjukkan bahwa hampir seluruh gambar yang mengandung asap berhasil dikenali dengan benar oleh *model*.

c. *F1-Score* sebesar 98,67% membuktikan bahwa performa *model* terhadap objek asap juga sangat baik, meskipun sedikit lebih rendah dari kelas api.

Tabel 4. 2 Rangkuman Evaluasi

Kelas	***Precision***	***Recall***
Fire	99.59%	99.45%
Smoke	98.48%	98.86%

```
============================================
      LAPORAN PERFORMA PER KELAS (TERMASUK F1)
============================================

--- Analisis Kelas: 'Fire' ---
Total Sampel Asli          : 726.0 gambar
  > Benar Terdeteksi       : 722.0 (True Positive)
  > Gagal Terdeteksi       : 4.0 (False Negative)
    - 4.0 gambar 'Fire' salah dideteksi sebagai 'Smoke'

> Akurasi Deteksi (Recall) : 99.45%
> Presisi Prediksi (Precision): 99.59%
> F1-Score                 : 99.52%

--- Analisis Kelas: 'Smoke' ---
Total Sampel Asli          : 263.0 gambar
  > Benar Terdeteksi       : 260.0 (True Positive)
  > Gagal Terdeteksi       : 3.0 (False Negative)
    - 3.0 gambar 'Smoke' salah dideteksi sebagai 'Fire'

> Akurasi Deteksi (Recall) : 98.86%
> Presisi Prediksi (Precision): 98.48%
> F1-Score                 : 98.67%

============================================
Analisis selesai.
```

Gambar 4. 5 Classification Report

Secara keseluruhan, *Classification Report* ini memperlihatkan bahwa *model* YOLOv11 memiliki performa sangat tinggi dan seimbang untuk kedua kelas, dengan *F1-Score* rata-rata melebihi 98,5%. Nilai *precision* dan *recall* yang tinggi menunjukkan bahwa *model* minim menghasilkan *False Positive* maupun *False Negative*, yang sangat penting dalam sistem peringatan dini kebakaran.

F1-Score yang tinggi pada kedua kelas juga menjadi indikator bahwa *model* tidak hanya akurat secara prediksi, namun juga konsisten dalam mendeteksi pola-pola visual yang beragam dari api dan asap.

4.4.3 Pembahasan Hasil Parameter Pelatihan

Parameter pelatihan seperti *loss*, *precision*, *recall*, dan *mean average precision* (*mAP*) merupakan indikator utama dalam menilai performa *model* YOLOv11. Selama proses *training*, *model* menunjukkan penurunan *loss* yang konsisten pada data pelatihan maupun validasi, yang menandakan bahwa proses konvergensi berjalan dengan baik dan stabil.

Nilai *precision* mencapai kestabilan pada kisaran 0,90, sedangkan *recall* berada pada rentang 0,85–0,87, mengindikasikan bahwa *model* mampu melakukan klasifikasi dengan akurasi tinggi dan kesalahan minimal. Pada metrik *mAP*, nilai *mAP@0.5* tercatat mendekati 0,88, sementara *mAP@0.5:0.95* berada di sekitar 0,69–0,70. Nilai tersebut menunjukkan bahwa *model* memiliki kemampuan generalisasi yang kuat terhadap berbagai ukuran dan jenis objek dalam citra.

Hasil klasifikasi per kelas menunjukkan performa yang sangat baik. Untuk kelas "*Fire*", *model* menghasilkan *F1-score* sebesar 0,9952, dengan *precision* sebesar 99,59% dan *recall* sebesar 99,45%. Angka ini merefleksikan keseimbangan yang sangat baik antara ketepatan dan sensitivitas *model* terhadap objek api.

Sementara itu, pada kelas "Smoke", diperoleh *F1-score* sebesar 0,9867, dengan *precision* sebesar 98,48% dan *recall* sebesar 98,86%. Performa ini sedikit lebih rendah dibandingkan kelas *Fire*, karena karakteristik visual asap yang cenderung lebih samar, transparan, dan sering bercampur dengan latar belakang.

Secara umum, kedua kelas utama menunjukkan hasil klasifikasi yang sangat tinggi dan stabil. Namun, perlu dicatat bahwa pada eksperimen sebelumnya yang melibatkan kelas *background*, *model* masih menghasilkan *false positive* yang cukup tinggi, di mana objek latar sering salah dikenali sebagai api atau asap. Masalah ini dapat diatasi dengan menambahkan lebih banyak data latar (*background augmentation*) pada

proses pelatihan dan melakukan penyesuaian terhadap ambang deteksi (*Confidence threshold*).

Dengan demikian, performa parameter pelatihan dan hasil klasifikasi memperkuat kesimpulan bahwa *model* YOLOv11 mampu mendeteksi objek api dan asap dengan akurasi tinggi dan reliabilitas yang baik untuk implementasi sistem peringatan dini kebakaran berbasis website.

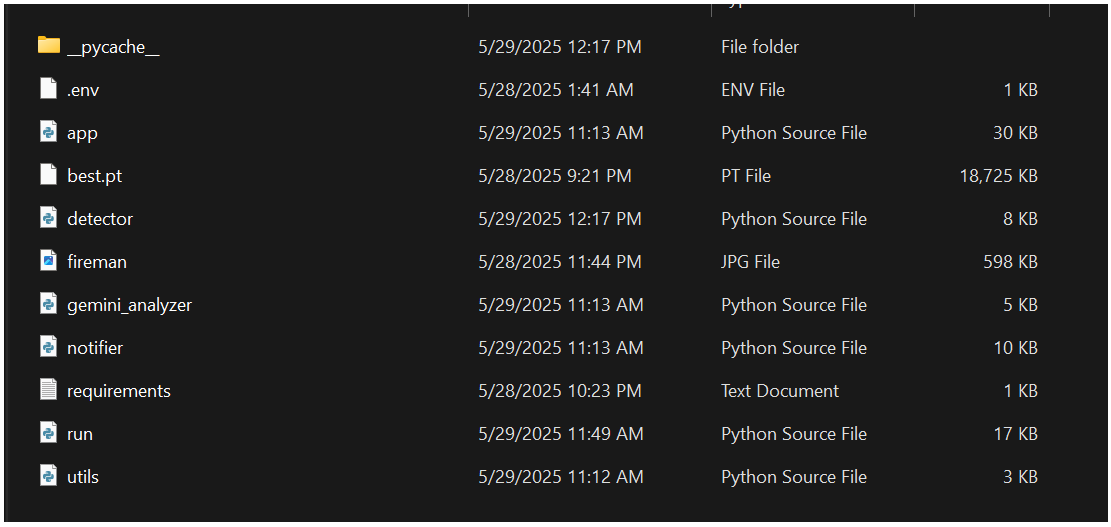
4.5 *Deployment*

Setelah proses pelatihan dan evaluasi selesai, *model* terbaik disimpan dalam format .pt dan diintegrasikan ke dalam aplikasi web berbasis *Streamlit*. Aplikasi ini dirancang untuk mendeteksi api dan asap secara interaktif, baik melalui unggahan gambar, pemutaran video, maupun streaming kamera secara *Real-Time*. *Model YOLOv11* akan secara otomatis menganalisis setiap *frame* dan menampilkan deteksi dalam bentuk *Bounding box* beserta nilai *Confidence* .

Sebagai bagian dari sistem peringatan dini, jika terdeteksi keberadaan api atau asap, sistem akan mengirimkan notifikasi instan ke pengguna melalui saluran *Telegram* yang telah dikonfigurasi. Selain itu, sistem juga terhubung dengan Gemini AI, yang berperan dalam memberikan analisis deskriptif tambahan terhadap hasil deteksi, seperti menjelaskan kemungkinan sumber api atau konteks visual secara umum. Antarmuka pengguna dibuat ringan dan responsif, memungkinkan akses dari berbagai perangkat tanpa memerlukan spesifikasi perangkat keras tinggi.

4.5.1 Struktur *directory* proyek

Sistem terstruktur dalam direktori yang terdiri dari file *model* AI 'best.pt', aplikasi utama *'app*.py', modul deteksi *'detector*.py', utilitas *'utils*.py', konfigurasi *'requirements*.txt' dan '*.env'*, serta modul notifikasi *'notifier*.py'. Struktur ini mendukung integrasi antara *model* deteksi, antarmuka pengguna, dan sistem notifikasi.

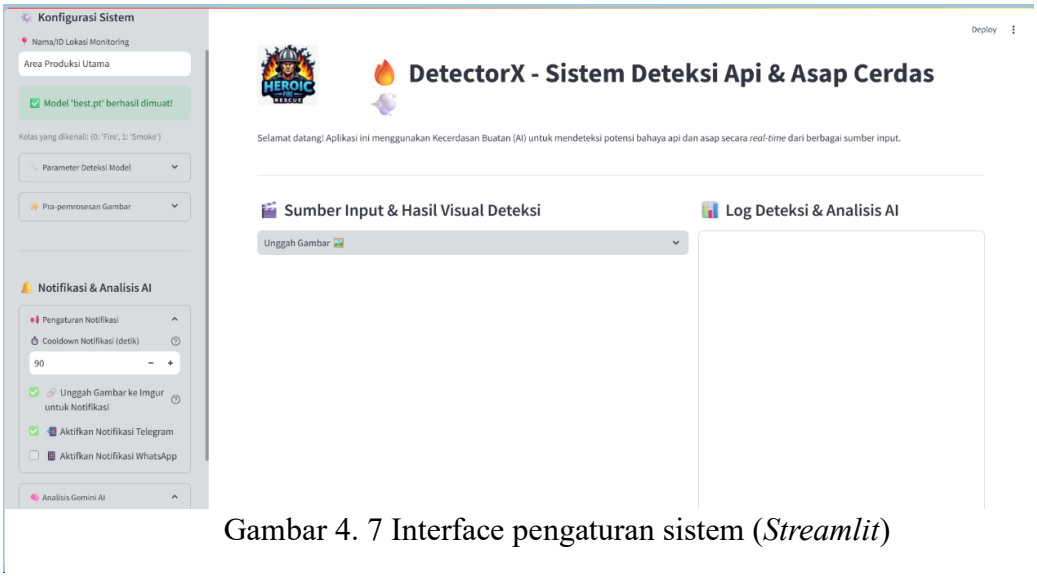


__pycache__	5/29/2025 12:17 PM	File folder	
.env	5/28/2025 1:41 AM	ENV File	1 KB
app	5/29/2025 11:13 AM	Python Source File	30 KB
best.pt	5/28/2025 9:21 PM	PT File	18,725 KB
detector	5/29/2025 12:17 PM	Python Source File	8 KB
fireman	5/28/2025 11:44 PM	JPG File	598 KB
gemini_analyzer	5/29/2025 11:13 AM	Python Source File	5 KB
notifier	5/29/2025 11:13 AM	Python Source File	10 KB
requirements	5/28/2025 10:23 PM	Text Document	1 KB
run	5/29/2025 11:49 AM	Python Source File	17 KB
utils	5/29/2025 11:12 AM	Python Source File	3 KB

Gambar 4. 6 Struktur directory proyek

4.5.2 *Interface* Pengaturan Sistem

Interface Streamlit menampilkan panel pengaturan yang mencakup lokasi monitoring 'Area Produksi Utama', status pe*model*an, parameter deteksi, dan pra-

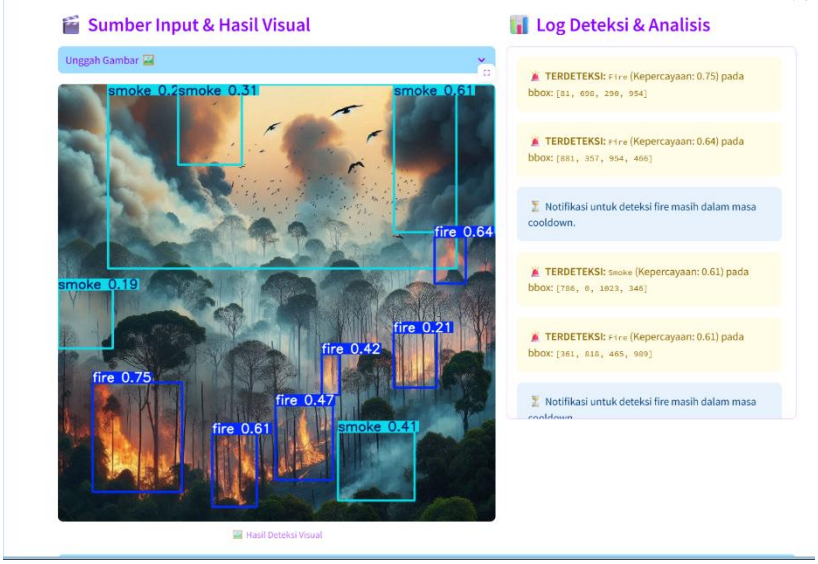


Gambar 4. 7 Interface pengaturan sistem (*Streamlit*)

pemrosesan gambar. Panel notifikasi menyediakan konfigurasi cooldown, integrasi *Imgur*, serta aktifasi notifikasi ke *Telegram* dan WhatsApp, dengan integrasi analisis Gemini AI untuk pemrosesan lanjutan.

4.5.3 *Interface* Deteksi Api & Asap

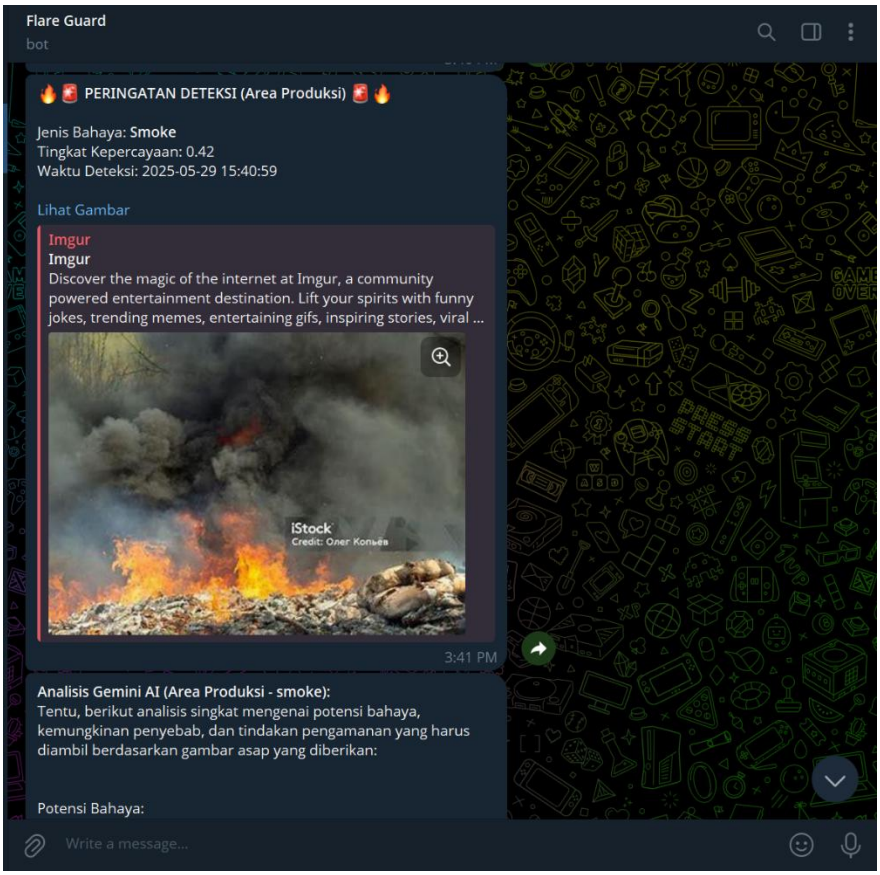
Interface deteksi menampilkan sumber input dengan fitur drag-and-drop, hasil deteksi visual dengan *Bounding box* dan skor kepercayaan, serta log deteksi yang mencakup analisis Gemini AI tentang potensi bahaya dan kemungkinan penyebab kebakaran, disertai rekomendasi mitigasi berdasarkan konteks lokasi.



Gambar 4. 8 Tampilan *Streamlit* setelah mendeteksi

4.5.4 Notifikasi Hasil Deteksi di *Telegram*

Bot *Telegram 'Flare Guard'* mengirim peringatan deteksi dengan detail lokasi, jenis bahaya, tingkat kepercayaan, dan waktu deteksi, serta link gambar deteksi via *Imgur*. Notifikasi juga mencakup analisis Gemini AI tentang potensi bahaya, kemungkinan penyebab, dan tindakan mitigasi yang disarankan, memberikan gambaran komprehensif tentang insiden deteksi.



Gambar 4. 9 *Interface* notifikasi di Telegram

4.5.5 *Confidence score* Deteksi

Dalam deteksi objek menggunakan YOLOv11, setiap objek yang terdeteksi diberikan nilai *Confidence score*, yaitu nilai antara 0 sampai 1 yang merepresentasikan tingkat keyakinan *model* bahwa objek tersebut benar-benar hadir dalam *Bounding box* tersebut. Nilai ini dihitung berdasarkan kombinasi dari probabilitas kelas dan skor overlap (IoU) antara prediksi dan *ground truth*.

Semakin tinggi nilai *Confidence*, semakin besar kemungkinan objek yang terdeteksi benar-benar sesuai dengan kelas yang dimaksud. Dalam konteks deteksi api dan asap, ini sangat penting karena bentuk, warna, dan karakteristik visual dari objek tersebut bisa sangat bervariasi.

Tabel 4. 3 Rentang *Confidence score*

Confidence score	**Tingkat Keyakinan**	**Interpretasi Model**	**Tindakan Sistem (Rekomendasi)**
≥ 0.70 – 0.89	Tinggi	Model cukup yakin objek api atau asap benar-benar ada. Cocok untuk notifikasi dan aksi sistem otomatis.	Tampilkan secara penuh dan kirim notifikasi (jika sistem diaktifkan).
0.50 – 0.69	Sedang	Model mendeteksi objek, namun masih ada potensi error. Umumnya valid, tapi perlu verifikasi tambahan.	Tampilkan hasil deteksi dengan label "Perlu Validasi". Cocok untuk aplikasi semi-otomatis.
0.30 – 0.49	Rendah	Model ragu, namun objek bisa terlihat jelas secara visual (terutama untuk asap atau api kecil).	Tampilkan opsional dengan label "Kemungkinan Objek". Aktifkan hanya jika mode debug/manual review.
< 0.30	Sangat Rendah	*Confidence* sangat lemah. Kemungkinan false positive	Abaikan secara *default*, tapi bisa ditampilkan dalam mode evaluasi atau

Confidence score	Tingkat Keyakinan	Interpretasi Model	Tindakan Sistem (Rekomendasi)
		tinggi. Namun jika objek jelas secara visual, bisa dipertimbangkan untuk validasi manual.	dengan label: Validasi Visual Diperlukan. Tidak dikirim notifikasi otomatis.

Tabel rentang *Confidence score* ini digunakan untuk memahami seberapa yakin model dalam mendeteksi api atau asap. Jika nilainya tinggi (≥ 0.70), sistem akan langsung menampilkan hasil dan bisa mengirim notifikasi karena deteksi dianggap akurat. Pada tingkat sedang (0.50–0.69), hasil tetap ditampilkan, tapi diberi tanda "perlu validasi" agar pengguna bisa mengecek ulang. Jika nilainya rendah (0.30–0.49), deteksi hanya ditampilkan dalam mode debug atau diberi label "kemungkinan objek", karena model masih ragu. Untuk nilai sangat rendah (<0.30), sistem akan menyembunyikan hasil secara *default*, kecuali jika objek api atau asap memang sangat terlihat jelas dalam kasus seperti itu, pengguna bisa diberi opsi untuk melakukan validasi manual. Tujuannya adalah menyesuaikan kepercayaan sistem dengan logika manusia, agar deteksi tetap akurat namun tetap fleksibel dalam kondisi nyata.

4.6 Konversi Visual ke Representasi Numerik

Dalam sistem deteksi objek berbasis YOLOv11, data visual dalam bentuk gambar tidak dapat diproses secara langsung oleh *model*. Oleh karena itu, diperlukan transformasi citra ke bentuk numerik melalui proses *anotasi* dan *encoding*. Proses ini dilakukan secara otomatis oleh Roboflow saat menyiapkan *Dataset*, dengan menghasilkan file *anotasi* dalam format .txt yang menyimpan informasi numerik untuk setiap objek terdeteksi. Setiap baris dalam file label mengikuti format *[class_id, x_center, y_center, width, height]*, dengan seluruh nilai telah dinormalisasi terhadap

dimensi gambar (misalnya, rentang 0 hingga 1). Ini memungkinkan YOLO mengenali posisi relatif objek tanpa terpengaruh oleh resolusi citra.

Kode menunjukkan bahwa *pipeline training* YOLOv11 menggunakan file data.yaml sebagai konfigurasi utama. File ini menunjuk ke direktori gambar dan label (*train, valid, test)* yang telah disiapkan dalam struktur yang dikenali oleh pustaka Ultralytics. Proses preprocessing tambahan seperti *resizing* citra ke ukuran 640x640 piksel, augmentasi (*flip horizontal*, rotasi, *brightness*), serta normalisasi warna dilakukan secara otomatis sebelum gambar masuk ke jaringan saraf konvolusional.
Secara teknis, setiap gambar yang masuk ke dalam YOLOv11 akan melalui tahapan:

1. *Gridization*: citra dibagi menjadi grid S x S.
2. *Bounding box Regression*: setiap grid memprediksi koordinat box: (*x_center, y_center, width, height*).
3. *Class Probability* & *Confidence score*: *model* menghitung probabilitas kelas dan skor keyakinan menggunakan rumus gabungan dari Pr(Class) × IOU.

Transformasi ini menjadikan data visual menjadi vektor numerik berdimensi tetap, yang kemudian diekstraksi fiturnya menggunakan lapisan konvolusi dan diproses oleh YOLO *head* untuk klasifikasi dan regresi *Bounding box*. Proses ini sangat penting untuk memungkinkan *inference Real-Time* yang efisien dan akurat.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil merancang dan mengimplementasikan sistem deteksi api dan asap secara *Real-Time* menggunakan algoritma *You Only Look Once* versi 11 (*YOLOv11*) yang diintegrasikan ke dalam platform *Streamlit*. Sistem dilatih menggunakan *dataset* Smoke and Fire v2 dari Roboflow dengan dukungan teknik augmentasi seperti *mosaic*, rotasi, dan penyesuaian kecerahan (*brightness adjustment*), yang membantu meningkatkan keberagaman data dan kemampuan generalisasi *model*.

Hasil evaluasi menunjukkan performa deteksi yang sangat baik, dengan nilai *mean average precision* (*mAP@0.5*) sebesar 0,88, *precision* sebesar 0,89, dan *recall* sebesar 0,87 pada pengujian global. Pada evaluasi klasifikasi per kelas, diperoleh *F1-score* sebesar 99,52% untuk kelas *Fire* dan 98,67% untuk kelas *Smoke*, menunjukkan bahwa *model* mampu mendeteksi dua objek tersebut secara akurat dan seimbang. Sistem ini juga dilengkapi antarmuka (*user interface*) berbasis web yang interaktif serta notifikasi otomatis melalui *Telegram Bot*, sehingga memungkinkan deteksi dini kebakaran dilakukan secara efisien tanpa memerlukan perangkat keras tambahan.

Pengujian performa juga mencatat waktu respons rata-rata sebesar 0,35 detik per frame, yang cukup cepat dan mendukung kebutuhan sistem deteksi berbasis *web application* secara *Real-Time*. Meskipun demikian, ditemukan beberapa tantangan dalam mendeteksi asap yang sangat samar atau kondisi citra dengan latar belakang kompleks, yang dapat memicu *false positive*. Hal ini menunjukkan bahwa sistem masih memiliki ruang pengembangan, terutama dalam meningkatkan daya tahan terhadap variasi kondisi lingkungan ekstrem serta memperkaya data latar (*background*) dalam proses pelatihan.

Secara keseluruhan, sistem deteksi berbasis YOLOv11 yang dikembangkan dalam penelitian ini dinilai layak untuk diimplementasikan sebagai solusi deteksi kebakaran berbasis web yang cepat, ringan, dan akurat.

5.2 Saran

Berdasarkan hasil penelitian dan implementasi sistem deteksi api dan asap menggunakan *YOLOv11*, terdapat beberapa saran yang dapat dijadikan pertimbangan untuk pengembangan sistem lebih lanjut:

1. **Penambahan dan Diversifikasi *Dataset***

   Disarankan untuk memperluas variasi data latar belakang non-kebakaran, termasuk kondisi ekstrem seperti kabut, pencahayaan rendah, pantulan cahaya, serta lingkungan industri atau perkotaan. Diversifikasi ini bertujuan untuk meningkatkan kemampuan generalisasi *model* dan mengurangi risiko *false positive*, terutama dalam mendeteksi asap samar atau objek visual yang menyerupai api.

2. **Peningkatan Kualitas *Labeling* dan Validasi Data**

   Proses pelabelan (*labeling*) pada *dataset* perlu ditingkatkan akurasinya melalui validasi ganda (*double-checking*), untuk menghindari ambiguitas antara objek *fire*, *smoke*, dan *background*. Konsistensi dan ketepatan label sangat penting agar *model* dapat belajar membedakan objek dengan lebih baik dan meminimalkan kesalahan klasifikasi.

3. **Integrasi dengan Perangkat *IoT* dan Kamera *CCTV***

   Untuk meningkatkan nilai guna dalam implementasi nyata, sistem dapat dikembangkan agar terintegrasi langsung dengan perangkat *Internet of Things (IoT)* seperti kamera *CCTV*, *microcontroller*, atau *drone*. Hal ini memungkinkan pemantauan wilayah rawan kebakaran secara otomatis, luas, dan *Real-Time*, serta mempercepat proses deteksi dan respons.

4. **Optimasi *Model* untuk *Edge Deployment***

   Mengingat keterbatasan daya komputasi di beberapa lingkungan operasional, disarankan untuk mengoptimasi *model* agar dapat dijalankan secara efisien di perangkat *edge computing* seperti *Raspberry Pi* atau *Orange Pi*. Dengan melakukan *pruning*, *quantization*, atau menggunakan arsitektur ringan, sistem tetap dapat memberikan deteksi cepat dengan konsumsi sumber daya minimal.

5. **Pengembangan Modul Prediksi dan Analisis Historis**

   Untuk meningkatkan kemampuan sistem dalam mendukung mitigasi kebakaran, dapat ditambahkan modul prediksi berbasis *historical data*, seperti analisis waktu kejadian, lokasi, dan intensitas deteksi. Penggunaan *heatmap* dan grafik analitik juga dapat membantu pengguna dalam mengenali pola kebakaran dan merancang strategi penanggulangan secara preventif.

DAFTAR PUSTAKA

Aboyomi, D. D., & Daniel, C. (2023). A Comparative Analysis of Modern Object Detection Algorithms: YOLO vs. SSD vs. Faster R-CNN. *ITEJ (Information Technology Engineering Journals)*, *8*(2), 96–106. https://doi.org/10.24235/itej.v8i2.123

Achmad Rizal, Ramadah, F., & Wibawa, IG. P. D. (2022). *Sistem Deteksi Api Menggunakan Pengolahan Citra Pada Webcam Dengan Metode Yolov3 Fire Detection System Using Image Processing on Webcam with Yolov3 Method*.

Alif, M. A. R. (2024). *YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems*. http://arxiv.org/abs/2410.22898

Andarsyah, R., & Yanuar, A. (2024). SENTIMEN ANALISIS APLIKASI POSAJA PADA GOOGLE PLAYSTORE UNTUK PENINGKATAN POSPAY SUPERAPP MENGGUNAKAN SUPPORT VECTOR MEACHINE. In *Jurnal Teknik Informatika* (Vol. 16, Issue 2).

Apostolidis, K. D., & Papakostas, G. A. (2025). Delving into YOLO Object Detection Models: Insights into Adversarial Robustness. *Electronics*, *14*(8), 1624. https://doi.org/10.3390/electronics14081624

Arnesia, P. D., Pratama, N. A., & Sjafrina, F. (2022). *APLIKASI ARTIFICIAL INTELLIGENCE UNTUK MENDETEKSI OBJEK BERBASIS WEB MENGGUNAKAN LIBRARY TENSORFLOW JS, REACT JS DAN COCO DATASET*.

Asshiddiqie, M. A. J., Rahmat, B., & Anggraeny, F. T. (2020). DETEKSI TANAMAN TEBU PADA LAHAN PERTANIAN MENGGUNAKAN

METODE CONVOLUTIONAL NEURAL NETWORK. In *Jurnal Informatika dan Sistem Informasi (JIFoSI)* (Vol. 1, Issue 1).

Azhar, K. M., Santoso, I., & Soetrisno, Y. A. A. (2021). IMPLEMENTASI DEEP LEARNING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DAN ALGORITMA YOLO DALAM SISTEM PENDETEKSI UANG KERTAS RUPIAH BAGI PENYANDANG LOW VISION. *Transient: Jurnal Ilmiah Teknik Elektro*, *10*(3), 502–509. https://doi.org/10.14710/transient.v10i3.502-509

Azzahra, L., Kusumawardhani, Dara Kusumawati Ramadani Yasir, I. B. I. K., & Suryo Adhi Wibowo, Rissa Rahmania, A. R. H. (2024). ANALISIS KINERJA DETEKSI GERAKAN DAN PENGENALAN OBJEK PRODUK RITEL BERBASIS YOLOV8. *Jurnal Elektro Dan Telekomunikasi Terapan*, *11*(1), 43–53. https://doi.org/10.25124/jett.v11i1.7482

Bindal, R., Deokar, S., Rathore, T. S., Sharma, A., Gangarde, R., & Jatti, A. (2024). AI-ML Innovations in Forest Fire Detection: Protecting Ecosystems and Communities. *2024 International Conference on Computing, Sciences and Communications (ICCSC)*, 1–6. https://doi.org/10.1109/ICCSC62048.2024.10830309

Dompeipen, T. A., & Sompie, S. R. U. A. (2020). Penerapan computer vision untuk pendeteksian dan penghitung jumlah manusia. *Jurnal Teknik Informatika*, *15*(4), 1–12.

Ferian, M., Akbari, R., Rahayudi, B., & Muflikhah, L. (2023). *Implementasi Deep Learning menggunakan Algoritma EfficientDet untuk Sistem Deteksi Kelayakan Penerima Bantuan Langsung Tunai berdasarkan Citra Rumah di Wilayah Kabupaten Kediri* (Vol. 7, Issue 4). http://j-ptiik.ub.ac.id

Gallu, A., Himamunanto, A. R., & Budiati, H. (2024). *Pengenalan Emosi pada Citra wajah menggunakan Metode YOLO* (Vol. 5, Issue 3).

Gede, I., Widharma, S., Alit, P., Santiary, W., Nengah Sunaya, I., Ketut Darminta, I., Gde, I., Sangka, N., Ardy, D. P., & Widiatmika, W. (2022). Deteksi api kebakaran berbasis computer vision dengan algoritma YOLO. *Journal of Applied Mechanical Engineering and Green Technology*, *3*, 53–58. https://ojs2.pnb.ac.id/index.php/JAMETECH

Gunawan, G. (2021). DATA MINING USING CRISP-DM PROCESS FRAMEWORK ON OFFICIAL STATISTICS: A CASE STUDY OF EAST JAVA PROVINCE. *Jurnal Ekonomi Dan Pembangunan*, *29*(2), 183–198. https://doi.org/10.14203/JEP.29.2.2021.183-198

Hendrianty, B. J., Ibrahim, A., Iskandar, S., & Mulyasari, E. (2024). *Kalam Cendekia: Jurnal Ilmiah Kependidikan Membangun Pola Pikir Deep Learning Guru Sekolah Dasar*.

Hidayat, F. T., & Whardana, A. K. (2024). *Deteksi Pelanggaran Sepeda Motor Menggunakan Algoritma Yolo Dan Mean Average Precision*.

Hindarto, D. (2023). Exploring YOLOv8 Pretrain for *Real-Time* Detection of Indonesian Native Fish Species. *Sinkron*, *8*(4), 2776–2785. https://doi.org/10.33395/sinkron.v8i4.13100

Jauhari, N. M. I., Wulanningrum, R., & Setiawan, A. B. (2024). Sistem Deteksi Kendaraan Menggunakan *Streamlit* Metode Yolo Universitas Nusantara PGRI Kediri. In *Seminar Nasional Inovasi Teknologi) 1331 INOTEK* (Vol. 8). Online.

Jegham, N., Koh, C. Y., Abdelatti, M., & Hendawi, A. (2024). *YOLO Evolution: A Comprehensive Benchmark and Architectural Review of YOLOv12, YOLO11, and Their Previous Versions*. http://arxiv.org/abs/2411.00201

Juliansyah, R., Ar Rachman, M. A. M., Amin, M. Al, Tyanafisya, A., Hanifah, N. A., Giri, E. P., & Mindara, G. P. (2024a). Development of a *Real-Time* Traffic Density Detection Website Using YOLOv8-Based Digital Image Processing with OpenCV. *Journal of Information Systems and Informatics*, *6*(4), 2649–2678. https://doi.org/10.51519/journalisi.v6i4.912

Juliansyah, R., Ar Rachman, M. A. M., Amin, M. Al, Tyanafisya, A., Hanifah, N. A., Giri, E. P., & Mindara, G. P. (2024b). Development of a *Real-Time* Traffic Density Detection Website Using YOLOv8-Based Digital Image Processing with OpenCV. *Journal of Information Systems and Informatics*, *6*(4), 2649–2678. https://doi.org/10.51519/journalisi.v6i4.912

Khanam, R., & Hussain, M. (2024a). *YOLOv11: An Overview of the Key Architectural Enhancements*. http://arxiv.org/abs/2410.17725

Khanam, R., & Hussain, M. (2024b). *YOLOv11: An Overview of the Key Architectural Enhancements*. http://arxiv.org/abs/2410.17725

Lin, Q., Ye, G., Wang, J., & Liu, H. (2021). *RoboFlow: a Data-centric Workflow Management System for Developing AI-enhanced Robots*. https://sites.google.com/u.northwestern.edu/roboflow

Luo, Y. (2025). *The Evolution of YOLO: From YOLOv1 to YOLOv11 with a Focus on YOLOv7's Innovations in Object Detection*. https://doi.org/10.54254/2753-8818/87/2025.20335

Narayana, G. V. S., Kuanar, S. K., & Patel, P. (2024). *Weed Detection in Cotton Production Systems Using Novel YOLOv7-X Object Detector* (pp. 303–314). https://doi.org/10.1007/978-981-99-3932-9_27

Nurdiyansyah, F., Akbar, I., & Ursaputra, L. (2025). Segmentasi Berbasis Warna Untuk Pengelompokan Kualitas Cacing Anc Menggunakan

Yolov8. *JIKO (Jurnal Informatika Dan Komputer)*, *9*(1), 239. https://doi.org/10.26798/jiko.v9i1.1779

Permana, A. A., Muttaqin, R., & Sunandar, A. (2024). SISTEM DETEKSI API SECARA REAL TIME MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO) VERSI 8. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Issue 5). https://universe.roboflow.com/.

Pinem, J., Lubis, A. A., Denia, Y., & Syaputra, M. A. (2023). Implementasi Algoritma YOLO Dalam Pengklasifikasian Objek Transportasi pada Lalu Lintas Kota Medan. *Populer: Jurnal Penelitian Mahasiswa*, *3*(1), 13–23. https://doi.org/10.58192/populer.v3i1.1641

Pradana, A. I., Harsanto, H., & Wijiyanto, W. (2024). Deteksi Rambu Lalu Lintas *Real-Time* di Indonesia dengan Penerapan YOLOv11: Solusi Untuk Keamanan Berkendara. *Jurnal Algoritma*, *21*(2), 145–155. https://doi.org/10.33364/algoritma/v.21-2.2106

Pratama, E. C., & Hidayati, N. (2024). Pemanfaatan Deep Learning Dalam Pembuatan Sistem Kecerdasan Buatan Pendeteksi Kantuk Menggunakan *Streamlit*. *Jurnal Pengembangan Rekayasa Dan Teknologi*, *8*(2), 43–50. https://journals.usm.ac.id/index.php/jprt■page43

Prokopyev, M. S., Vlasova, E. Z., Tretyakova, T. V., Sorochinsky, M. A., & Solovyeva, R. A. (2020). Development of a Programming Course for Students of a Teacher Training Higher Education Institution Using the Programming Language Python. *Propósitos y Representaciones*, *8*(3). https://doi.org/10.20511/pyr2020.v8n3.484

Raya Ismail, D., Rahmadewi Teknik Elektro, R., Singaperbangsa Karawang HSRonggo Waluyo, U., Telukjambe Tim, K., Karawang, K., & Barat, J. (2025). SISTEM DETEKSI JALAN BERLUBANG SECARA *REAL-TIME* MENGGUNAKAN YOLOV11: INTEGRASI DATA DAN

LOKASI MELALUI WEBSITE (STUDI KASUS : DAERAH KARAWANG). In *Jurnal Mahasiswa Teknik Informatika)* (Vol. 9, Issue 3).

Ruswanti, D., Susilo, D., & Riani, R. (2024). Implementasi CRISP-DM pada Data Mining untuk Melakukan Prediksi Pendapatan dengan Algoritma C.45. *Go Infotech: Jurnal Ilmiah STMIK AUB*, *30*(1), 111–121. https://doi.org/10.36309/goi.v30i1.266

Saputri, R. S., Apriliani, A., & Mukminin, A. (2025). Detecting the Number of Students Using YOLOv11 to Prevent Proxy Attendance at Universitas Dinamika Bangsa. *Media Journal of General Computer Science*, *2*(1), 48–56. https://doi.org/10.62205/mjgcs.v2i1.38

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, *2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

Setiawan, R. A., & Setyanto, A. (2024). Evaluasi Trade-off Akurasi dan Kecepatan YOLOv5 dalam Deteksi Kebakaran pada Edge Devices. In *Syntax Admiration* (Vol. 5, Issue 11).

Sobron M, Y, L. B., & T, M. (2021). *IMPLEMENTASI ARTIFICIAL INTELLIGENCE PADA SYSTEM MANUFAKTUR TERPADU*.

Surbakti, N. M., Angelyca Angelyca, Anita Talia, Cecilia Br Perangin-Angin, Dina Olivia Nainggolan, Nia Devi Friskauly, & Sikap Ruth Br Tumorang. (2024). Penggunaan Bahasa Pemrograman Python dalam Pembelajaran Kalkulus Fungsi Dua Variabel. *Algoritma : Jurnal Matematika, Ilmu Pengetahuan Alam, Kebumian Dan Angkasa*, *2*(3), 98–107. https://doi.org/10.62383/algoritma.v2i3.67

Tungady, C. A. P., & Purnomo, H. D. (2023). Perancangan Sistem Pendukung Objek Deteksi untuk Permainan Kartu Cardfight!Vanguard Menggunakan Aplikasi Roboflow dan Flask. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, *6*(3), 283–290. https://doi.org/10.32493/jtsi.v6i3.30303

Vetrita, Y., Albar, I., Santoso, I., Prasasti, I., Kartika, T., Usman, A. B., Tosiani, A., Haryanto, D., Endrawati, Famurianty, E., Ulfa, K., & Purwanto, J. (2024). Monthly mapping of Indonesia's burned areas: implementation, history, techniques, and future directions. *International Journal of Remote Sensing*. https://doi.org/10.1080/01431161.2024.2421942

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for Real-Time object detectors*. http://arxiv.org/abs/2207.02696

Wang, C.-Y., & Liao, H.-Y. M. (2024). *YOLOv1 to YOLOv10: The fastest and most accurate Real-Time object detection systems*. https://doi.org/10.1561/116.20240058

Wardhana, R. G., Wang, G., & Sibuea, F. (2023). PENERAPAN MACHINE LEARNING DALAM PREDIKSI TINGKAT KASUS PENYAKIT DI INDONESIA. In *Journal of Information System Management (JOISM) e-ISSN* (Vol. 5, Issue 1).

Wijoyo, A., Saputra, A. Y., Ristanti, S., Sya'ban, R., Amalia, M., & Febriansyah, R. (2024). *Pembelajaran Machine Learning*.

Yanto, Faruq Aziz, & Irmawati. (2023). *YOLO-V8 PENINGKATAN ALGORITMA UNTUK DETEKSI PEMAKAIAN MASKER WAJAH*.