

RESEARCH ARTICLE

## Sistem Pendeteksi Api Dini Berbasis Raspberry Pi Termonitoring Aplikasi Telegram

Bayu Aji Prakoso, Rizka Reza Pahlevi\* and Febryanti Sthevanie

Fakultas Informatika, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia

\*Corresponding author: [rizkarezap@telkomuniversity.ac.id](mailto:rizkarezap@telkomuniversity.ac.id)

Received on 5 August 2023; accepted on 5 September 2023

### Abstrak

Kebakaran merupakan peristiwa yang menimbulkan terjadinya api, dimana bencana kebakaran dapat menimbulkan kerugian yang sangat besar dan dapat mengancam keselamatan jiwa. Peristiwa kebaran lebih sering disebabkan oleh kelalaian manusia. Banyak penelitian mengenai permasalahan ini, namun mereka hanya berfokus pada akurasi pendeteksian tanpa mempertimbangkan mobilitas dan kemudahan yang pada kenyataannya ini sangat dibutuhkan dalam merancang sistem pendeteksian api. Penelitian ini dirancang untuk membuat sistem pendeteksi api berbasis kamera didalam ruangan, dengan memanfaatkan aplikasi Bot Telegram sebagai layanan notifikasi saat terjadinya kebaran. Pada sistem ini menggunakan *TensorFlow Lite* sebagai kerangka kerja untuk pembelajaran mesin dan model *Yolov5* sebagai model pretraining pendeteksi api lalu *Mini PC* (Raspberry Pi 4 Model B) sebagai pengolah data dan alat pengontrol sistem ini. Setelah dilakukan pengujian sistem pendeteksi api ini mendapatkan akurasi sebesar 85%, dan membuktikan bahwa sistem ini dibangun dengan mobilitas yang baik tanpa mengurangi akurasi pendeteksian api.

**Key words:** kebakaran, pendeteksian, bot telegram, *tensorflow lite*, model *yolov5*, *mini pc*

### Pendahuluan

Rumah merupakan bangunan yang penting karena merupakan tempat tinggal, aset, dan tempat menyimpan kekayaan [1]. Oleh karena itu, keamanan merupakan pertimbangan penting dalam menjaga rumah dari kejadian yang tidak diinginkan atau kecelakaan. Salah satu solusi tradisional untuk keamanan rumah adalah *Closed Circuit Television* (CCTV), yang merupakan alat untuk memantau keadaan di sekitar area rumah. Namun, ada beberapa masalah dengan implementasi CCTV, yaitu tidak memberikan pemberitahuan atau peringatan apapun saat menangkap objek yang mencurigakan, serta menghasilkan konsumsi bandwidth dan media penyimpanan yang besar karena *streaming* dan penyimpanan *video* terus menerus meskipun tidak ada objek atau aktivitas yang mencurigakan [2]. Kebakaran merupakan salah satu bencana yang tinggi angka kejadiannya di Indonesia. Berdasarkan catatan kejadian bencana DIBI Badan Nasional Penanggulangan Bencana (BNPB) hingga tahun 2019 terdapat 2.551 kejadian kebakaran [3]. Saat ini umumnya sistem pendeteksi kebaran umumnya menggunakan sensor gas dan sensor *infrared*. Kedua sensor tersebut umumnya rentan terhadap *false alarm*. *False alarm* terjadi karena pemicu api tidak dideteksi secara visual. Dengan kata lain sumber api dapat dipicu dari sumber yang tidak sah, seperti asap dan korek api [4].

Sistem pendeteksi kebakaran yang diusulkan dalam penelitian ini dapat mengurangi terjadinya false alarm karena mengandalkan metode alarm visual. Sistem ini dirancang dengan metode yang sederhana

tanpa mengurangi akurasi pendeteksian api. Pada sistem ini terdapat layanan notifikasi yang dapat diakses secara jarak jauh, yang akan memberi tahu pengguna apabila terdeteksi titik api di dalam ruangan. Layanan notifikasi tersebut menggunakan aplikasi telegram karena dapat dijalankan pada beragam perangkat dan sistem operasi, tidak hanya pada telepon genggam, tapi juga komputer dan perangkat pintar lainnya [5]. Dengan memanfaatkan fitur bot pada Application Programming Interface (API) yang memungkinkan untuk menerima data berupa pesan singkat [6]. Sistem ini mengusulkan peringatan dini dengan kamera berbasis machine learning menggunakan framework *Tensorflow Lite*. *Tensorflow Lite* adalah seperangkat alat yang memungkinkan pembelajaran mesin berjalan di komputer yang dibatasi dengan melatih kumpulan sampel yang diambil menggunakan library *OpenCV* [7].

Banyak penelitian yang menawarkan penelitiannya untuk solusi peringatan dini deteksi api. Solusinya dengan kecerdasan buatan dengan pembelajaran mesin. Umumnya teknologi yang dapat memecahkan masalah ini adalah pembelajaran mesin berbasis kamera [8,9]. Namun pembelajaran mesin membutuhkan mesin atau sumber daya yang cukup besar. Karena membutuhkan mesin dengan kapasitas yang cukup besar ini membuatnya kurang mobile dan sulit dijangkau oleh orang umum. *Mini Pc* adalah perangkat yang mapu memberikan mobilitas yang tinggi, karena hanya berukuran seperti kartu ID yang memungkinkan membuatnya menjadi *mobile* [7]. Penelitian ini menempatkan prioritas pada akurasi dalam proses deteksi serta kemudahan dalam pembangunan, dengan mempertimbangkan ukuran kecil dan

bobot ringan dari perangkat keras (*hardware*) yang diperlukan untuk mendukung mobilitasnya. Mobilitas dalam konteks ini merujuk pada kemampuan untuk ditempatkan di area yang terbatas tanpa mengganggu atau memerlukan ruang yang luas untuk penempatan sistem. Oleh karena itu, penelitian ini menggunakan Raspberry pi model 4 b sebagai alat untuk mengontrol dan mengorganisir sistem pendeteksi api. *TensorFlow Lite* adalah alat untuk menerapkan pembelajaran mesin ke Mini PcTensorFlow Lite, yang dirancang untuk memberikan kinerja tinggi dengan memori minimal dan konsumsi daya pada perangkat dengan sumber daya terbatas seperti sistem mikrokontroler IoT tertanam dan perangkat yang dapat dikenakan.

*TensorFlow Lite* memungkinkan pengembang membuat model pembelajaran mesin yang kecil dan cepat serta menyediakan alat untuk mengoptimalkan model untuk eksekusi pada perangkat keras dengan daya komputasi terbatas [8]. Oleh karena itu, sistem deteksi dini kebakaran dapat diterapkan pada komputer mini, karena dapat mengubah model machine learning menjadi ukuran yang lebih kecil dan sederhana tanpa mengurangi akurasi pendeteksian.

Ini membutuhkan model terlatih yang dapat berjalan di *framework TensorFlow Lite*. Model YOLOv5 dipilih karena dapat berjalan di *platform TensorFlow Lite*. Model YOLOv5 dapat digunakan dalam sistem ini dengan proses konversi yang disediakan. You Only Look Once (YOLO) adalah algoritma pengenalan objek yang digunakan dalam pembelajaran mesin. YOLOv5 adalah versi terbaru dari YOLO, yang menawarkan kinerja dan efisiensi yang lebih baik dalam pendeteksian objek. Algoritma YOLOv5 menggunakan arsitektur *Convolutional Neural Network (CNN)* dan teknik augmentasi data untuk mendeteksi objek dengan akurasi lebih tinggi dan lebih cepat [9]. Proses mengapa model YOLOv5 dapat digunakan di *TensorFlow Lite* adalah bahwa YOLOv5 dapat dikonversi dan dioptimalkan untuk dijalankan di *TensorFlow Lite*. Dalam hal ini, model YOLOv5 dilatih lalu dikonversi menjadi *TensorFlow Lite*. Model tersebut kemudian dioptimalkan menggunakan alat yang disediakan oleh *TensorFlow Lite* untuk memastikan bahwa model tersebut dapat digunakan pada perangkat yang memiliki keterbatasan daya dengan kecepatan inferensi tinggi. Hal ini memungkinkan model YOLOv5 berjalan secara efisien dalam sistem tersemat menggunakan *TensorFlow Lite*.

Dalam pengimplementasiannya, sistem pendeteksi api harus mampu mengenali bentuk dan ciri khas dari api yang dapat menyebabkan kebakaran. Oleh karena itu, dalam penelitian ini, harus dibuat batasan yang jelas dalam mengkategorikan jenis dan ciri api yang dimaksud, yaitu api yang berpotensi menyebabkan kebakaran dan yang tidak. Hal ini dilakukan untuk memastikan bahwa sistem hanya mengidentifikasi api yang memang berbahaya dan menghindari terjadinya kesalahan identifikasi yang dapat menyebabkan kepanikan atau kerugian yang tidak perlu. Dengan demikian, pengembangan sistem pendeteksi api harus didasarkan pada penelitian dan pengujian yang cermat untuk memastikan bahwa sistem dapat membedakan jenis api dengan akurasi yang tinggi. Solusi untuk meminimalisir kesalahan adalah sistem pendeteksi api harus dilengkapi dengan parameter yang meminimalkan terjadinya kesalahan dan memastikan bahwa notifikasi yang diberikan ke pengguna akurat dan efektif. Maka dari itu, sistem dapat dikonfigurasi untuk memicu notifikasi ketika api telah terdeteksi selama jangka waktu tertentu.

## Tinjauan Pustaka

Beberapa penelitian menawarkan solusi untuk mencegah teradinya kebakaran. Beberapa penelitian ini, berfokus pada peningkatan akurasi deteksi saja. Penelitian yang dilakukan oleh Wang, Zhicheng, dkk. mengusulkan pendekatan deteksi kebakaran baru yang menggunakan *Convolutional Neural Network - Support Vector Machine (CNN-SVM)* dengan menggunakan *Tensorflow*. Pendekatan ini menggabungkan

kekuatan CNN untuk mengekstrak fitur dari citra dan kekuatan SVM untuk membuat keputusan berdasarkan fitur tersebut. Dari pendekatan tersebut didapatkan hasil akurasi deteksi sebesar 96,67% [10]. Penelitian oleh Ramasubramanian, dkk. mengeksplorasi penggunaan *Artificial Intelligence (AI)* untuk mendeteksi kebakaran dalam aplikasi robot pemadam kebakaran. Penelitian ini mengembangkan sistem deteksi kebakaran yang menggunakan *Artificial Neural Network (ANN)* dan *Tensor Neural Network (TNN)* untuk memproses data sensor yang terkait dengan kebakaran. Sistem tersebut kemudian digunakan untuk mengendalikan robot pemadam kebakaran secara otomatis dan mampu mengenali kebakaran dengan tingkat keberhasilan sebesar 98% [11]. Penelitian yang dilakukan Hongyu, Huang, dkk. mengusulkan metode deteksi kebakaran multiskala yang diperbaiki berdasarkan *Convolutional Neural Network (CNN)*. Metode ini mengambil citra kebakaran dari beberapa skala yang berbeda dan memasukkannya ke dalam CNN untuk mendapatkan fitur yang lebih baik. Selain itu, metode ini juga menggunakan strategi dropout untuk menghindari overfitting pada model CNN. Metode ini diuji pada dataset kebakaran yang terdiri dari citra kebakaran dan citra non-kebakaran dan memberikan hasil yang baik dalam mendeteksi kebakaran dengan tingkat keberhasilan sebesar 96,2% [12].

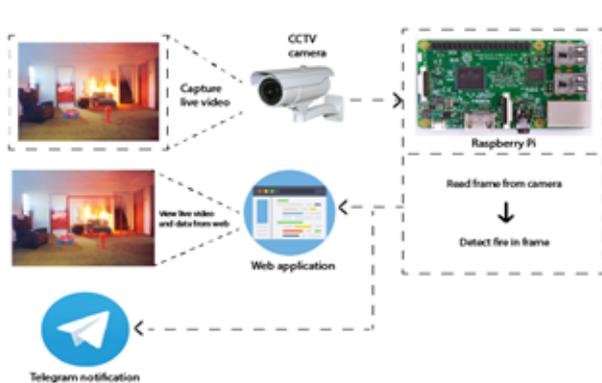
Sebuah sistem deteksi kebakaran real-time yang handal diusulkan oleh Ghani dan Rana Fareed menggunakan *Convolutional Neural Network (CNN)* dan *Long Short-Term Memory (LSTM)*. Sistem ini menggabungkan kekuatan CNN untuk mengekstrak fitur dari citra dengan kekuatan LSTM untuk menangani masalah pengenalan pola jangka panjang. Sistem ini kemudian diuji pada dataset kebakaran yang terdiri dari citra kebakaran dan citra non-kebakaran dan memberikan hasil yang baik dalam mendeteksi kebakaran dengan tingkat keberhasilan sebesar 95,2% [13]. Penelitian oleh Barmpoutis, Panagiotis, dkk. mengusulkan metode deteksi kebakaran dari citra menggunakan Faster R-CNN dan analisis tekstur multidimensional. Metode ini mengekstrak fitur tekstur dari citra kebakaran menggunakan algoritma *Gray-Level Co-occurrence Matrix (GLCM)* dan kemudian memasukkannya ke dalam Faster R-CNN untuk membuat keputusan deteksi kebakaran. Metode ini diuji pada dataset kebakaran yang terdiri dari citra kebakaran dan citra non-kebakaran dan memberikan hasil yang baik dalam mendeteksi kebakaran dengan tingkat keberhasilan sebesar 93,4% [14]. Sebuah algoritma pengenalan kebakaran yang menggunakan fusion fitur dinamis dan klasifikasi *Support Vector Machine (SVM)* berbasis indeks varians *Incremental Variable Span Support Vector Machine (IV-SVM)* yang diteliti oleh Chen, Yuantao, dkk. Algoritma ini mengekstrak fitur dari citra kebakaran pada beberapa skala yang berbeda dan kemudian menggabungkan fitur tersebut menjadi satu untuk digunakan dalam proses pembelajaran klasifikasi. Algoritma ini kemudian diuji pada dataset kebakaran yang terdiri dari citra kebakaran dan citra non-kebakaran dan memberikan hasil yang baik dalam mendeteksi kebakaran dengan tingkat keberhasilan sebesar 97,4% [15]. Berikut ini adalah tabel perbandingan sistem yang ditawarkan untuk pendeteksian api yang ditunjukkan pada Tabel 1.

## Metodologi Penelitian

Dalam penelitian ini, komputer dengan kemampuan terbatas digunakan untuk melakukan pembelajaran mesin. Untuk mengatasi keterbatasan tersebut, Raspberry Pi 4 digunakan sebagai solusi. Raspberry Pi 4 memiliki port khusus yang dapat digunakan untuk memasang kamera, serta ukuran yang relatif kecil (85,6mm x 56,5mm) dan konsumsi daya yang efisien (antara 2,7 Watt dan 6,7 Watt), sehingga sangat cocok untuk aplikasi mobilitas. Dalam proses mendeteksi objek melalui kamera, perpustakaan perangkat lunak yang bernama OpenCV diperlukan. OpenCV dapat menyediakan kemampuan pengambilan gambar.

Table 1. Perbandingan Sistem

No	Judul Penelitian	Tahun	Akurasi	Mobilitas
1	A Novel Fire Detection Approach Based on CNNsVM Using Tensorflow	2017	96.67%	X
2	Fire Detection Using Artificial Intelligence For Fire-fighting Robots	2020	98%	V
3	An Improved Multi-scale Fire Detection Method Based on Convolutional Neural Network (CNN)	2020	96.20%	X
4	Robust Real-Time Fire Detector Using CNN And LSTM	2019	95.20%	X
5	Fire Detection From Images U Faster R-CNN And MultidimensionaI Texture Analysys	2019	93.40%	X
6	The fire recognition algorithm using dynamic feature fusion and IVSVM classifier	2019	97.40%	X
7	Sistem Pendeteksi Api Dini Berbasis Raspberry Pi Termonitoring Aplikasi Telegram	2022	85%	V



Gambar 1. Arsitektur Solusi Sistem

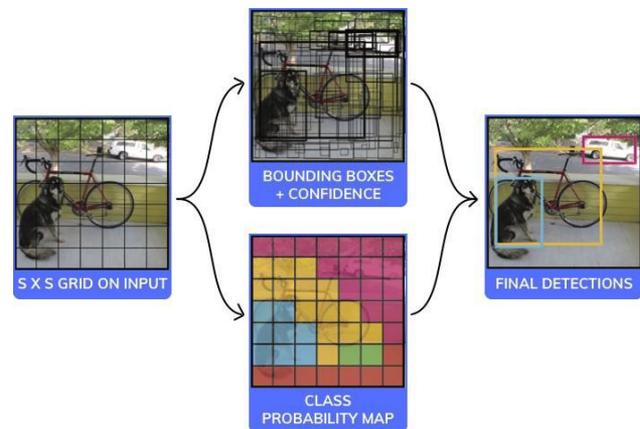
Selain itu, kerangka kerja *Tensorflow Lite* juga digunakan untuk menjalankan pembelajaran mesin, serta model pra-pelatihan Yolov5 sebagai model deteksi api. Gambar 1 menunjukkan sistem yang diusulkan, di mana kamera merekam gambar lingkungan secara *streaming* langsung. Gambar per bingkai kemudian diproses oleh komputer dengan kemampuan terbatas menggunakan *Tensorflow Lite* untuk mendeteksi api dalam bingkai tersebut. Kemudian, Raspberry Pi akan mengirimkan notifikasi peringatan kebakaran ke Telegram melalui Bot Telegram.

### Deteksi Api

Dalam sistem deteksi api yang diusulkan, deteksi api merupakan kegiatan utama yang dilakukan. Objek api merupakan sasaran utama dari sistem ini, sehingga kamera akan terfokus pada deteksi objek tersebut. Setiap bingkai gambar yang diambil akan diekstrak fitur untuk keperluan deteksi objek. Untuk melakukan pembelajaran mesin, sistem akan menggunakan dataset yang terdiri dari gambar api sebagai model dataset. Algoritma Yolov5 kemudian dipilih sebagai metode untuk melatih dataset tersebut, dan kemudian di Convert ke model *Tensorflow Lite* untuk dapat dijalankan pada platform *Tensorflow Lite*. Model Yolov5 dipilih karena merupakan model deteksi objek yang efisien dalam hal waktu dan akurasi, serta dapat dengan mudah dikonvert ke model *Tensorflow Lite* untuk digunakan pada perangkat dengan keterbatasan komputasi, seperti Raspberry Pi

### Notifikasi

Setelah sistem deteksi api mampu mendeteksi adanya api, raspberry pi sebagai sistem pengontrol akan memberikan notifikasi kepada pengguna menggunakan telegram bot sebagai *platformnya*. Proses pengiriman notifikasi ini akan dilakukan setelah sistem mendeteksi adanya



Gambar 2. Arsitektur YOLO

api. Untuk dapat mengirimkan notifikasi melalui telegram bot, raspberry pi pertama-tama harus terdaftar pada platform telegram bot dan mendapatkan token yang akan digunakan untuk autentikasi. Setelah mendapatkan token, raspberry pi dapat mengirimkan request HTTP ke server telegram untuk mengirimkan notifikasi. Request tersebut berisi informasi seperti ID *chat* tujuan, pesan yang ingin dikirim, dan lain-lain. Dengan demikian, sistem deteksi api dapat memberikan notifikasi kepada pengguna melalui telegram bot setelah api terdeteksi.

### Model Pembelajaran Dan Proses pemodelan

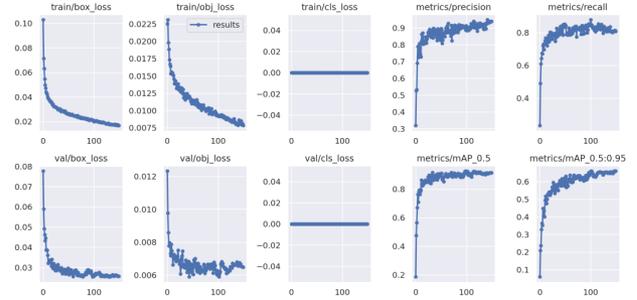
Pembelajaran mesin merupakan teknologi yang digunakan untuk mengidentifikasi pola dalam data dan digunakan untuk mengambil keputusan. Dalam penelitian ini, kami menggunakan model pembelajaran mesin YOLOv5 untuk mendeteksi api. Model YOLOv5 merupakan salah satu model pembelajaran mesin yang digunakan untuk objek deteksi, yang dapat mengidentifikasi objek dalam gambar dengan akurasi yang tinggi [16]. Tujuan dari penggunaan model pembelajaran mesin YOLOv5 dalam penelitian ini adalah untuk meningkatkan akurasi dalam mendeteksi api. Kami memilih untuk menggunakan dataset api yang diambil dari RoboFlow, yang terdiri dari 2.560 gambar. Dataset ini dipilih karena menyediakan data api dengan berbagai kondisi, seperti api yang berbeda-beda ukurannya, posisi dan kondisi lingkungan. Gambar 2 adalah gambaran arsitektur YOLO bekerja.

Algoritma YOLO bekerja dengan membagi gambar menjadi ukuran  $S \times S$ , dimana setiap sel memiliki nilai boundingbox dan nilai Confidence. Kotak pembatas biasanya dinyatakan sebagai empat koordinat  $(x1, y1, x2, y2)$  yang masing-masing mewakili posisi atas, bawah, kiri, dan kanan dari kotak virtual. Confidence adalah nilai antara 0

dan 1 yang menunjukkan kepercayaan model terhadap prediksi subjek. Nilai kepercayaan yang tinggi menunjukkan bahwa model sangat yakin bahwa objek berada di dalam bingkai, sedangkan nilai kepercayaan yang rendah menunjukkan bahwa model tidak yakin bahwa objek berada di dalam bingkai. Selain 6 prediksi kotak pembatas, algoritma ini juga memprediksi kelas objek. Model YOLO dapat dilatih untuk mengenali objek milik kelas yang ditentukan pengembang, dan model tersebut membuat prediksi berdasarkan kelas objek yang teridentifikasi. Setelah nilai prediksi bounding box dan prediksi kelas digabungkan, langkah selanjutnya adalah thresholding dan nonmaximum suppression (NMS). Ambang batas digunakan untuk memastikan bahwa objek diamati dan dilacak hanya ketika skor kepercayaannya melebihi ambang batas tertentu. Ini membantu meminimalkan jumlah deteksi palsu dan memastikan bahwa objek yang terdeteksi benar-benar ada di dalam gambar. Non-Maximum Suppression (NMS) adalah algoritma yang digunakan untuk mencegah deteksi tumpang tindih yang dapat terjadi selama deteksi objek. Di YOLO, NMS digunakan untuk memastikan bahwa setiap objek diamati dan dilacak hanya sekali, meskipun objek tersebut dapat muncul dalam beberapa pengamatan yang berbeda. Kombinasi ambang batas dan NMS membantu membuat deteksi objek lebih efisien dan akurat dengan memastikan bahwa setiap objek diamati dan dilacak.

Proses pemodelan pertama adalah membuat dataset. Dataset berasal dari situs web Roboflow, di mana Roboflow adalah perusahaan yang menyediakan alat untuk membuat dan mengelola dataset untuk aplikasi pembelajaran mesin. Roboflow menyediakan alat untuk mengatur dataset, mengubah format, menambahkan tag, dan menilai dataset. Selain itu, Roboflow juga menawarkan integrasi dengan beberapa framework pembelajaran mesin seperti TensorFlow, Pytorch, dan OpenCV. Kumpulan data berisi 2.560 gambar berasal dari situs ini, terbagi menjadi 70% data latih, 20% testing, dan 10% validasi. Data ini kemudian diolah menggunakan proses YOLO dengan parameter ukuran citra masukan sebesar 416 piksel. Ukuran 416 x 416 piksel adalah ukuran optimal untuk pelatihan model YOLOv5 yang tepat dan menawarkan keseimbangan antara kecepatan dan akurasi. Nilai ukuran tumpukan 16 adalah ukuran yang dapat diterima dan sering digunakan dalam proses pelatihan. Bernilai 150 zaman. Hal ini memungkinkan model untuk melihat kumpulan data sebanyak 150 kali, yang membantu meningkatkan performa dan akurasi. Penggunaan bobot YOLOv5s dalam perintah ini menunjukkan bahwa bobot awal. Hal ini dapat mempercepat proses pelatihan karena memulai dengan bobot yang diketahui dapat menghasilkan kinerja yang lebih baik dan bobot ini juga akan mempengaruhi hasil model karena bobot yang terlalu tinggi akan mengubah model menjadi lebih kompleks, yang akan mempengaruhi perhitungan jika hal ini terjadi. Model ini dioperasikan dalam sistem alarm kebakaran ini. Setelah model YOLOv5 dibuat dengan set data yang dipilih, model tersebut akan dikonversi menjadi model TensorFlow Lite. Karena TensorFlow Lite memungkinkan untuk menjalankan model pembelajaran mesin pada perangkat dengan sumber daya terbatas seperti Raspberry Pi. Dengan menggunakan TensorFlow Lite, kami dapat mengoptimalkan sumber daya tanpa mengorbankan akurasi deteksi kebakaran. Implementasi model machine learning YOLOv5 dengan TensorFlow Lite bertujuan untuk meningkatkan akurasi pendeteksian kebakaran dan dapat dijadikan acuan untuk pengembangan sistem pendeteksi kebakaran di masa mendatang.

Berdasarkan Gambar 3 analisis grafik train/box\_loss, train/obj\_loss, val/box\_loss, dan val/obj\_loss, model YOLO yang diuji telah mencapai konvergensi pada akhirnya, menunjukkan bahwa model telah belajar dan menyesuaikan dengan data latih dengan baik. Dalam hal ini, kita dapat melihat bahwa kerugian objek (obj\_loss) dan kerugian kotak (box\_loss) dari pelatihan dan validasi terus menurun seiring berjalannya waktu dan stabil pada titik tertentu. Selanjutnya, dari analisis metrics/precision, metrics/recall, metrics/mAP.0.5, dan metrics/mAP.0.5:0.95, model juga dapat mengenali objek dengan baik



Gambar 3. Grafik Analisis Pre-Training

dalam dataset validasi seiring berjalannya waktu. Kita dapat melihat bahwa presisi dan recall meningkat seiring bertambahnya iterasi, menunjukkan bahwa model menjadi lebih baik dalam mengenali objek yang ada di dalam gambar. Metrik penting lainnya, yaitu mean average precision (mAP) juga menunjukkan peningkatan seiring waktu. Peningkatan ini terlihat pada kedua nilai threshold 0,5 dan 0,5 hingga 0,95. Hal ini menunjukkan bahwa model 7 telah belajar untuk mengenali objek dengan lebih akurat dan konsisten. Secara keseluruhan, hasil pengujian model menunjukkan bahwa model YOLO yang diuji dapat mengenali objek dengan baik pada dataset validasi dan telah mencapai konvergensi dengan baik. Oleh karena itu, model ini cocok untuk digunakan dalam aplikasi deteksi objek di dunia nyata.

#### Perhitungan Matriks

Perhitungan metrik konfusi digunakan untuk mengevaluasi performa model deteksi api. Metrik konfusi menyajikan jumlah prediksi model yang benar dan salah dalam bentuk tabel yang memudahkan untuk memahami seberapa baik model tersebut bekerja. Berikut adalah cara menghitung matriks konfusi pada deteksi api. Jumlah prediksi deteksi api benar positif (TP), prediksi deteksi api salah positif (FP), prediksi deteksi api benar negatif (TN), dan prediksi deteksi api salah negatif (FN). Nilai-nilai ini dapat diperoleh dari hasil uji coba model deteksi api pada dataset yang telah disiapkan. Rumus ini juga diterapkan pada penelitian sebelumnya[7].

Hitung nilai akurasi (*Accuracy*) dengan menggunakan rumus:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

Hitung nilai presisi (*Precision*) dengan menggunakan rumus:

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

Hitung nilai recall (*Recall*) dengan menggunakan rumus:

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

Hitung nilai F1-score dengan menggunakan rumus:

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \quad (4)$$

#### Data Uji

Data uji digunakan untuk melihat akurasi deteksi sistem yang digunakan. Data diperoleh dari proses pengujian secara *realtime*. Pengujian dilakukan pada ruang yang terdapat api dan tidak terdapat api, pengujian ini diuji selama 2 menit sebanyak 30 kali pada setiap pengujian. Nantinya sistem ini akan menguji api yang akan diuji adalah video api yang

Table 2. Pengujian

Pengujian	Pengujian			
	Ruang ada Api	Ruang Tanpa Api	Notifikasi	Validasi
Mendeteksi Api	25	4	Terkirim	Valid
Gagal Mendeteksi Api	5	26	Tidak Terkirim	Valid
Total	30	30		60



Gambar 4. Proses Deteksi Sistem saat Pengujian

dijalankan melalui layar komputer, selain untuk mendapatkan tampilan dari api yang terlihat nyata juga untuk keamanan pada saat pengujian berlangsung agar tidak terjadi hal yang tidak diinginkan saat pengujian. Hasil dari pengujian ini kemudian dihitung secara manual dengan mempertimbangkan kapan pengujian ini dilakukan dalam kenyataannya. Gambar 4 menunjukkan hasil pada saat proses pengujian sedang berlangsung.

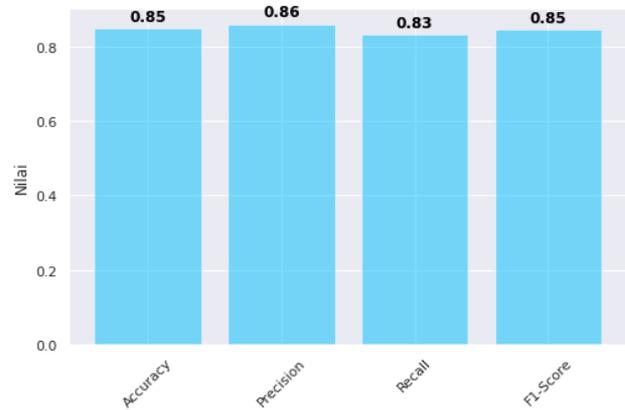
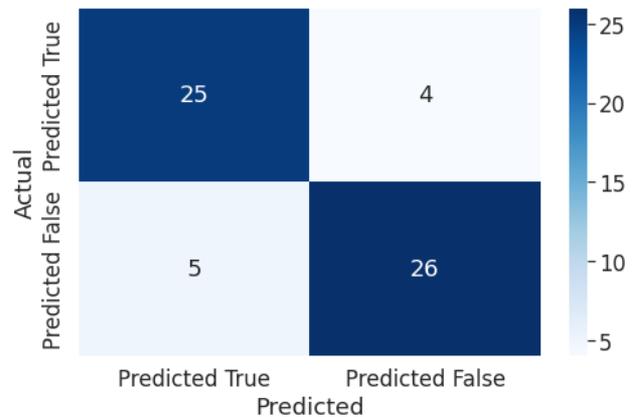
## Hasil dan Pembahasan

### Hasil Pengujian

Pengujian sistem deteksi api telah dilakukan pada dua kondisi: ruangan yang terdapat api dan ruangan yang tidak terdapat api. Masing-masing pengujian dilakukan selama 2 menit, dengan jumlah iterasi sebanyak 8-30 kali. Hasil pengujian menunjukkan bahwa pada kondisi ruangan tanpa api, sistem berhasil tidak mendeteksi api dengan benar sebanyak 26 kali (validitas sebesar 86,7%) dan salah mendeteksi api sebanyak 4 kali (validitas sebesar 13,3%). Sementara itu, pada kondisi ruangan dengan api, sistem berhasil mendeteksi api sebanyak 25 kali (validitas sebesar 83,3%) dan gagal mendeteksi api sebanyak 5 kali (validitas sebesar 16,7%). Gagalnya sistem dalam mendeteksi api pada kondisi tersebut disebabkan oleh masuknya cahaya terlalu terang ke dalam ruangan pengujian. Data pengujian ditunjukkan oleh Tabel 2.

### Analisis Hasil Pengujian

Hasil pengujian deteksi api menunjukkan bahwa sistem berhasil mendeteksi api dengan akurasi sebesar 85% dengan persamaan (1). Ini artinya, dari seluruh pengujian yang dilakukan, sistem mampu secara akurat mendeteksi keberadaan api. Selain akurasi, hasil pengujian juga menunjukkan bahwa sistem memiliki presisi sebesar 86% dengan persamaan (2). Recall sistem sebesar 83% dengan persamaan (3). F1-score sistem sebesar 85% dengan persamaan (4), yang menunjukkan bahwa sistem memiliki kinerja yang baik dalam mendeteksi api. Dari hasil pengujian ini, dapat disimpulkan bahwa sistem deteksi api yang telah dibuat memiliki akurasi, presisi, *recall*, dan F1-score yang cukup tinggi. Ini menunjukkan bahwa sistem mampu secara efektif mendeteksi keberadaan api dan juga selain mobilitas yang baik sistem ini juga

Gambar 5. Diagram perbandingan Akurasi, Presisi, *Recall*, dan F-1 ScoreGambar 6. Diagram *Confusion Matrix* Pengujian

tidak lupa memperhatikan akurasi yang baik, sehingga dapat memberikan perlindungan terhadap kebakaran. Namun, agar kinerja sistem dapat terus ditingkatkan, perlu dilakukan pengujian ulang dengan kondisi yang berbeda, seperti variasi intensitas cahaya. Gambar 5 adalah gambar perbandingan diagram batang Akurasi, Presisi, *Recall*, dan F1-Score dan gambar 6 adalah gambar diagram metrik konfusi.

## Kesimpulan

Kebakaran merupakan bencana yang dapat menimbulkan kerugian besar dan mengancam keselamatan jiwa. Banyak penelitian telah dilakukan untuk mengatasi masalah ini, namun kebanyakan hanya berfokus pada akurasi pendeteksian tanpa memperhatikan mobilitas dan kemudahan. Penelitian ini bertujuan untuk membuat sistem pendeteksi api berbasis kamera di dalam ruangan yang menggunakan

aplikasi Bot Telegram sebagai layanan notifikasi saat terjadinya kebakaran. Sistem ini menggunakan *TensorFlow Lite* sebagai kerangka kerja untuk pembelajaran mesin dan model YOLOv5 sebagai model pretraining pendeteksi api, serta *Mini PC* (Raspberry Pi 4 Model B) sebagai pengolah data dan alat pengontrol. Setelah dilakukan pengujian, sistem ini mendapatkan akurasi sebesar 85% dan membuktikan bahwa sistem ini dapat dibangun dengan mobilitas yang baik tanpa mengurangi akurasi pendeteksian api. Diharapkan dalam penelitian kedepannya, dapat menemukan dan mengembangkan sistem ini agar dapat menemukan solusi untuk mengoptimalkan sistem ini dari gangguan eksternal seperti intensitas cahaya supaya dapat digunakan pada lingkungan *indoor* maupun *outdoor*

## Daftar Pustaka

- Bangali J, Shaligram A. Design and Implementation of Security Systems for Smart Home based on GSM technology. *International Journal of Smart Home*. 2013;7(6):201-8.
- Surantha N, Wicaksono WR. Design of smart home security system using object recognition and PIR sensor. *Procedia Computer Science*. 2018;135:465-72.
- Yakub M, Phuspa SM. Manajemen Risiko Kebakaran pada PT Pertamina EP Asset 4 Field Sukowati. *Journal of Industrial Hygiene and Occupational Health*. 2019;3(2).
- Konaite M, Owolawi PA, Mapayi T, Malele V, Odeyemi K, Aiyetoro G, et al. Smart Hat for the blind with Real-Time Object Detection using Raspberry Pi and TensorFlow Lite. In: *Proceedings of the International Conference on Artificial Intelligence and its Applications*; 2021. p. 1-6.
- Hadisantoso FS. Sistem notifikasi kebakaran gedung menggunakan telegram. *Jurnal Elektra*. 2019;4(2):20-8.
- Maulana MI. Implementasi Bot Telegram pada Proses Retrieval Data dalam Database. *Indonesian Journal of Data and Science*. 2021;2(1):13-20.
- Hadi A, Pahlevi RR, Suryani V. Early Warning System for Physical Distancing Detection in the Prevention of COVID-19 Spread. In: *2021 International Conference on Data Science and Its Applications (ICoDSA)*; 2021. p. 252-6.
- David R, Duke J, Jain A, Reddi VJ, Jeffries N, Li J, et al. *Tensorflow lite micro: Embedded machine learning for tinyml systems*. *Proceedings of Machine Learning and Systems*. 2021;3:800-11.
- Nguyen QT. Detrimental Starfish Detection on Embedded System: A Case Study of YOLOv5 Deep Learning Algorithm and TensorFlow Lite framework. *Journal of Computer Sciences Institute*. 2022;23:105-11.
- Wang Z, Zhang H, Guo X. A novel fire detection approach based on CNN-SVM using tensorflow. In: *International Conference on Intelligent Computing*; 2017. p. 682-93.
- Ramasubramanian S, Muthukumaraswamy SA, Sasikala A. Fire detection using artificial intelligence for fire-fighting robots. In: *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*; 2020. p. 180-5.
- Hongyu H, Ping K, Li F, Huaxin S. An improved multi-scale fire detection method based on convolutional neural network. In: *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*; 2020. p. 109-12.
- Ghani RF. Robust Real-Time Fire Detector Using CNN And LSTM. In: *2019 IEEE Student Conference on Research and Development (SCoReD)*; 2019. p. 204-7.
- Barmpoutis P, Dimitropoulos K, Kaza K, Grammalidis N. Fire detection from images using faster R-CNN and multidimensional texture analysis. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; 2019. p. 8301-5.
- Chen Y, Xu W, Zuo J, Yang K. The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier. *Cluster Computing*. 2019;22(3):7665-75.
- Muhammad K, Ahmad J, Baik SW. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing*. 2018;288:30-42.
- Huang P, Chen M, Chen K, Zhang H, Yu L, Liu C. A combined real-time intelligent fire detection and forecasting approach through cameras based on computer vision method. *Process Safety and Environmental Protection*. 2022;164:629-38.
- Zhu X, Lyu S, Wang X, Zhao Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In: *Proceedings of the IEEE/CVF international conference on computer vision*; 2021. p. 2778-88.