

SISTEM DETEKSI KEBAKARAN BERBASIS VISI KOMPUTER DENGAN YOLOv11 TERINTEGRASI DENGAN TELEGRAM

Nico Demeus Hasoloan Manurung¹, Rajes Khana², Jemie Muliadi³ dan Muhammad Sobirin³

¹Fakultas Teknik dan Informatika, Universitas 17 Agustus 1945 Jakarta, Indonesia

²Program Studi Informatika, Universitas 17 Agustus 1945, Jakarta, Indonesia

³Program Studi Teknik Elektro, Universitas 17 Agustus 1945, Jakarta, Indonesia

*Penulis korespondensi, E-mail: manurungnico681@gmail.com, rajes.khana@uta45jakarta.ac.id,
jemie.muliadi@uta45jakarta.ac.id, muhammad.sobirin@uta45jakarta.ac.id

Abstrak

Deteksi dini api menjadi penting dalam upaya mitigasi risiko kebakaran. Sistem deteksi tersebut dikembangkan dalam penelitian ini dengan menggunakan kamera dan algoritma YOLOv11, serta notifikasi *real-time* terintegrasi dengan Telegram. Pelatihan model melibatkan dataset citra nyata dan hasil augmentasi, serta dievaluasi menggunakan metrik *precision*, *recall*, dan *mAP*. Hasil pelatihan menunjukkan bahwa model mencapai *precision*, *recall*, dan *mAP@0.5* serta *mAP@0.5:0.95* lebih tinggi dari YOLO versi lainnya dengan peningkatan *precision* sebesar 7,06%, *recall* 4,02%, *mAP@0.5* sebesar 1,42%, dan *mAP@0.5:0.95* mencapai peningkatan tertinggi sebesar 25,77%. Tingkat *confidence* rata-rata dari sistem adalah 0,63 pada *indoor* dan 0,576 pada *outdoor* yang menunjukkan keandalannya. Sistem juga mengirimkan notifikasi otomatis ke Telegram saat deteksi api terjadi, sehingga mempercepat respons user terhadap insiden kebakaran. Dapat disimpulkan bahwa sistem yang dirancang cocok untuk diterapkan sebagai sistem peringatan dini kebakaran berbasis visi komputer yang dapat diterapkan pada PC dan mudah diintegrasikan ke *smartphone*.

Kata kunci: YOLOv11, deteksi kebakaran, visi komputer, notifikasi Telegram, *mAP*, sistem *real-time*

Abstract

Early fire detection is crucial for fire risk mitigation. This study developed a detection system using a camera and the YOLOv11 algorithm, along with real-time notifications integrated with Telegram. Model training involved real-world image datasets and augmentation results, and was evaluated using precision, recall, and mAP metrics. The training results showed that the model achieved higher precision, recall, and mAP@0.5 and mAP@0.5:0.95 than other YOLO versions, with an increase in precision of 7.06%, recall of 4.02%, mAP@0.5 of 1.42%, and mAP@0.5:0.95 reaching the highest increase of 25.77%. The average confidence level of the system was 0.576 indoors and 0.548 outdoors, demonstrating its reliability. The system also sends automatic notifications to Telegram when a fire is detected, accelerating user response to fire incidents. This study concludes that the designed system is suitable for implementation as a computer vision-based fire early warning system that can be implemented on a PC and easily integrated into a smartphone.

Keywords: YOLOv11, fire detection, computer vision, Telegram notification, *mAP*, real-time system

1. Pendahuluan

Kebakaran merupakan salah satu ancaman serius yang dapat menyebabkan kerusakan besar pada bangunan dan mengancam keselamatan jiwa. Kejadian kebakaran dapat disebabkan oleh berbagai faktor, seperti kelalaian manusia atau kegagalan sistem peralatan. Oleh karena itu, sistem proteksi kebakaran yang efektif sangat penting, termasuk penggunaan Alat Pemadam Api Ringan (APAR), sistem hidran, dan *Fire Alarm System* (FAS). Standar internasional seperti NFPA 10, NFPA 14, dan NFPA 72 memberikan pedoman yang digunakan dalam merancang dan mengevaluasi sistem proteksi kebakaran di berbagai

bangunan [1]. Selain itu, regulasi nasional seperti SNI 03-1746-2000 dan SNI 03-6574-2001 juga menjadi acuan penting untuk memastikan keselamatan kebakaran di gedung-gedung publik. Pengelolaan sistem proteksi kebakaran yang baik dapat meminimalkan risiko kebakaran dan memastikan keamanan penghuni bangunan [2].

Dalam dekade terakhir, berbagai pendekatan telah dikembangkan untuk mendeteksi kebakaran secara otomatis, mulai dari penggunaan sensor suhu, sensor asap, hingga deteksi visual berbasis kamera [3]. Sistem berbasis kamera memiliki keunggulan karena mampu mendeteksi nyala api dan asap secara langsung, serta memberikan

informasi visual yang berguna bagi petugas dalam pengambilan [4].

Seiring perkembangan teknologi visi komputer, pendekatan visual seperti MS-FRCNN menunjukkan performa unggul dalam mendeteksi objek kecil seperti titik api. Model ini menggabungkan Feature Pyramid Network (FPN), Parallel Attention Mechanism (PAM), dan Soft-NMS [5]. Di sisi lain, model CNN pra-latih seperti VGG16, InceptionV3, dan Xception yang dikombinasikan dengan metode *Learning without Forgetting (LwF)*, juga mencatatkan akurasi tinggi (98,72%) dan mampu mengklasifikasikan empat kategori visual kebakaran [6].

Salah satu pendekatan visual yang menunjukkan efektivitas tinggi dalam implementasi sistem deteksi kebakaran adalah pemanfaatan YOLO (*You Only Look Once*), sebuah algoritma deteksi objek berbasis *deep learning* yang mampu melakukan deteksi secara *real-time* [7]. Versi terbaru, YOLOv11, menghadirkan modularitas efisien melalui penggunaan MobileNetV2 dan Attention Mechanism, menjadikannya ringan untuk digunakan di perangkat edge [8], [9].

Pengembangan YOLOv11 juga mencakup pendekatan *meta-learning* yang memungkinkan pelatihan dengan data terbatas [11], serta integrasi segmentasi yang memperjelas objek api di antara latar kompleks [12]. *Backbone* ringan dan attention module meningkatkan efisiensi pemrosesan tanpa menurunkan akurasi [13]. Bahkan dalam kondisi ekstrem seperti kabut dan asap samar, sistem ini tetap menunjukkan performa stabil [14], [15]. YOLOv11 juga banyak diterapkan pada platform UAV untuk pemantauan kebakaran hutan di wilayah sulit dijangkau [14], [16]. Segmentasi *real-time* yang diusungnya mampu mengenali api secara lebih presisi dibanding metode konvensional [12].

Beberapa studi telah mengembangkan YOLOv11 lebih lanjut dengan menggunakan *dataset* buatan dari simulator seperti *Unreal Engine* untuk mengatasi keterbatasan data nyata. Salah satu pendekatan memanfaatkan gambar kebakaran yang dihasilkan melalui AirSim pada platform *Unreal Engine* untuk melatih model deteksi dalam mengenali kebakaran perbukitan dengan jumlah data terbatas [11]. Pendekatan lain menggunakan *dataset* simulasi seperti M4SFWD untuk menguji performa model berbasis YOLOv11 dalam mendeteksi api dan asap di lingkungan alami, yang dinilai efektif dalam menyesuaikan sistem terhadap berbagai skenario dunia nyata [17].

Sistem deteksi kebakaran otomatis berbasis kamera dengan algoritma YOLOv11 yang terintegrasi dengan aplikasi Telegram dirancang untuk bekerja secara *real-time*, akurat, dan efisien pada berbagai kondisi lingkungan. YOLOv10 sebelumnya telah terbukti mampu beroperasi secara efisien pada perangkat *edge*, dan prinsip efisiensi

ini diadopsi dalam pengembangan sistem berbasis YOLOv11 untuk mendukung kebutuhan pemrosesan cepat di perangkat dengan sumber daya terbatas [20]. Selain itu, kemampuan YOLOv11 dalam mengklasifikasikan berbagai jenis kebakaran secara akurat juga telah diuji pada perangkat berdaya rendah, menjadikannya sangat cocok untuk digunakan pada sistem pemantauan tetap maupun yang berbasis kamera [10].

Penelitian ini menghadirkan kebaruan melalui pengembangan sistem deteksi kebakaran otomatis berbasis kamera, yang menggunakan algoritma YOLOv11 dan terintegrasi dengan platform komunikasi Telegram. Sistem ini dirancang untuk mampu melakukan deteksi api dan asap secara *real-time*, mengklasifikasikan jenis kebakaran, serta mengirimkan notifikasi peringatan secara otomatis kepada pengguna. Fokus utama dari penelitian ini adalah merancang sistem yang efisien dan ringan, sehingga dapat diimplementasikan pada perangkat dengan daya komputasi terbatas serta mampu beroperasi secara andal di lingkungan dengan tingkat risiko kebakaran yang tinggi, seperti area permukiman padat, fasilitas industri, dan wilayah rawan kebakaran. Dengan pendekatan ini, sistem yang dikembangkan diharapkan mampu memberikan respon dini yang cepat, akurat, dan adaptif, sehingga berkontribusi dalam menurunkan potensi kerugian dan dampak yang ditimbulkan akibat kebakaran.

2. Metode

2.1. Sistem Deteksi Kebakaran

Sistem deteksi kebakaran adalah komponen utama proteksi aktif yang berfungsi mengidentifikasi tanda awal kebakaran dan memberikan peringatan dini. Selain sensor asap, panas, dan api, teknologi modern kini mengadopsi visi komputer dan kecerdasan buatan untuk deteksi yang lebih cepat dan akurat. Penerapan teknologi deteksi visual dengan kamera dan algoritma *deep learning* seperti YOLOv11 memungkinkan identifikasi api atau asap secara *real-time* dengan akurasi tinggi. Teknologi ini mendukung pemrosesan efisien di perangkat *smartphone* dan dapat diintegrasikan dengan sistem notifikasi IoT untuk respons cepat dan otomatis terhadap kebakaran.

2.2. YOLOv11 Sebagai Detektor Api

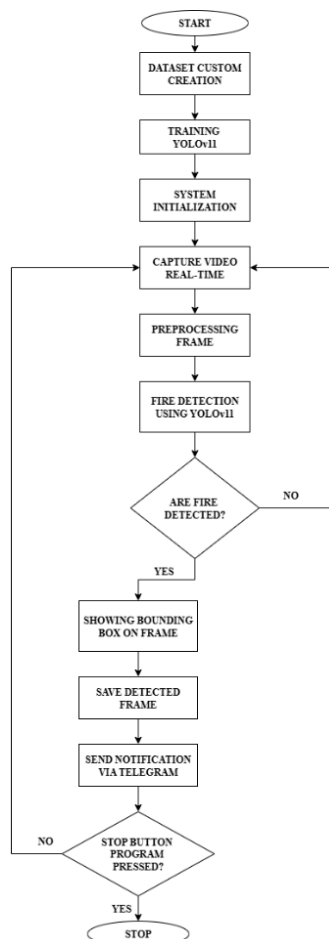
YOLOv11 merupakan versi lanjutan dari arsitektur YOLO yang dirancang untuk deteksi objek *real-time* dengan akurasi tinggi dan efisiensi tinggi. Dengan integrasi MobileNetV2 dan *Attention Mechanism*, model ini mampu mendeteksi objek kecil seperti api dan asap dalam berbagai kondisi pencahayaan. Keunggulannya mencakup klasifikasi multi-kelas kebakaran, minimnya *false positive*, serta kecepatan inferensi tinggi yang cocok untuk implementasi pada perangkat *edge*.

2.3. Telegram Untuk Notifikasi Terkini

Telegram adalah aplikasi pesan instan berbasis cloud yang mendukung integrasi dengan sistem otomatisasi melalui API terbuka. Dalam deteksi kebakaran, Telegram berfungsi mengirim notifikasi real-time ke pengguna, memungkinkan respons cepat melalui perangkat mobile atau desktop. Integrasi ini meningkatkan efisiensi dan responsivitas sistem saat terjadi indikasi kebakaran.

2.4. Desain Sistem

Penelitian ini merupakan studi eksperimental yang merancang sistem deteksi api berbasis kamera menggunakan *deep learning*. Sistem ini menggunakan citra visual tanpa sensor suhu atau asap, dan dapat dijalankan secara lokal di laptop menengah. Tahapan meliputi pengumpulan dan anotasi data, pra-pemrosesan, pelatihan model YOLOv11 di Google Colab, implementasi lokal, dan evaluasi performa di berbagai kondisi.



Gambar 1. Gambar Bagan Desain Sistem Deteksi Api

Sistem deteksi kebakaran ini menggunakan teknologi visi komputer dan *deep learning* dengan algoritma YOLOv11 untuk mendeteksi api secara *real-time* dari video kamera.

Sistem dirancang memberikan peringatan dini melalui notifikasi otomatis ke aplikasi Telegram, sehingga memungkinkan respons cepat terhadap kondisi darurat.

Pengembangan sistem dimulai dari pembuatan *dataset* kustom yang dilabeli secara manual menggunakan Roboflow, dilanjutkan dengan *preprocessing* dan augmentasi data. Model YOLOv11 dilatih menggunakan data ini untuk mengenali pola visual api. Setelah pelatihan, model dijalankan secara lokal, mengolah *input* video kamera menjadi *frame* yang kemudian dianalisis.

Jika terdeteksi api dengan *confidence* score tertentu, sistem menampilkan *bounding box*, menyimpan *frame* sebagai dokumentasi, dan mengirimkan notifikasi ke Telegram lengkap dengan gambar deteksi. Proses ini berjalan secara looping dan dapat dihentikan secara manual atau otomatis. Sistem ini efektif digunakan di berbagai lingkungan seperti industri, kantor, dan area rawan kebakaran.

2.5. Akuisisi dan Anotasi Data

Data citra api diperoleh dari dua sumber dengan *dataset* publik yaitu FIRENET [21] dan BoWFire [6] dan data lapangan yang direkam menggunakan kamera laptop atau USB. Video dari api lilin, korek api, dan nyala pada benda padat dipecah menjadi *frame* per detik dan disimpan sebagai gambar. Anotasi dilakukan manual dengan Label Img, memberi *bounding box* dan label "fire" pada setiap objek api, dengan pemeriksaan ulang untuk memastikan kualitas data pelatihan.

2.6. Pra-Pemrosesan dan Augmentasi Data

Citra yang sudah dianotasi kemudian dipra-pemroses untuk menyesuaikan format input model YOLOv11 dengan mengubah ukuran menjadi 416x416 piksel, menormalisasi piksel, dan mengonversi warna ke RGB menggunakan Python, OpenCV, dan Albumentations. Di platform Roboflow, juga diterapkan preprocessing seperti Auto-Orient, Isolate Objects, dan Dynamic Crop khusus untuk kelas "FIRE" agar fokus pada objek api tetap terjaga. Untuk meningkatkan performa model, dilakukan augmentasi otomatis berupa rotasi hingga 25°, flipping horizontal dan vertikal, *crop* dengan *zoom* 0-20%, serta pengaturan saturasi, *brightness*, *exposure*, dan penambahan *noise*. Data kemudian dibagi menjadi 70% untuk pelatihan, 20% validasi, dan 10% pengujian, agar model mampu mengenali api dalam berbagai kondisi nyata. Teknik augmentasi ini memperkaya variasi data dan membantu meningkatkan akurasi deteksi api.

2.7. Pelatihan Model Deteksi Api

Model yang digunakan dalam penelitian ini adalah YOLOv11, sebuah varian terbaru dari arsitektur YOLO yang dirancang untuk deteksi objek secara real-time dengan akurasi dan efisiensi yang lebih tinggi

dibandingkan versi sebelumnya. Pelatihan model dilakukan di lingkungan Google Colab, yang menyediakan akses GPU secara gratis (NVIDIA Tesla T4) untuk mempercepat proses pelatihan.

Dataset yang telah diproses diunggah ke Google Drive, kemudian dihubungkan ke Colab. Model dilatih menggunakan framework PyTorch dengan parameter sebagai berikut: optimizer Adam, learning rate 0.001, epoch sebanyak 100, dan batch size 16. Evaluasi model dilakukan menggunakan metrik *precision*, *recall*, *F1-score*, dan *mean average precision* (mAP@0.5). Model terbaik disimpan dan diekspor dalam format best.pt untuk digunakan dalam proses inferensi di laptop lokal.

2.8. Implementasi Sistem Inferensi

Api di deteksi menggunakan kamera dengan spesifikasi atau USB digunakan untuk menangkap citra secara real-time. Model YOLOv11 yang telah dilatih di Google Colab diimpor ke dalam aplikasi Python lokal menggunakan OpenCV dan PyTorch. Aplikasi deteksi ini dirancang untuk menampilkan video secara langsung dari kamera, mendeteksi api, dan menampilkan bounding box pada area api yang terdeteksi. Jika terdeteksi adanya api, sistem akan menampilkan notifikasi visual dan mencatat waktu serta gambar sebagai bukti insiden. Sistem tidak memerlukan koneksi internet saat berjalan, sehingga cocok untuk skenario darurat.

2.9. Evaluasi Sistem

Evaluasi pada penelitian ini dilakukan untuk membandingkan tingkat akurasi algoritma YOLOv11 menggunakan confusion matrix sebagai alat ukur kinerja. *Confusion matrix* memberikan informasi mengenai jumlah prediksi yang benar maupun salah berdasarkan empat parameter utama, yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). Matriks ini digunakan untuk mengukur sejauh mana model mampu mengklasifikasikan objek dengan tepat serta mengidentifikasi kesalahan prediksi dalam proses deteksi kebakaran secara otomatis [22].

Tabel 1. Ilustrasi *Confusion Matrix*

	Predicted : Yes	Predicted : No
Actual : Yes	TP	FN
Actual : No	FP	TN

Selain menggambarkan struktur *confusion matrix*, Table 1 juga merangkum empat metrik evaluasi utama, yaitu *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). Berdasarkan nilai-nilai tersebut, dilakukan perhitungan terhadap metrik kinerja model berupa *accuracy*, *precision*, *recall*, dan *F1-Score*. Rumus perhitungan masing-masing metrik dirumuskan sebagai berikut:

- 1) *Precision* merupakan sejauh mana prediksi positif model benar.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

- 2) *Recall* merupakan kemampuan model untuk mengidentifikasi semua data positif yang sebenarnya.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

- 3) *F1 Score* merupakan keseimbangan antara *precision* dan *recall* dalam satu nilai, yang membantu mengevaluasi kinerja model.

$$F1Score = \frac{2 \times recall \times precision}{recall + precision} \quad (3)$$

- 4) *Mean Average Precision* (mAP) merupakan presisi rata-rata yang sangat sering digunakan dalam mengukur keakuratan model deteksi.

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

3. Hasil dan Pembahasan

3.1. Hasil *Training* Deteksi Api Google Colab

Model dilatih menggunakan Google Colab dengan GPU Tesla T4 dan framework Ultralytics YOLOv11s. Dataset terdiri dari 1921 gambar, dengan 1540 gambar untuk training dan 465 gambar untuk validasi, dilatih selama 100 epoch menggunakan resolusi gambar 640x640 piksel.

Tabel 2. Hasil *Training* Deteksi Api di Google Colab

No.	Parameter Evaluasi	Hasil
1	<i>Precision</i> (Box(P))	0.97
2	<i>Recall</i> (R)	0.88
3	mAP@0.5	0.926
4	mAP@0.5:0.95	0.781

Berdasarkan Tabel 2 menunjukkan hasil training model YOLOv11 di Google Colab menunjukkan performa tinggi dengan *precision* 0,97 dan *recall* 0,88, menandakan deteksi yang akurat serta kemampuan yang baik dalam mengenali objek api secara menyeluruh dengan kesalahan minimal.

Nilai mAP@0.5 sebesar 0,926 dan mAP@0.5:0.95 sebesar 0,781 menunjukkan bahwa model YOLOv11 mampu mendeteksi api secara akurat dan stabil pada berbagai tingkat ketelitian, menandakan bahwa model telah terlatih dengan baik dan memiliki performa deteksi yang andal.

Perbandingan hasil training dilakukan dengan menggunakan tiga versi model YOLO, yaitu YOLOv8, YOLOv9, dan YOLOv10, untuk mengevaluasi kinerja deteksi kebakaran berdasarkan *precision*, *recall*, mAP@0,5, dan mAP@0,5:0,95.

Tabel 3. Perbandingan Pelatihan Deteksi Api YOLOv8, YOLOv9, YOLOv10, dan YOLOv11

Parameter Evaluasi	YOLO v8 [17]	YOLO v9 [18]	YOLO v10 [23]	YOLO v11
Precision (Box(P))	0,759	0,872	0,906	0,97
Recall (R)	0,667	0,783	0,846	0,88
mAP@0.5	0,746	0,871	0,913	0,926
mAP@0.5:0.95	0,436	0,650	0,621	0,781

Berdasarkan Tabel 3 menunjukkan performa YOLOv11 meningkat signifikan dibandingkan YOLOv8, dengan nilai *precision* tertinggi sebesar 0,97, nilai *recall* stabil sebesar 0,88, serta mAP@0.5 sebesar 0,926 dan mAP@0.5:0.95 sebesar 0,781, menandakan deteksi yang lebih akurat, sensitif, dan andal di berbagai tingkat *threshold*.

Secara keseluruhan, YOLOv11 menunjukkan kinerja paling unggul dalam mendeteksi kebakaran, baik dari segi ketepatan prediksi maupun generalisasi model, menjadikannya pilihan terbaik di antara keempat versi untuk tugas deteksi api.

3.2. Hasil Training Confusion Matrix Deteksi Api Google Colab

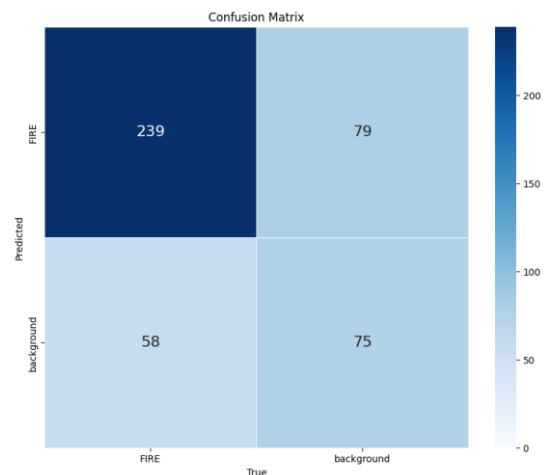
Confusion Matrix digunakan untuk mengevaluasi kinerja klasifikasi model dalam mengenali objek “fire” (kebakaran) dan “background” (latar belakang).

Tabel 4. Hasil Training Confusion Matrix Deteksi Api Google Colab

Keterangan	Nilai	Deskripsi
True Positive (TP)	239	Model berhasil mengenali gambar kebakaran dengan benar
False Positive (FP)	79	Model salah mengenali latar belakang sebagai kebakaran
False Negative (FN)	58	Model gagal mengenali kebakaran dan menganggapnya sebagai latar belakang
True Negative (TN)	75	Tidak ada latar belakang yang dikenali dengan benar

Berdasarkan Tabel 4 menunjukkan model YOLOv11 menunjukkan performa deteksi yang baik dengan 239 *True Positive*, namun masih terdapat 79 *False Positive* dan 58 *False Negative*, mengindikasikan perlunya peningkatan akurasi untuk meminimalkan kesalahan deteksi pada kondisi visual kompleks.

Sementara itu, nilai *True Negative* sebesar 75 menunjukkan kemampuan model dalam membedakan latar belakang, namun tingginya jumlah *False Positive* dan *False Negative* menandakan perlunya peningkatan akurasi dan sensitivitas, terutama pada kondisi visual yang kompleks.



Gambar 2. Hasil Training Confusion Matrix Deteksi Api Google Colab

Berdasarkan Gambar 2 menunjukkan hasil *confusion matrix* menunjukkan bahwa model mendeteksi api dengan cukup baik, menghasilkan 239 *True Positive* dan 75 *True Negative*. Namun, masih terdapat 79 *False Positive* dan 58 *False Negative*, yang mengindikasikan perlunya peningkatan akurasi dan sensitivitas dalam mendeteksi api dan membedakan latar belakang secara lebih tepat.

Sementara itu, nilai *True Positive* yang tinggi menandakan keandalan model dalam mendeteksi api. Namun, tingginya *False Positive* berisiko menimbulkan *false alarm*, sementara *False Negative* mencerminkan potensi bahaya karena api yang sebenarnya tidak terdeteksi.

Secara keseluruhan, meskipun model menunjukkan ketepatan yang dominan, perlu dilakukan peningkatan terutama dalam mengurangi kesalahan deteksi agar sistem lebih akurat dan dapat diandalkan dalam kondisi nyata, khususnya untuk aplikasi keamanan kebakaran.

Tabel 5. Hasil Evaluasi Kinerja Model YOLOv11

Metrik	Rumus	Perhitungan	Nilai (%)
Akurasi	$(TP + TN) / (TP + FP + FN + TN)$	$(239 + 0) / (239 + 79 + 58 + 0)$	63.56%
Precision (Fire)	$TP / (TP + FP)$	$239 / (239 + 79)$	75.16%
Recall (Fire)	$TP / (TP + FN)$	$239 / (239 + 58)$	80.45%
F1-Score	$2 \times (Precision \times Recall) / (Precision + Recall)$	$2 \times (0.7516 \times 0.8045) / (0.7516 + 0.8045)$	77.72%

Berdasarkan Tabel 5 menunjukkan hasil evaluasi menggunakan *confusion matrix*, model YOLOv11 menghasilkan akurasi sebesar 63,56%, dengan *precision* untuk kelas kebakaran sebesar 75,16%, dan *recall* sebesar 80,45%. Nilai *precision* yang cukup tinggi menunjukkan bahwa sebagian besar prediksi kebakaran yang dilakukan model adalah benar, sementara nilai *recall* mengindikasikan kemampuan model dalam mengenali

sebagian besar kejadian kebakaran dari seluruh kasus yang ada. Selain itu, nilai F1-score sebesar 77,72% menunjukkan keseimbangan yang baik antara *precision* dan *recall* dalam performa deteksi kebakaran.

3.3. Hasil Training Grafik Deteksi Api Google Colab

Grafik menunjukkan bahwa model YOLOv11 mengalami peningkatan performa signifikan selama 100 epoch pelatihan.

Tabel 6. Ringkasan Nilai Loss Selama Proses Pelatihan YOLOv11

Jenis Loss	Nilai Awal	Nilai Akhir/Stabil	Keterangan
train/box_loss	≈ 0.43	≈ 0.12	Bounding box loss menurun signifikan → akurasi prediksi kotak meningkat
train/cls_loss	≈ 0.45	≈ 0.10	Kesalahan klasifikasi turun drastis → model makin yakin membedakan kelas
train/dfl_loss	≈ 1.02	≈ 0.85	Distribusi lokasi objek makin presisi
val/box_loss	≈ 1.45	≈ 1.10	Validasi bounding box loss menurun stabil → tidak ada overfitting
val/cls_loss	≈ 1.25	≈ 1.00	Kesalahan klasifikasi validasi menurun secara bertahap
val/dfl_loss	≈ 1.40	≈ 1.10	Validasi prediksi distribusi lokasi membaik konsisten

Berdasarkan Tabel 6 menunjukkan hasil selama proses pelatihan, seluruh jenis loss menunjukkan penurunan yang signifikan, menandakan peningkatan performa model. Train/box_loss turun dari sekitar 0,43 menjadi 0,12, menunjukkan peningkatan akurasi dalam prediksi *bounding box*. Train/cls_loss juga turun drastis dari 0,45 ke 0,10, mengindikasikan model semakin akurat dalam membedakan kelas. Train/dfl_loss menurun dari 1,02 ke 0,85, yang berarti prediksi posisi objek semakin presisi.

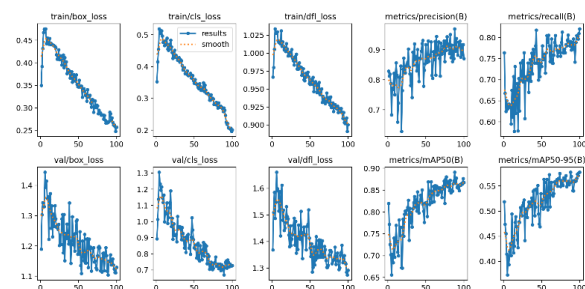
Pada data validasi, semua nilai loss juga menurun dengan stabil tanpa menunjukkan indikasi *overfitting*. Penurunan val/box_loss dan val/cls_loss menunjukkan generalisasi model yang baik, sedangkan val/dfl_loss yang menurun konsisten memperkuat bahwa model mempertahankan presisi lokasi pada data yang belum pernah dilihat sebelumnya. Hal ini menunjukkan bahwa pelatihan berjalan efektif dan model terlatih dengan baik.

Berdasarkan Tabel 7 menunjukkan hasil percobaan menunjukkan peningkatan performa model YOLOv11 yang sangat signifikan. *Precision* meningkat dari kurang dari 0,2 menjadi lebih dari 0,9, menandakan lonjakan akurasi dalam mendeteksi objek kebakaran. *Recall* juga naik tajam dari di bawah 0,3 ke sekitar 0,88, mencerminkan kemampuan model yang jauh lebih baik dalam mengenali semua objek yang relevan.

Nilai mAP@0.5 melonjak dari kurang dari 0,1 ke sekitar 0,93, menunjukkan akurasi prediksi *bounding box* yang sangat tinggi. Sementara itu, mAP@0.5:0.95 naik dari hampir 0 menjadi 0,781, menandakan bahwa model mampu mendeteksi objek secara konsisten pada berbagai tingkat ketelitian IoU. Peningkatan ini mencerminkan efektivitas proses pelatihan dan keandalan model dalam tugas deteksi kebakaran.

Tabel 7. Peningkatan Metrik Performa Model YOLOv11

Metrik	Nilai Awal	Nilai Akhir	Keterangan
<i>Precision</i>	< 0.2	> 0.9	Peningkatan signifikan dalam akurasi prediksi objek kebakaran.
<i>Recall</i>	< 0.3	≈ 0.88	Kemampuan model dalam mengenali objek kebakaran meningkat pesat.
mAP@0.5	< 0.1	≈ 0.93	Rata-rata presisi pada IoU 0.5 membaik secara drastis.
mAP@0.5:0.95	≈ 0.0	≈ 0.781	Konsistensi deteksi pada berbagai tingkat <i>threshold</i> IoU terus meningkat stabil.



Gambar 3. Hasil Training Grafik Deteksi Api Google Colab

Berdasarkan Gambar 3 menunjukkan hasil *training* grafik menunjukkan tren pelatihan yang stabil dan positif selama 100 epoch. Nilai loss untuk train/box_loss, train/cls_loss, dan train/dfl_loss mengalami penurunan tajam, menandakan bahwa model semakin baik dalam memprediksi lokasi dan kelas objek. Hal serupa terlihat pada val/box_loss, val/cls_loss, dan val/dfl_loss, yang juga menurun secara konsisten tanpa gejala *overfitting*.

Pada metrik performa, terjadi peningkatan signifikan pada *precision*, *recall*, mAP@0.5, dan mAP@0.5:0.95, menunjukkan bahwa model semakin akurat dan sensitif dalam mendeteksi api. Lonjakan stabil pada semua metrik validasi ini mencerminkan bahwa model berhasil belajar dengan baik dan memiliki generalisasi yang kuat terhadap data yang tidak dilatih.

3.4. Hasil Pengujian Deteksi Api

Pada pengujian deteksi api menjelaskan tentang kondisi *indoor* dan *outdoor* untuk mengetahui performa sistem yang di bangun.

3.4.1. Hasil Pengujian Deteksi Api pada Kondisi Indoor

Berdasarkan hasil pengujian yang telah dilakukan, deteksi api pada kondisi *indoor* menunjukkan performa sistem yang dibangun.

Tabel 8. Hasil Pengujian Deteksi Api pada Kondisi Indoor

No.	Kondisi Pencahayaan	Status	Hasil Deteksi	Confidence
1	Terang alami	Ada api	Terdeteksi	0,65
2	Redup	Ada api	Terdeteksi	0,66
3	Lampu putih buatan	Ada api	Tidak terdeteksi	-
4	Pencahayaan redup total	Ada api	Terdeteksi	0,64
5	Terang + pantulan kaca	Ada api	Terdeteksi	0,67
6	Siluet (api di belakang)	Ada api	Terdeteksi	0,62
7	Terang	Ada api	Terdeteksi	0,63
8	Lampu kuning redup	Ada api	Tidak terdeteksi	-
9	Redup + asap tipis	Ada api	Terdeteksi	0,59
10	Pencahayaan tidak merata	Ada api	Terdeteksi	0,58
Rata-rata				0,63

Berdasarkan Tabel 8 menunjukkan pengujian sistem deteksi api dalam kondisi *indoor* dengan 10 variasi pencahayaan menunjukkan bahwa model memiliki tingkat adaptabilitas yang cukup baik terhadap berbagai jenis pencahayaan. Sistem berhasil mendeteksi keberadaan api pada 8 dari 10 skenario, dengan nilai *confidence* berkisar antara 0,58 hingga 0,67, dan rata-rata *confidence* sebesar 0,63.

Namun, terdapat dua kondisi pencahayaan spesifik, yakni penggunaan lampu putih buatan dan lampu kuning redup, di mana sistem gagal mendeteksi api. Hal ini mengindikasikan bahwa model masih memiliki keterbatasan dalam mengenali api di bawah spektrum pencahayaan buatan dengan intensitas rendah. Kondisi semacam ini umum terjadi di ruang dalam gedung, terutama di malam hari atau ruangan dengan pencahayaan minim.

Hasil ini menguatkan bahwa sistem berbasis YOLOv11 cukup andal dalam mendeteksi api pada pencahayaan alami dan semi-alami, tetapi masih memerlukan peningkatan pada kondisi dengan pencahayaan ekstrem atau buatan. Oleh karena itu, pengujian lanjutan dengan pengaturan pencahayaan yang lebih variatif dan realistis perlu dilakukan agar sistem memiliki robustnes yang lebih tinggi dan mampu diandalkan dalam situasi nyata di lingkungan *indoor* yang kompleks.

3.4.2. Hasil Pengujian Deteksi Api pada Kondisi Outdoor

Berdasarkan hasil pengujian yang telah dilakukan, deteksi api pada kondisi *outdoor* menunjukkan performa sistem yang dibangun.

Tabel 9. Hasil Pengujian Deteksi Api pada Kondisi Outdoor

No.	Kondisi Pencahayaan	Status	Hasil Deteksi	Confidence
1	Terang alami	Ada api	Terdeteksi	0,58
2	Redup	Ada api	Terdeteksi	0,56
3	Lampu putih buatan	Ada api	Tidak terdeteksi	-
4	Pencahayaan redup total	Ada api	Terdeteksi	0,57
5	Terang + pantulan kaca	Ada api	Terdeteksi	0,55
6	Siluet (api di belakang)	Ada api	Terdeteksi	0,61
7	Terang	Ada api	Terdeteksi	0,60
8	Lampu kuning redup	Ada api	Tidak terdeteksi	-
9	Redup + asap tipis	Ada api	Terdeteksi	0,59
10	Pencahayaan tidak merata	Ada api	Terdeteksi	0,55
Rata-rata				0,576

Berdasarkan Tabel 9 menunjukkan pengujian sistem deteksi api dalam kondisi *outdoor* dengan 10 variasi pencahayaan menunjukkan bahwa model memiliki kemampuan adaptasi yang cukup baik terhadap berbagai kondisi pencahayaan luar ruangan. Sistem berhasil mendeteksi keberadaan api pada 8 dari 10 skenario, dengan nilai *confidence* berkisar antara 0,55 hingga 0,61, dan rata-rata *confidence* sebesar 0,576.

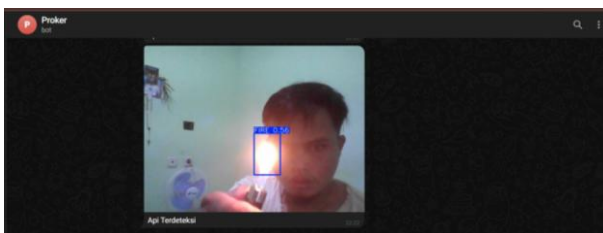
Namun, terdapat dua kondisi pencahayaan spesifik, yaitu lampu putih buatan dan lampu kuning redup, di mana sistem gagal mendeteksi api. Kegagalan ini menunjukkan bahwa model masih menghadapi tantangan dalam mengenali api di bawah pencahayaan buatan berintensitas rendah dengan suatu kondisi yang umum terjadi di lingkungan *outdoor* pada malam hari, seperti area parkir, taman, atau halaman belakang dengan pencahayaan minimal.

Hasil ini menunjukkan bahwa sistem berbasis YOLOv11 tetap mampu mendeteksi api dengan stabil pada pencahayaan alami dan semi-alami di luar ruangan, meskipun performanya sedikit lebih rendah dibandingkan dengan kondisi *indoor*. Hal ini memperkuat pentingnya pengembangan lanjutan, khususnya untuk meningkatkan deteksi pada kondisi pencahayaan ekstrem dan buatan yang tidak teratur.

Hasil ini menguatkan bahwa sistem berbasis YOLOv11 cukup andal dalam mendeteksi api pada pencahayaan alami dan semi-alami, tetapi masih memerlukan peningkatan pada kondisi dengan pencahayaan ekstrem atau buatan. Oleh karena itu, pengujian lanjutan dengan pengaturan pencahayaan yang lebih variatif dan realistis perlu dilakukan agar sistem memiliki robustnes yang lebih tinggi dan mampu diandalkan dalam situasi nyata di lingkungan *outdoor* yang kompleks.

3.5. Hasil Deteksi Api pada Notifikasi Telegram

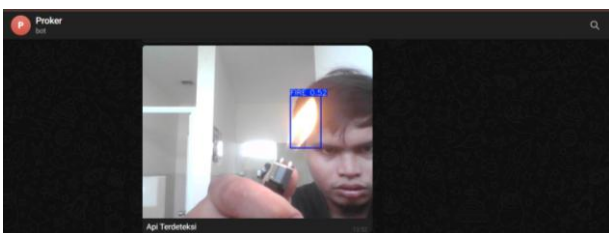
Dalam pengujian deteksi api *indoor* menunjukkan sistem berhasil mengirimkan notifikasi *real-time* via Telegram.



Gambar 4. Hasil Deteksi Api Pada Kondisi Indoor Notifikasi Telegram

Berdasarkan hasil pengujian deteksi api pada kondisi *indoor*, sistem menunjukkan kapabilitas dalam mengidentifikasi keberadaan api, di mana informasi deteksi tersebut secara otomatis dikirimkan melalui notifikasi Telegram kepada pengguna yang berwenang, sehingga memungkinkan respons cepat dan mitigasi dini potensi bahaya kebakaran.

Selanjutnya, pada pengujian deteksi api di area *outdoor*, notifikasi Telegram juga berhasil mengindikasikan adanya kebakaran yang terdeteksi oleh sistem, memastikan pemantauan yang komprehensif di berbagai lingkungan.



Gambar 5. Hasil Deteksi Api Pada Kondisi Outdoor Notifikasi Telegram

Berdasarkan hasil pengujian yang telah dilakukan, deteksi api pada kondisi *outdoor* menunjukkan kapabilitas sistem dalam mengidentifikasi keberadaan api, di mana informasi deteksi tersebut secara otomatis diteruskan melalui notifikasi Telegram kepada pengguna yang berwenang.

4. Kesimpulan

Berdasarkan kesimpulan dari sistem deteksi kebakaran berbasis YOLOv11 yang terintegrasi dengan Telegram berhasil menunjukkan performa tinggi dengan *precision* 0,97, *recall* 0,88, dan *mAP@0.5* sebesar 0,926. Kecepatan inferensi 5,9 ms per citra membuktikan efisiensinya untuk deteksi *real-time*. Pengujian *indoor* dan *outdoor* menunjukkan stabilitas kinerja dengan rata-rata *confidence* masing-masing 0,63 dan 0,576. Sistem juga terbukti mampu mengirimkan peringatan secara *real-time* melalui Telegram, sehingga meningkatkan respons terhadap potensi kebakaran. Temuan ini menunjukkan bahwa sistem memiliki potensi besar sebagai solusi peringatan dini, dengan peluang pengembangan lanjutan pada dataset, arsitektur, dan parameter model untuk peningkatan akurasi dan reliabilitas. Untuk ke depan, disarankan dilakukan pengembangan lebih lanjut, seperti optimasi *dataset* pelatihan, peningkatan arsitektur model, serta penyesuaian parameter deteksi guna meningkatkan akurasi dan reliabilitas sistem secara menyeluruh.

Referensi

- [1]. S. M. Ibrahim, J. A. Muhammad, and K. Widodo, "Evaluasi Sistem Proteksi Kebakaran Aktif pada Gedung Administrasi Perusahaan Listrik Surabaya," 2025.
- [2]. E. Januari *et al.*, "Nyak Syekh Universitas Abulyatama Aceh Evaluation of Fire Evacuation Routes in the Bale," no. 26, pp. 437–452, 2024.
- [3]. Pratama dkk., "Rancang Bangn Pendeteksi Kebakaran Menggunakan Telegram Berbasis IOT," *J. Comput. Sci.* ..., no. 2, pp. 127–135, 2023, [Online]. Available: [https://jurnal.ulb.ac.id/index.php/JCoInT/article/view/127-135/3926](https://jurnal.ulb.ac.id/index.php/JCoInT/article/view/127-135%0Ahttps://jurnal.ulb.ac.id/index.php/JCoInT/article/view/127-135/3926)
- [4]. L. T. Ramos, E. Casas, C. Romero, F. Rivas-Echeverría, and E. Bendek, "A study of YOLO architectures for wildfire and smoke detection in ground and aerial imagery," *Results Eng.*, vol. 26, no. April, p. 104869, 2025, doi: 10.1016/j.rineng.2025.104869.
- [5]. L. Zhang, M. Wang, Y. Ding, and X. Bu, "MS-FRCNN: A Multi-Scale Faster RCNN Model for Small Target Forest Fire Detection," *Forests*, vol. 14, no. 3, 2023, doi: 10.3390/f14030616.
- [6]. V. E. Sathishkumar, J. Cho, M. Subramanian, and O. S. Naren, "Forest fire and smoke detection using deep learning-based learning without forgetting," *Fire Ecol.*, vol. 19, no. 1, 2023, doi: 10.1186/s42408-022-00165-0.
- [7]. E. H. Alkhamash, "A Comparative Analysis of YOLOv9, YOLOv10, YOLOv11 for Smoke and Fire Detection," *Fire*, vol. 8, no. 1, 2025, doi: 10.3390/fire8010026.
- [8]. H. Kwon, S. Choi, W. Woo, and H. Jung, "Evaluating Segmentation-Based Deep Learning Models for Real-Time Electric Vehicle Fire Detection," *Fire*, vol. 8, no. 2, 2025, doi: 10.3390/fire8020066.
- [9]. Y. Tao, B. Li, P. Li, J. Qian, and L. Qi, "Improved Lightweight YOLOv11 Algorithm for Real-Time Forest Fire Detection," pp. 1–18, 2025.

- [10]. E. H. Alkhamash, "Multi-Classification Using YOLOv11 and Hybrid YOLO11n-MobileNet Models: A Fire Classes Case Study," *Fire*, vol. 8, no. 1, pp. 1–20, 2025, doi: 10.3390/fire8010017.
- [11]. Y. Huo *et al.*, "A Small-Sample Target Detection Method for Transmission Line Hill Fires Based on Meta-Learning YOLOv11," *Energies*, vol. 18, no. 6, pp. 1–14, 2025, doi: 10.3390/en18061511.
- [12]. C. Zhao, L. Zhao, K. Zhang, Y. Ren, H. Chen, and Y. Sheng, "Smoke and Fire-You Only Look Once: A Lightweight Deep Learning Model for Video Smoke and Flame Detection in Natural Scenes," *Fire*, vol. 8, no. 3, pp. 1–16, 2025, doi: 10.3390/fire8030104.
- [13]. W. T. Sung, I. G. T. Isa, C. H. Tseng, and S. J. Hsiao, "Constructing a Lightweight Fire and Smoke Detection through the Improved GhostNet Architecture and Attention Module Mechanism," *IEEE Access*, vol. PP, p. 1, 2025, doi: 10.1109/ACCESS.2025.3550539.
- [14]. T. Luan, S. Zhou, L. Liu, and W. Pan, "Tiny-Object Detection Based on Optimized YOLO-CSQ for Accurate Drone Detection in Wildfire Scenarios," *Drones*, vol. 8, no. 9, p. 454, 2024, doi: 10.3390/drones8090454.
- [15]. S. N. Saydirasulovich, M. Mukhiddinov, O. Djuraev, A. Abdusalomov, and Y. I. Cho, "An Improved Wildfire Smoke Detection Based on YOLOv8 and UAV Images," *Sensors (Basel)*, vol. 23, no. 20, 2023, doi: 10.3390/s23208374.
- [16]. F. Akhmedov, R. Nasimov, and A. Abdusalomov, "Dehazing Algorithm Integration with YOLO-v10 for Ship Fire Detection," *Fire*, vol. 7, no. 9, pp. 1–19, 2024, doi: 10.3390/fire7090332.
- [17]. T. Zhang, F. Wang, W. Wang, Q. Zhao, W. Ning, and H. Wu, "Research on Fire Smoke Detection Algorithm Based on Improved YOLOv8," *IEEE Access*, vol. 12, pp. 117354–117362, 2024, doi: 10.1109/ACCESS.2024.3448608.
- [18]. X. Geng, X. Han, X. Cao, Y. Su, and D. Shu, "YOLOV9-CBM: An improved fire detection algorithm based on YOLOV9," *IEEE Access*, vol. XX, 2025, doi: 10.1109/ACCESS.2025.3534782.
- [19]. M. L. Hoang, "Smart Drone Surveillance System Based on AI and on IoT Communication in Case of Intrusion and Fire Accident," *Drones*, vol. 7, no. 12, 2023, doi: 10.3390/drones7120694.
- [20]. S. H. Patel and K. Prajapati, "Advanced Forest Fire Detection Using YOLOv10: A Deep Learning Approach," vol. 3, pp. 4115–4125, 2024.
- [21]. A. Shees, M. S. Ansari, A. Varshney, M. N. Asghar, and N. Kanwal, "FireNet-v2: Improved Lightweight Fire Detection Model for Real-Time IoT Applications," *Procedia Comput. Sci.*, vol. 218, pp. 2233–2242, 2022, doi: 10.1016/j.procs.2023.01.199.
- [22]. M. Ainur Rohman and D. Arifianto, "Penerapan Metode Euclidean Probality dan Confusion Matrix dalam Diagnosa Penyakit Koi," *JurnalSmartTeknologi*, vol. 2, no. 2, pp. 122–30, 2021, [Online]. Available: <http://jurnal.unmuhjember.ac.id/index.php/JST>
- [23]. T. T. Huynh, H. T. Nguyen, and D. T. Phu, "Enhancing Fire Detection Performance Based on Fine-Tuned YOLOv10," *Comput. Mater. Contin.*, vol. 81, no. 2, pp. 2281–2298, 2024, doi: 10.32604/cmc.2024.057954.