RANCANG BANGUN SISTEM DETEKSI API PADA SISTEM TERTANAM MENGGUNAKAN ALGORITMA YOLOV4 BERBASIS IOT

ANNASTYA BAGAS DEWANTARA

ABSTRAK

Kebakaran pada daerah dalam gedung seperti perumahan, perkantoran dan pemukiman merupakan penyebab tertinggi terjadinya kebakaran, dengan penyebab terbesar pada memasak, malfungsi listrik dan merokok. Pada penerapan pencegahan kebakaran menggunakan perangkat deteksi asap konvensional terdapat keterbatasan dalam mengukur intensitas tingkat kebakaran dan jangkauan terbatas dalam mendeteksi kebakaran. Menggunakan metode pendekatan objek lokalisasi YOLOv4, objek api dapat dideteksi dengan jangkauan yang lebih luas dan akurasi yang lebih tinggi, sehingga kebakaran dapat lebih mudah di antisipasi. Sistem ini menggunakan deteksi objek dengan algoritma YoloV4 pada sistem tertanam Raspberry Pi 4 yang tersambung dengan kamera. Sistem deteksi api mampu menyalakan alarm melalui *piezzo buzzer* dan mampu memberikan *alerting* kepada pengguna melalui sistem berbasis IoT pada *real-time* website. Hasil pengujian menunjukkan bahwa model yang dilatih dengan menggunakan 996 gambar memiliki kemampuan generalisasi yang baik dalam mendeteksi api berukuran besar dan kecil serta mampu memprediksi api di dalam dan di luar ruangan. Model memiliki nilai hasil mAP@0.50 sebesar 0.81, precission sebesar 0.83, recall 0.79 dan F1-Score sebesar 0.81.

Kata Kunci : Kebakaran, YOLOv4, Raspberry Pi, IoT, *Convolutional Neural Network*

DESIGN OF A FIRE DETECTION SYSTEM IN AN EMBEDDED SYSTEM USING IOT-BASED YOLOV4 ALGORITHM

ANNASTYA BAGAS DEWANTARA

ABSTRACT

Fire in indoor areas such as housing, offices and settlements is the leading cause of fires, with the main causes being cooking, electrical malfunction, and smoking. In the implementation of fire prevention using conventional smoke detection devices, there are limitations in measuring the intensity of fire levels and limited scope in detecting fires. Using the YOLOv4 object localization approach method, fire objects can be detected with a wider range and higher accuracy, making it easier to anticipate fires. This system uses object detection with the YoloV4 algorithm on a Raspberry Pi 4 embedded system connected to a camera. The fire detection system is able to turn on an alarm through a piezzo buzzer and able to provide alerting to users through an IoT-based real-time website. Test results show that the model trained using 996 images has good generalization ability in detecting both large and small fires and is able to predict fires inside and outside of rooms. The model has a mAP@0.50 value of 0.81, a precision of 0.83, a recall of 0.79 and an F1-Score of 0.81.

Keywordss : Fire, YOLOv4, Raspberry Pi, IoT, *Convolutional Neural Network*

\

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Pertumbuhan jumlah populasi di dunia kian meningkat tiap tahunnya, dengan angka sebesar 83 juta per tahun atau sebesar 1.1%. Peningkatan populasi global telah tumbuh dari 1 miliyar pada tahun 1800 ke angka 7.9 miliyar di tahun 2020 dan telah di prediksi akan mencapai angka 8.6 miliyar pada pertengahan tahun 2030 dan 9.8 miliyar di tahun 2050 [1] Pertumbuhan dan peningkatan pertumbuhan penduduk akan berbanding lurus dengan peningkatan kepadatan penduduk khususnya di daerah perkotaan. Tingginya aktivitas perkotaan yang diiringi oleh kurangnya perencanaan dan penyediaan lahan pemukiman yang layak mengakibatkan tingkat potensi kebakaran meningkat [2].

Kebakaran pada daerah dalam gedung seperti perumahan, perkantoran dan pemukiman merupakan penyebab tertinggi terjadinya kebakaran, dengan penyebab terbesar pada memasak, malfungsi listrik dan merokok [3]. Tingginya angka kecelakaan dan kematian yang dialami oleh pemadam api dalam usaha memadamkan api akibat terjatuh, tabrakan kendaraan serta luka bakar akibat panas api dan ledakan [4], menjadi landasan pada peningkatan tindakan pencegahan kebakaran. Pada penerapan pencegahan kebakaran menggunakan perangkat deteksi asap konvensional terdapat keterbatasan dalam mengukur intensitas tingkat kebakaran dan jangkauan terbatas dalam mendeteksi kebakaran. Menggunakan metode *Artificial Intelligence* dan *Machine Learning* yang di tambahkan sensor infrared untuk mendeteksi panas, objek api dapat dideteksi dengan jangkauan yang lebih luas dan akurasi yang lebih tinggi.

Beberapa metode yang telah digunakan dalam studi literatur pada deteksi objek api pada *embedded system*, Deteksi objek api dengan menggunakan *Haar Cascade Classifier* pada YoloV3 [5], Deteksi api pada *noisy image* menggunakan *Faster R-CNN* model [6], *Bayes Classifier* dan *Support Vector Machine* (SVM) pada gambar RGB [7] Algoritma *Single Shot*

Multibox Detector (SSD) pada UAV [8] dan penerapan deteksi kebakaran hutan menggunakan EfficientDet-Lite dan Yolov5 [9].

Kebakaran dapat terjadi kapan pun tanpa disadari. Sistem deteksi api yang dapat mengirimkn alarm dan notifikasi kepada pemiliknya akan sangat berguna terutama di beberapa situasi. Sistem ini menggunakan deteksi objek dengan algoritma YoloV4 pada sistem tertanam Raspberry Pi 4 yang tersambung dengan kamera. Lokalisasi objek dilakukan untuk mendeteksi lokasi piksel dari frame, sistem mampu mendeteksi lokasi terjadinya api. Sistem deteksi api mampu menyalakan alarm melalui modul suara yang tertanam padanya, dan mampu mengirimkan notifikasi kepada pengguna melalui sistem IoT via *HTML request* untuk melakukan *monitoring* secara *real-time.*

1.2 Tujuan Penelitian

1. Merancang desain sistem monitoring dari Sistem Deteksi Api.
2. Merancang desain sirkuit alert dari Sistem Deteksi Api.
3. Merancang model deteksi objek dari Sistem Deteksi Api.
4. Melakukan analisis dan evaluasi terhadap hasil *recall, precission* dan *accuracy* dari model yang telah dibuat.

1.3 Rumusan Masalah

1. Bagaimana desain sistem *monitoring* dari Sistem Deteksi Api?
2. Bagaimana desain *sistem* alert dari Sistem Deteksi Api *t*?
3. Bagaimana model deteksi objek dari Sistem Deteksi Api?
4. Bagaimana hasil analisis dan evaluasi terhadap hasil *recall, precission* dan *accuracy* dari model yang telah dibuat?

1.4 Batasan Masalah

1. Klasifikasi dibagi ke dalam tiga kelas yakni api menyala, berasap dan padam.
2. Sistem *monitoring* dilakukan pada server lokal berbasis web.

1.5 Sistematika Penulisan

Penelitian ini disusun dan diuraikan dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan permasalahan, dan sistematika penulisan.

Bab II Tinjauan Pustaka

Bab ini berisikan dasar ilmu yang mendukung dan penelitian terdahulu yang berkaitan dengan penelitian ini.

Bab III Metodologi Penelitian

Bab ini berisi rancangan penelitian berupa metode penelitian digunakan, prosedur penelitian berupa flowchart, dimana dalam tahap ini akan dilakukan pengolahan citra digital sesuai dengan metode dan algoritma yang telah ditetapkan. Penelitian ini menggunakan metode eksperimental dalam melakukan uji coba algoritma Yolov4 pada *embedded system* dalam deteksi api.

Bab IV Hasil dan Pembahasan

Bab ini berisi proses uji coba berdasarkan parameter – parameter yang ditetapkan, dan kemudian dilakukan analisa terhadap hasil uji coba tersebut.

Bab V Penutup

Bab ini berisi kesimpulan yang dapat diambil dari penelitian ini serta saran untuk pengembangan selanjutnya.

BAB 2
LANDASAN TEORI

2.1 State of Art

Penelitian ini dilakukan dengan memperhatikan beberapa penelitian terdahulu yang membahas mengenai tema yang bersinggungan terhadap Sistem Deteksi Api meliputi sistem IoT, penerapan pada sistem tertanam, deteksi objek, dan sistem monitoring . Beberapa penelitian terdahulu dapat dilihat pada tabel 2.1.1 dibawah ini:

Tabel 2.1 State of Art Penelitian

No.	Judul Jurnal	Tipe	Ringkasan
1	M. Kanwar and L. Agilandeeswari, "IOT Based Fire Fighting Robot," 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2018, pp. 718-723, doi: 10.1109/ICRITO.2018. 8748619. [10]	IoT	Sistem monitoring dan kontrol jarak jauh pada robot pemadam api berbasis IoT pada Raspberry Pi dan Arduino yang dihubungkan dengan sensor ultrasonik dan sensor deteksi api.
2	S. Zhao, B. Liu, Z. Chi, T. Li and S. Li, "Characteristics Based Fire Detection System Under the Effect of Electric Fields With Improved Yolo-v4 and	Deteksi Api dengan YoloV4	Mendeteksi api yang muncul akibat penerapan perlakuan efek medan elektrik yang berbeda menggunakan YoloV4.

	ViBe," in IEEE Access, vol. 10, pp. 81899-81909, 2022, doi: 10.1109/ACCESS.2022.3190867. [11]		
3	Kulshreshtha M, Chandra SS, Randhawa P, Tsaramirsis G, Khadidos A, Khadidos AO. OATCR: Outdoor Autonomous Trash-Collecting Robot Design Using YOLOv4-Tiny. Electronics. 2021; 10(18):2292.https://doi.org/10.3390/electronics10182292. [12]	Yolov4 pada Sistem Tertanam	Navigasi kontrol robot menggunakan DC motor yang terhubung *driver* motor L298N serta *stepper* motor berdasarkan koordinat pixel dari objek sampah pada YoloV4.
4	Madhar, Madhar. "Rancang Bangun Sistem Monitoring Deteksi Dini Kebakaran Dengan Fitur Gps Berbasis Website." JATI (Jurnal Mahasiswa Teknik Informatika) 2.1 (2018): 367-372. [13]	Sistem Monitoring Web	Sistem Monitoring web menggunakan ESP8266 menggunakan LM35, DHT11 dan GPS dalam memberikan informasi lokasi dan suhu dalam mendeterminasikan tingkat kebakaran.
5	Suwarjono, Suwarjono, Izak Habel Wayangkau, Teddy Istanto, Rachmat Rachmat, Marsujitullah Marsujitullah, Hariyanto Hariyanto,	Sistem *Alerting*	Sistem *alerting* yang terhubung dengan sensor api dan Arduino UNO. Intensitas pembacaan pada sensor api apabila melebihi

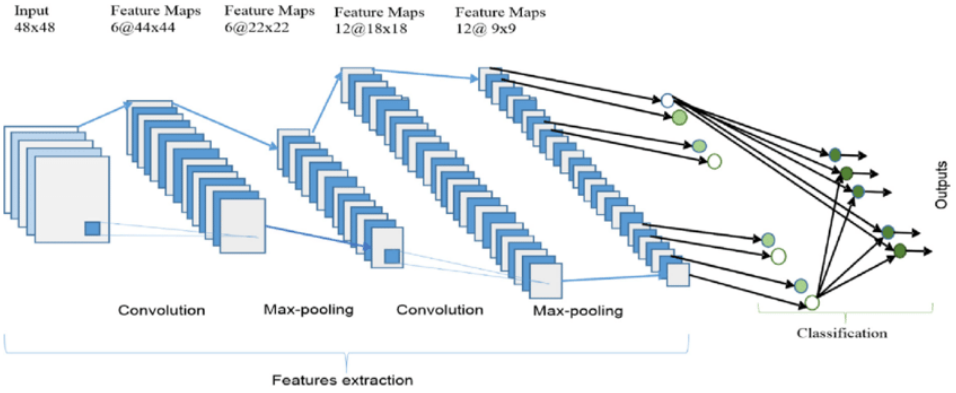
	Wahyu Caesarendra, Stanislaw Legutko, and Adam Glowacz. 2021. "Design of a Home Fire Detection System Using Arduino and SMS Gateway" Knowledge 1, no. 1: 61-74. https://doi.org/10.3390/knowledge1010007. [14]		batasan tertentu maka *buzzer* akan berbunyi dan mengirimkan pesan peringatan melalui omodul GSM.

Keterbaharuan yang terdapat pada penelitian ini terletak pada pengimplementasian algoritma YoloV4 dalam mendeteksi api ke dalam perangkat sistem tertanam yang terintegrasi IoT.

2.2 Convolutional Neural Network (CNN)

Deep Learning adalah ilmu di bidang *Machine Learning* yang berkembang karena perkembangan GPU teknologi akselerasi yang memiliki kemampuan luar biasa dalam *computer vision*. Salah satu kemampuannya ada di kasus klasifikasi objek dalam gambar. Untuk menerapkan Deep Learning yang dapat digunakan untuk objek gambar klasifikasi, yaitu CNN. Cara kerja CNN mirip dengan *Multi Layer Perceptron* (MLP), tetapi di CNN, setiap neuron disajikan dalam dua dimensi. MLP menerima input data pada satu dimensi dan meneruskan informasi ke keluaran *Neural Network*. Setiap tautan antara neuron dalam dua lapisan yang berdekatan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Pada setiap lapisan data input, operasi linier dilakukan dengan nilai bobot yang ada, kemudian hasil komputasi tersebut ditransformasikan menggunakan operasi non-linier yang disebut fungsi aktivasi. Di CNN, data yang disebarkan pada jaringan adalah dua dimensi, sehingga operasi linier dan parameter bobot pada CNN berbeda. Di CNN,

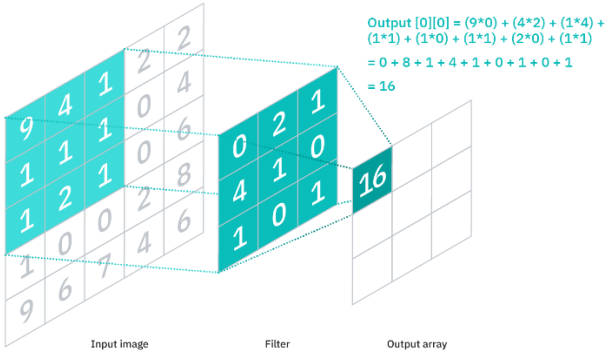
operasi linier menggunakan operasi konvolusi, sedangkan beratnya tidak lagi satu dimensi tetapi empat dimensi yang merupakan kumpulan dari konvolusi kernel. Sebuah CNN terdiri dari beberapa jenis layer, yaitu *Convolutional Layer*, *Max Pooling Layer*, dan *Fully Connected Layer* [15].



Gambar 2.1 Arsitektur Convolutional Neural Network

1. *Convolutional Layer*

Lapisan konvolusi adalah blok bangunan utama CNN. Ini berisi satu set filter (atau kernel). Ukuran filter biasanya lebih kecil dari gambar sebenarnya. Setiap filter menyatu dengan gambar dan membuat peta aktivasi. Untuk konvolusi, filter meluncur melintasi tinggi dan lebar gambar dan *dot product* antara setiap elemen filter dan input dihitung pada setiap posisi spasial [16].

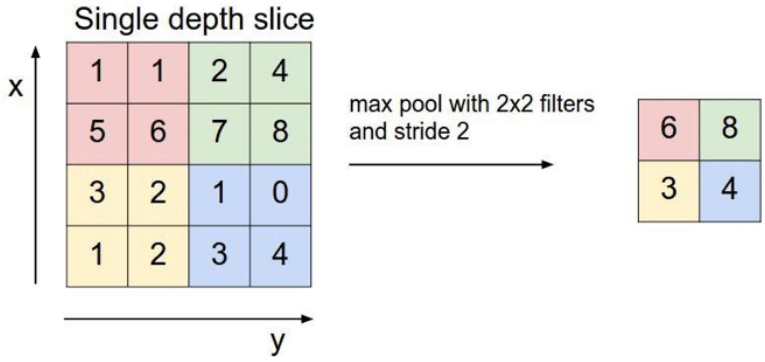


Gambar 2.2 Lapisan Konvolusi

2. *Max Pooling Layer*

Fungsi dari pooling ini adalah untuk mereduksi input secara spasial (mengurangi jumlah parameter) dengan operasi down-sampling. Umumnya, metode pooling yang digunakan adalah max

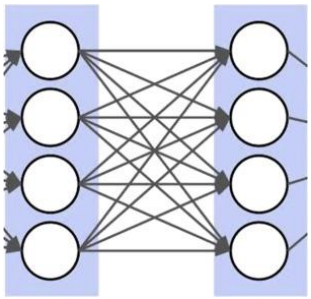
pooling atau mengambil nilai terbesar dari bagian tersebut. Namun terdapat metode pooling lain yang dapat digunakan seperti average pooling atau L2-norm pooling.



Gambar 2.3 Lapisan *Max Pooling*

3. *Fully Connected Layer*

Neuron yang telah melewati lapisan *Convolution* dan *Max Pooling Average* diteruskan ke lapisan *Flatten* dimana setiap bagian diubah menjadi vektor sehingga data dapat diklasifikasikan secara linear. Output diubah menjadi 1 x 1 dimensi dimana jumlahnya didapat dari perkalian antara lebar (width), tinggi (height) dan channel ditambah jumlah bias. Proses ini akan berlanjut pada fungsi aktivasi yang akan menghitung probabilitas dari setiap kelas yang memungkinkan menentukan class dari input citra yang diberikan.



Gambar 2.4 Lapisan *Fully Connected*

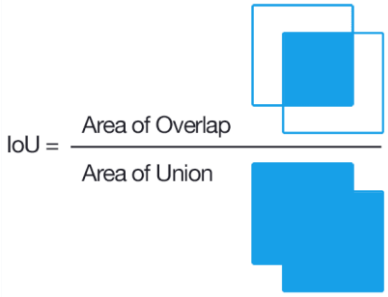
2.3 YOLOv4

YOLO merupakan abreviasi dari *You Only Look Once* adalah algoritma yang digunakan untuk objek deteksi karena ukurannya yang kecil

dan kecepatan pada kemampuan deteksinya. YOLO menggunakan *Convolution Neural Network* (CNN) pada *single forward propagation*

1. *Intersection over Union (IoU)*

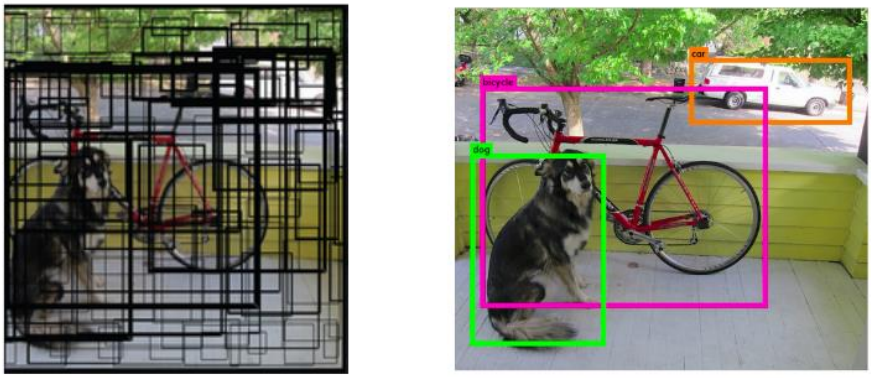
IoU, juga dikenal sebagai indeks Jaccard, adalah yang metrik yang paling umum untuk membandingkan kesamaan antara dua bentuk arbitrer. IoU mengkodekan properti bentuk dari objek yang dibandingkan, yakni lebar, tinggi, dan lokasi dua *bounding boxes*, ke dalam properti wilayah dan kemudian menghitung ukuran yang dinormalisasi yang berfokus pada area gambar[17].



Gambar 2.5 Visualisasi *Intersection of Union* (IoU)

Nilai *confidence* didapat dari perhitungan IoU. Pada Computer Vision digunakan untuk mendeteksi objek dengan benar. Saat pelatihan memprediksi objek dengan *bounding box*, maka *bounding box* yang diprediksi dibandingkan dengan bounding box dari *ground-truth.*

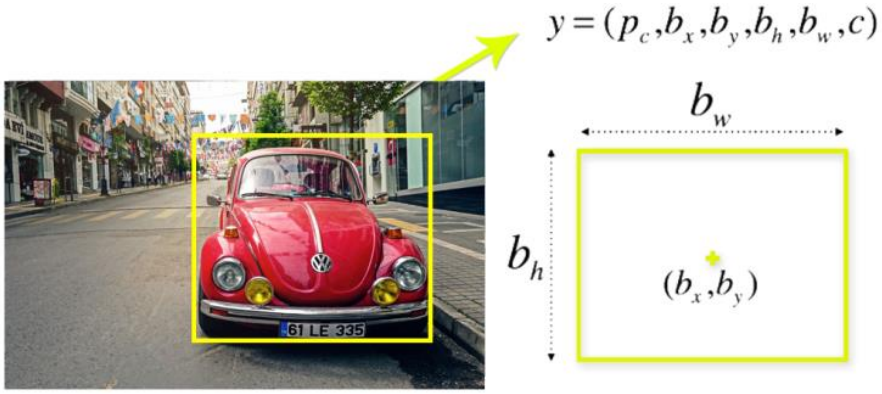
Pada proses deteksi objek dalam gambar, *bounding boxes* seringkali terbentuk. Algoritma *Non-max suppression* (NMS) digunakan untuk mengeliminasi *bounding boxes* terbaik dari gambar. NMS merupakan algoritma yang bertugas untuk mencari local maxima berdasarkan nilai batasan IoU awal model [18] Apabila nilai batasan awal IoU adalah 0.5, maka proses eliminasi pada *bounding boxes* yang memiliki nilai dibawah 0.5 akan di eliminasi, kemudian setelah dilakukan proses eliminasi pertama akan dialnjutkan dengan proses eliminasi kedua dengan mengambil *bounding boxes* dengan *confidence score* terbesar dari *bounding boxes.*



Gambar 2.6 *Non-max Suppression* (NMS)

2. *Object Localization*

Model lokalisasi objek mirip dengan model klasifikasi. Tetapi model lokalisasi yang terlatih juga memprediksi lokasi objek dalam gambar dengan menggambar kotak pembatas di sekitarnya. Sebagai contoh, sebuah mobil terletak pada gambar di bawah ini. Informasi kotak pembatas, koordinat titik pusat, lebar dan, tinggi juga disertakan dalam keluaran model.



Gambar 2.7 Object Localization dan *Classification*

Output dari lokalisasi model akan mencakup informasi dari bounding boxes, sehingga *output* akan terlihat pada vektor dibawah ini:

p_c: *Confidence score* dari objek pada *bounding boxes*

$$y = \begin{bmatrix} \rho_c \\ t_x \\ t_y \\ t_h \\ t_w \\ c_1 \\ \cdots \\ c_n \end{bmatrix}$$

b_x: Pusat koordinat x terhadap bagian kiri *bounding boxes*

b_y: Pusat koordinat y terhadap bagian atas *bounding boxes*

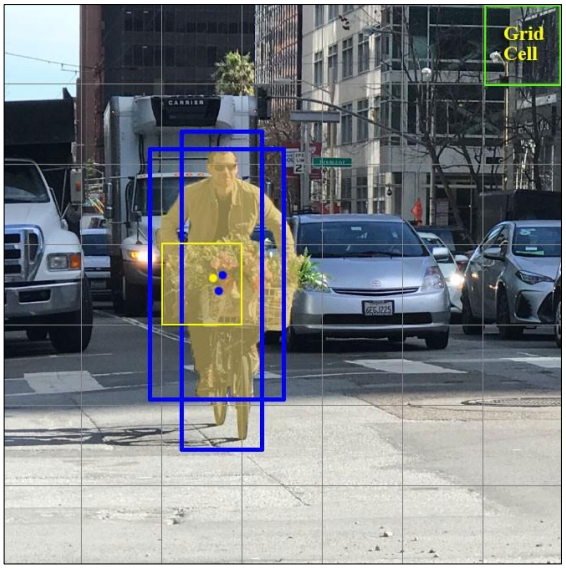
b_h: Tinggi *bounding boxes*

b_w: Lebar *bounding boxes*

c_n : Bernilai satu apabila terdapat kelas-n yang terdeteksi, nol apabila tidak terdapat kelas-n yang terdeteksi\

3. Grid Cells

YOLO membagi masukan gambar ke dalam *S×S grid.* Apabila pusat dari objek terletak pada *grid cell* maka *grid cell* trsebut bertanggung jawab dalam mendeteksi objek tersebut. Setiap *grid cell* melakukan prediksi terhadap nilai *B Bounding Boxes* dan menghitung *Confidence Score* dari masing-masing *grid cells.*



Gambar 2.8 *Residual Block* pada sampel gambar

Confidene score merefleksikan tingkat akurasi dan kepercayaan dari model dalam memprediksi objek di dalam *Boxes* [19]. Apabila tidak terdeteksi objek pada *cell* maka nilai *Confidence Score* bernilai nol, namun apabila tidak maka nilai *Confidence Score* akan berbanding lurus dengan *bounding boxes* terhadap *ground*

truth (IoU). Persamaan dari nilai *Confidence Score* dapat dihitung dengan persamaan:

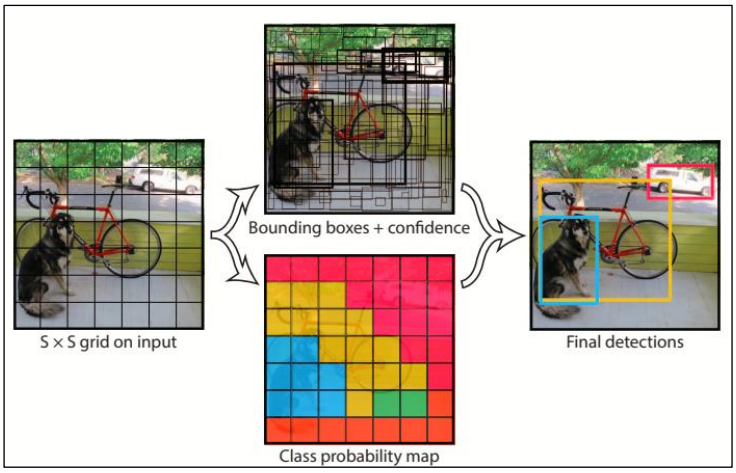
$$Box\ Confidence\ Score = P_r(Object) \times IoU$$

$P_r(Object)$: Probabilitas dari *bounding boxes* terdapat objek
IoU : *Intersection of Union* (IoU)

Setiap *grid cells* memprediksi nilai kondisional dari probabilitas masing-masing kelas. *Grid cell* menghasilkan prediksi kemungkinan kelas terbesar yang kemudian digunakan untuk memprediksi tingkat akurasi kelas berdasarkan nilai IoU nya.

$$Class\ Probability = P_r(Class_i|Object)$$
$$Class\ Confidence\ Score = P_r(Class_i) \times IoU$$

$P_r(Class_i|Object)$: Probabilitas suatu objek berada di $Class_i$ ketika suatu objek terdeteksi
IoU : *Intersection of Union* (IoU)
$P_r(Class_i)$: Probabitas objek itu berada di $Class_i$



Gambar 2.9 Tahapan deteksi menggunakan Algoritma YOLO

4. *Data Augmentation*

Data augmentasi merupakan metode yang digunakan untuk meningkatkan masukan input data dengan membuat data replika yang telah di manipulasi untuk mengatasi permasalahan ketidakseimbangan masukan data pada kelas, meningkatkan generalisasi dan mencegah terjadinya *overfitting* [20] YOLOv4 menggunakan beberapa teknik dari data augmentasi yang disebut sebagai "the bag of freebies" karena kemampuannya yang mampu meningkatkan akurasi tanpa memberatkan proses inferensi, adapun augmentasi data yang digunakan pada YOLOV4 yakni *Photometric Distortion, Image Occlusion* dan *Geometric Distortion* [21]

Photometric Distortion mencakup augmentasi gambar dengan memanipulasi tingkat saturasi, kecerahan, kontras pada gambar. *Geometric Distortion* mencakup *cropping, flipping* dan *rotating. Image Occlusion* juga digunakan sebagai legularisasi dalam menghindari terjadinya *overfitting* dengan menggunakan *random erase, cut out, hide and seek, CutMix, grid mask* dan *MixUp*.

5. *Recall, Precission* dan *mAP*

Recall merupakan perbandingan atau proporsi antara kasus *real positive* terhadap *predicted positive. Predicted positive* merupakan penjumlahan antara *true positive* dan *false negative*.

$$Recall = \frac{\sum True\ Positive}{\sum True\ Positive + False\ Negative}$$

Precission merupakan perbandingan atau proporsi antara kasus *predicted positive* yang merupakan *real positive. Predicted positive* merupakan penjumlahan antara *true positive* dan *false negative* [22].

$$Precission = \frac{\sum True\ Positive}{\sum True\ Positive + False\ Positive}$$

Average precission (AP) merupakan area yang terbentuk antara *recall* dan *precission. Average precission* (AP) merupakan standard ukuran yang digunakan untuk mengukur performansi dari objek deteksi. Nilai dari AP digdapatkan berdasarkan nilai awal IoU. Persamaan yang digunakan dalam mencari AP adalah persamaan yang digunakan untuk mencari area antara grafik *Recall-Precission* yakni :

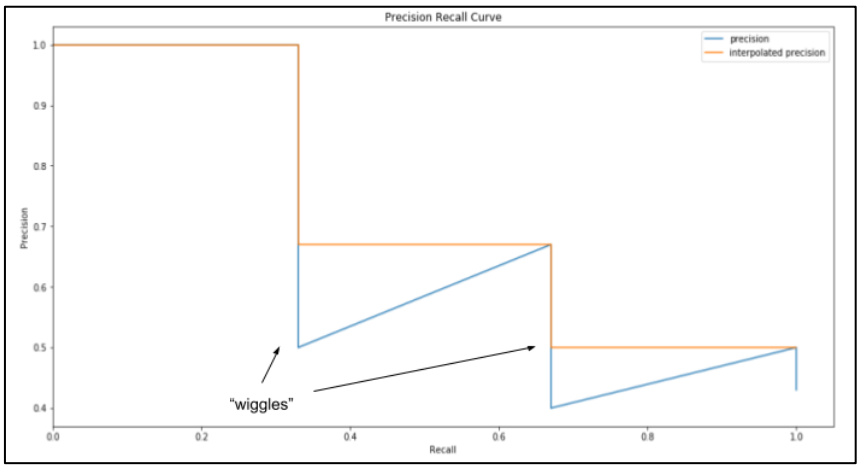
$$AP = \int_0^1 p(r)\, dr$$

$p(r)$: Fungsi *precission* terhadap *recall*
r : Nilai *recall*

Proses interpolasi dilakukan terhadap nilai PR untuk memnimalisir variasi yang ditimbulkan akibat urutan sampel. Interpolasi AP dilakukan dengan mengganti nilai *recall* $\hat{r}$ dengan maksimum *precission* pada *recall* $recall \geq \tilde{r}$[23]

$$p_{interpolasi} = \max_{\tilde{r} \geq r} p(\tilde{r})$$

$p_{interpolasi}$: *Precission* yang telah terinterpolasi
$\tilde{r}$: Nilai *recall* dengan *precission* terbesar



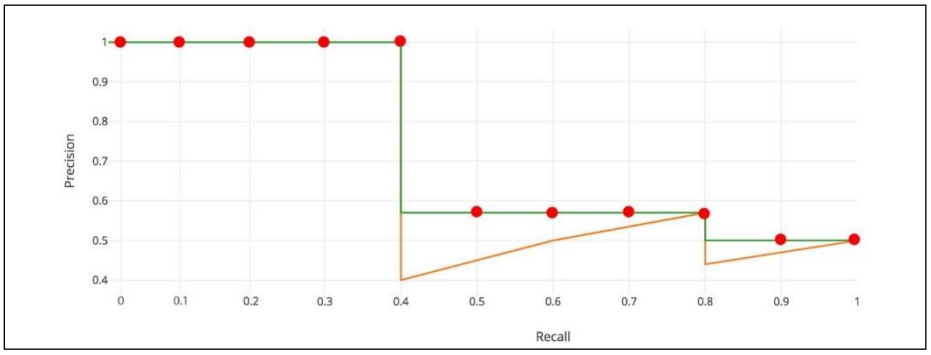
Gambar 2.10 Nilai interpolasi AP berdasarkan nilai maksimum antara *Recall-Precission*

Nilai *recall* di bagi ke dalam 11 titik dari interval [0, 1] terhadap nilai AP yang telah dilakukan interpolasi untuk dihitung nilai rata-rata. Nilai rata-rata dari nilai ke-n *recall values* setelah

dilakukan interpolasi terhadap nilai maximum *precission* dapat dirumuskan ke dalam persamaan berikut [24] :

$$mAP = \frac{1}{N}\sum_{1}^{N} p_{interpolasi}(r)$$

N : Nilai *recall* pada interval [0, 1]
mAP : *mean Average-Precission*
$p_{interpolasi}(r)$: Nilai *precission* yang telah ter-interpolasi



Gambar 2.11 mean Average-Precission (mAP)

2.4 Raspberry Pi

Raspberry Pi, sering juga disingkat dengan nama Raspi, adalah komputer papan tunggal yang memiliki ukuran sebesar kartu kredit. Raspberry Pi bisa digunakan untuk berbagai keperluan, seperti spreadsheet, game, bahkan bisa digunakan sebagai media player karena kemampuannya dalam memutar video high definition. Raspberry Pi dikembangkan oleh yayasan nirlaba, Rasberry PiFoundation yang digawangi sejumlah developer dan ahli komputer dari Universitas Cambridge, Inggris.

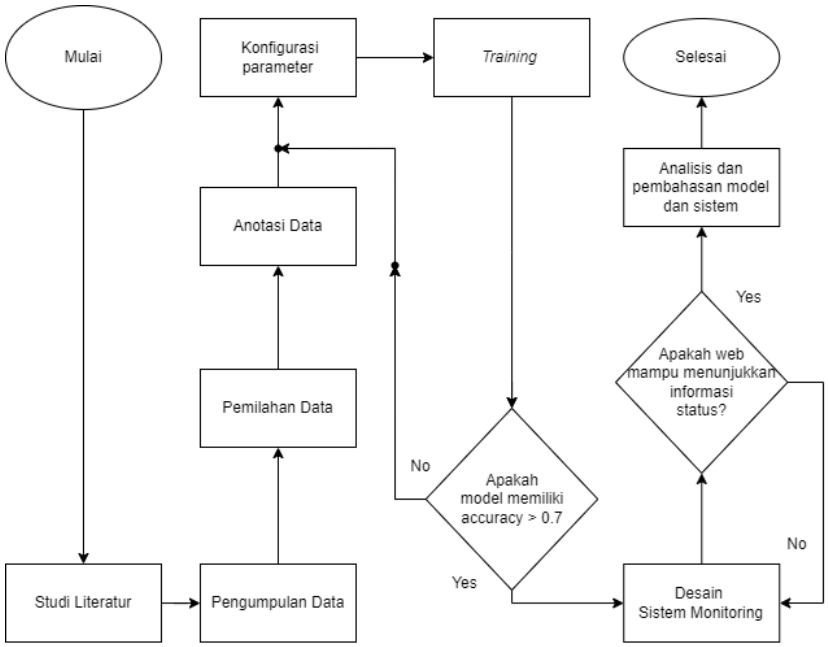
Raspberry Pi memiliki dua model yaitu model A dan model B. Secara umum Raspberry Pi Model B, 512MB RAM. Perbedaan model A dan B terletak pada memori yang digunakan, Model A menggunakan memori 256 MB dan model B 512 MB. Selain itu model B juga sudah dilengkapai dengan ethernet port (kartu jaringan) yang tidak terdapat di model A. Desain Raspberry Pi didasarkan seputar *SoC (System-on-a-chip)* Broadcom BCM2835, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, VideoCore IV GPU, dan 256 MegabyteRAM (model B).

Penyimpanan data didisain tidak untuk menggunakan hard disk atau solidstate drive, melainkan mengandalkan kartu SD (SD memory card) untuk booting dan penyimpanan jangka panjang.

BAB 3
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian ini dibagi ke dalam beberapa bagian besar yakni *pre-research* tahapan sebelum penelitian di lakukan dengan melakukan studi literatur terkait penelitian serupa yang memiliki keterkaitan topik yang sedang dibahas. Tahapan selanjutnya adalah *pre-processing* data, yakni tahapan pengolahan data sebelum di *training*, tujuannya untuk meningkatkan akurasi dari prediksi pada proses interferensi. Setelah fase *pre-processing* data dilanjutkan ke fase *Training* dengan menyesuaikan dan mengkonfigurasikan parameter yang digunakan. Fase berikutnya dilanjutkan dengan evaluasi model untuk mengevaluasi model terhadap kriteria yang di tetapkan, apabila model telah memenuhi kriteria yang ditentukan dilakukan analisis dan pembahasan terkait kinerja model pada system deteksi api.



Gambar 3.1 Diagram Alir Penelitian

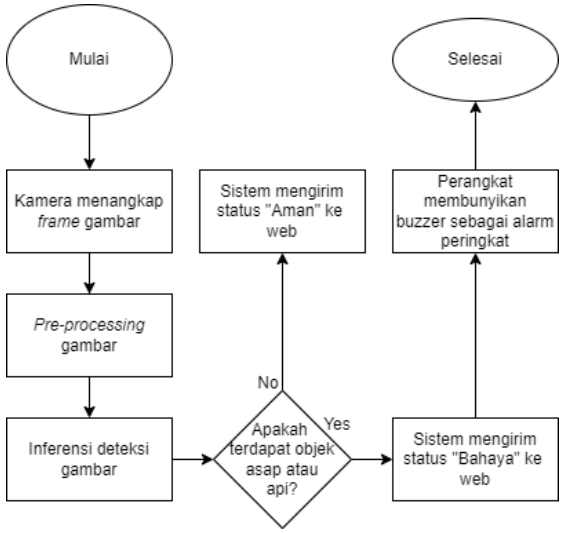
Penelitian di awali dengan melakukn studi literatur terkait proses *pre-processing* gambar dengan menerapkan morfologi, sebagai proses filter

dan juga untuk memperkuat fitur-fitur pada gambar. Gambar tersebut kemudian di anotasi sebagai bahan *training* dengan mengkonfigurasi parameter seperti besar *epochs* dan *batch* pada *training.* Informasi prediksi gambar meliputi kelas, skor prediksi dan kondisi akan dikirimkn dan di visualisasikan ke dalam bentuk web. Analisis kinerja dari web dan sistem deteksi di lakukan sebagai optimalisasi dari kinerja sistem deteksi api.

3.2 Implementasi

1. Alur Kerja

Proses kerja dari alat di awali dengan perangkat mengambil gambar *frame* per *frame* gambar. *Frame* gambar tersebut di lakukan *pre-processing* dengan menerapkan morfologi sebelum di inferensikan untuk mendapatkan prediksi kelas dan besar skor prediksi. Skor prediksi dan prediksi kelas dari gambar dikirimkan ke *website* untuk di tampilkan kepada pengguna.



Gambar 3.2 Diagram Alur Kerja Alat

2. Perangkat

Adapun perangkat keras yang digunakan pada fase *Training* dan interferensi :

1. *Training*

Tabel 3.1 Spesifikasi Perangkat *Training*

GPU	:	Nvidia RTX3090Ti
Processor	:	Intel i7 9750H

RAM	:	12 GB
Penyimpanan	:	SSD *PCIe* 1 TB

2. Interferensi

Tabel 3.2 Spesifikasi Perangkat Interferensi

Model	:	Raspberry Pi 4
Processor	:	Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
RAM	:	4 GB
Penyimpanan	:	32 GB SD Card

3.3 Implementasi

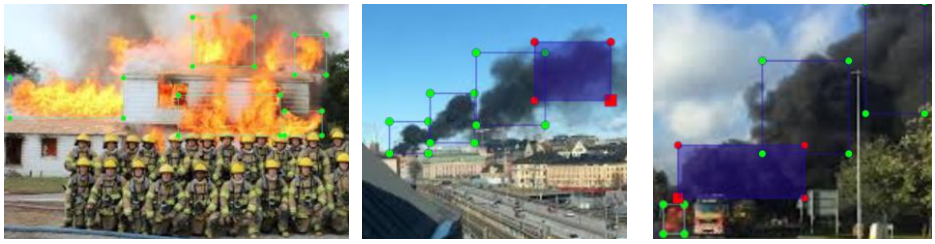
Tabel 3.3 Jadwal Penelitian

Waktu / Kegiatan	Bulan Ke 1				Bulan Ke 2				Bulan Ke 3				Bulan Ke 4				Bulan Ke 5			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah																				
Studi Literatur																				
Prancangan Harware dan Sofware																				
Pengujian Alat																				
Pengumpulan Data																				
Hasil dan Pembahasan																				
Kesimpulan																				

BAB 4
HASIL DAN PEMBAHASAN

4.1 Pengambilan Data

Data yang digunakan untuk model objek lokalisasi YOLOv4 pada sistem deteksi kebakaran harus mencakup berbagai jenis kebakaran dengan berbagai tingkat intensitas. Data ini harus diperoleh dari berbagai sumber seperti video kebakaran yang diperolah dari CCTV, rekaman video simulasi kebakaran dan gambar kebakaran.



Gambar 4.1 Pembagian Dataset Api and Asap

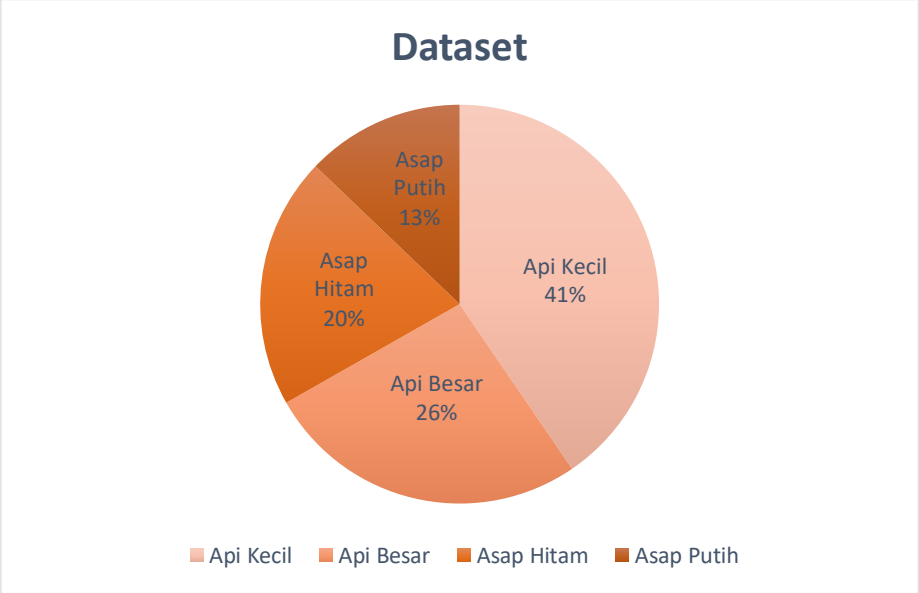
Untuk mendapatkan data yang representatif, pengambilan data harus dilakukan di berbagai situasi dan kondisi yang berbeda, data yang di ambil mencakup data api dan asap dengan ukuran daan intensitas yang berbeda. Data yang beragam ini akan membantu model untuk menangani berbagai kondisi yang berbeda dan menghindari terjadinya *overfitting* [25].

Filterisasi data pada dataset kebakaran sangat penting untuk menjamin kualitas dari model objek lokalisasi yang akan dibangun. Proses ini meliputi pemilihan data yang sesuai dengan kriteria yang ditentukan dan penghapusan data yang tidak sesuai atau tidak valid.kebakaran serta label yang sesuai dengan objek tersebut, dalam hal ini menghAnnotasi yang baik akan membantu model untuk belajar dan mengenali objek kebakaran dengan baik.

Beberapa kriteria yang dapat digunakan dalam filterisasi data adalah ukuran bounding box, label objek, dan kualitas gambar. Ukuran bounding box yang terlalu kecil atau terlalu besar dapat dihapus karena tidak representatif. Label objek yang tidak sesuai dengan objek yang akan dideteksi juga harus dihapus. Kualitas gambar yang buruk atau tidak jelas juga harus dihapus untuk menjamin kualitas data yang digunakan.

Selain itu, data yang digunakan harus juga diperiksa secara manual untuk memastikan bahwa semua data yang digunakan sesuai dengan kriteria yang ditentukan dan tidak ada data yang tidak valid. Dengan filterisasi data yang baik, model objek lokalisasi yang akan dibangun akan lebih baik dan akurat dalam mengenali objek kebakaran.

Pengambilan data dilakukan dengan menggabungkan dataset api dan kebakaran, dengan total dataset mencapai 996 gambar setelah dilakukan filterisasi. Data tersebut terdiri atas 403 gambar api berukuran kecil dan 262 gambar api berukuran besar, pada dataset asap terdiri atas 203 gambar asap berwarna hitam dan 128 gambar asap berwarna putih.



Gambar 4.2 Pembagian Dataset

4.2 Training

Proses training di bagi ke dalam dua model Deep Learning yakni YOLOv4 dan YOLOv4-tiny, dengan membagi proses training ke dalam tiga dataset berbeda, yakni dataset yang berisikan asap, dataset yang berisikan api dan dataset yang berisikan keduanya, pembagian training ini bertujuan untuk mengetahui model mana yang memiliki akurasi, precission dan recall terbesar serta kecepatan inferensi terbaik dari ketiga model.

Untuk mengukur mAP, recall, dan precision dari model YOLOv4, dilakukan dengan membandingkan hasil deteksi dari model dengan ground truth atau data asli yang digunakan sebagai acuan. Kemudian, menghitung Intersection over Union (IoU) antara hasil deteksi dengan ground truth, untuk dapat mengetahui seberapa baik model dalam menemukan objek yang sebenarnya ada. Kemudian, dapat menghitung *mAP, recall, precision* dan F1-*Score* dengan menggunakan rumus yang telah ditentukan.

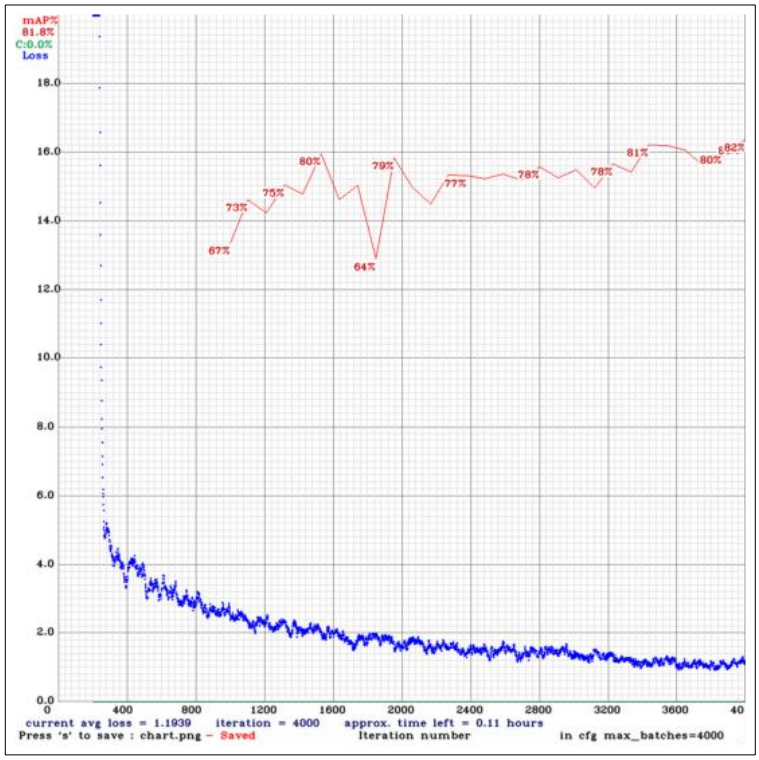
Model dari *deep learning* memiliki kemampuan yang tinggi apabila memanfaatkan graphics processing units (GPU) , namun keterbatasan kemampuan proses deteksi pada sistem tertanam dalam memanfaatkan GPU membuat hal ini sulit untuk dicapai. Framework tensorflow digunakan untuk meningkatkan performa dengan meningkatkan kemampuan latensi dari inferensi, mengurangi konsumsi daya dan mengurangi ukuran dari model dengan melakukan optimalisasi kinerja pada hardware [26]. Latensi dengan akurasi terbaik dari ketiga model akan dipilih untuk di implementastikan. Besar latensi dan ukuran dari model akan memengaruhi kecepatan inferensi. Berdasarkan parameter tersebut hasil inferensi di dapatkan sebagai berikut :

Tabel 4.1 Hasil Training

Parameter	Api	Asap	Api dan Asap
mAP@0.50	**0.83**	0.61	0.81
Precission	0.78	0.73	**0.83**
Recall	**0.81**	0.52	0.79
F1-Score	0.80	0.69	**0.81**
Average	0.80	0.64	**0.81**

Berdasarakan hasil *training* dengan memperlakukan pembedaan dataset, di dapatkan nilai parameter pada model yang di latih menggunakan menggunakan dataset asap memiliki nilai hasil parameter lebih kecil ketimbang model yang di latih dengan menggunakan dataset api, hal ini dikarenakan keberagaman bentuk dan warna dari asap. Berdasarkan data tersebut, model dari YOLOv4 menggunakan dataset gabungan api dan asap di pilih karena nilai parameter terbaik dari model lainnya, kemampuannya

dalam melakukan generalisasi, serta waktu inferensi yang singkat, membuat model ini di pilih sebagai model yang di gunakan pada implementasi.



Gambar 4.3 Grafik Loss dan mAP dari YOLOv4 dataset Api dan Asap

Verifikasi dilakukan dengan melakukan deteksi gambar api di dalam berbagai kondisi, pengujian pertama di lakukan dengan menguji kemampuan deteksi di tempat yang berbeda, di dalam ruangan dan di ruang terbuka. Pengujian ini dilakukan untuk memastikan bahwa sistem yang dikembangkan mampu mendeteksi api dan asap di berbagai ruangan.

Gambar 4.4 Pengujian deteksi api di dalam ruangan

Gambar 4.5 Pengujian deteksi api di luar ruangan

Pengujian kedua di lakukan dengan menguji kemampuan deteksi pada ukuran yang berbeda. Pengujian ini di bagi ke dalam dua bagian yakni pengujian dengan menggunakan gambar api berukuran besar dan pengujian menggunakan gambar api yang berukuran kecil. Pengujian ini dilakukan untuk memastikan bahwa sistem yang dikembangkan mampu mendeteksi api dan asap di berbagai ukuran.

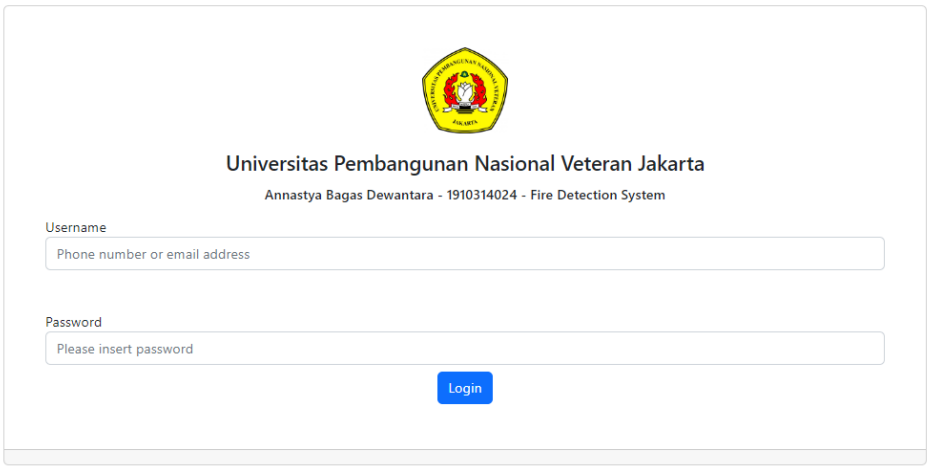
Gambar 4.6 Pengujian deteksi dengan gambar api berukuran besar

Gambar 4.7 Pengujian deteksi dengan gambar api berukuran kecil

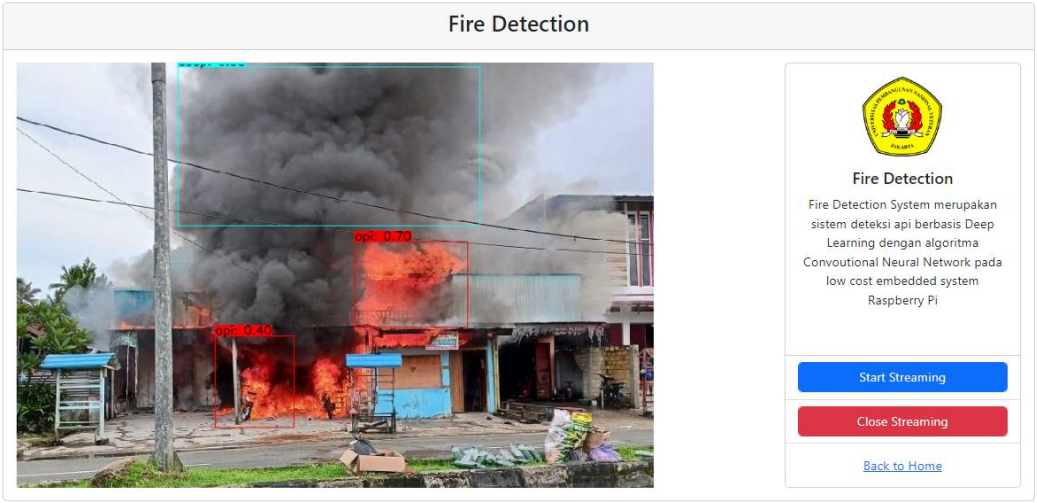
4.3 Tampilan Web

Proses inferensi akan melakukan deteksi objek dari setiap frame yang tertangkap pada webstreaming yang berisikan hasil bounding boxes, skor prediksi, dan kelas dari objek. Hasil dari proses deteksi ini akan di tampilkan dalam bentuk visualiasi berbasis web yang akan digunakan sebagai media monitoring oleh pengguna. Sistem tertanam akan terhubung dengan piezzo buzzer yang berfungsi sebagai indikator apabila terdeteksi api yang berpotensi terjadi kebakaran.

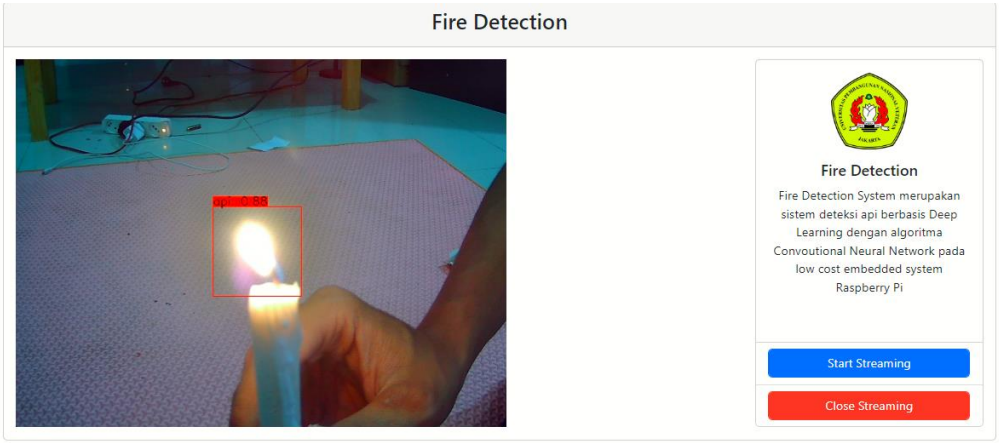
Tampilan web terdiri atas tiga halaman yakni, halaman *login*, halaman utama dan halaman *webstrean.* Halaman *login* berfungsi untuk melakukan autentikasi dan otorisasi terhadap pengguna yang memiliki akses atas web. Halaman utama adalah halaman yang berisikan deskripsi singkat terkait sistem deteksi api dan halaman yang memiliki fitur tombol yang akan merujuk ke halaman *webstream*, Halaman webstream merupakan halaman yang bertujuan untuk menunjukkan hasil inferensi *real-time* dari kamera.



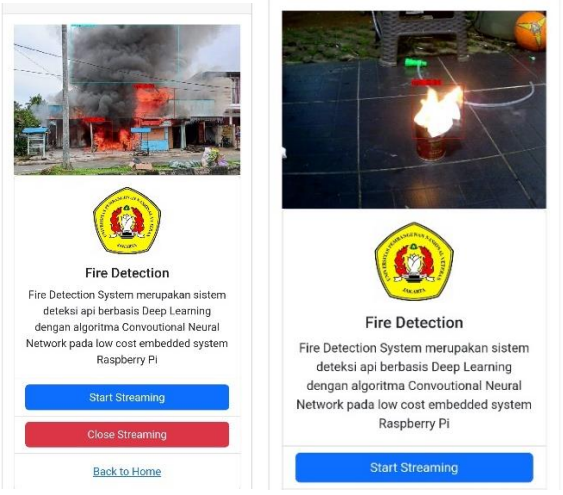
Gambar 4.8 Tampilan halaman *login* dari web



Gambar 4.9 Tampilan halaman utama dari web



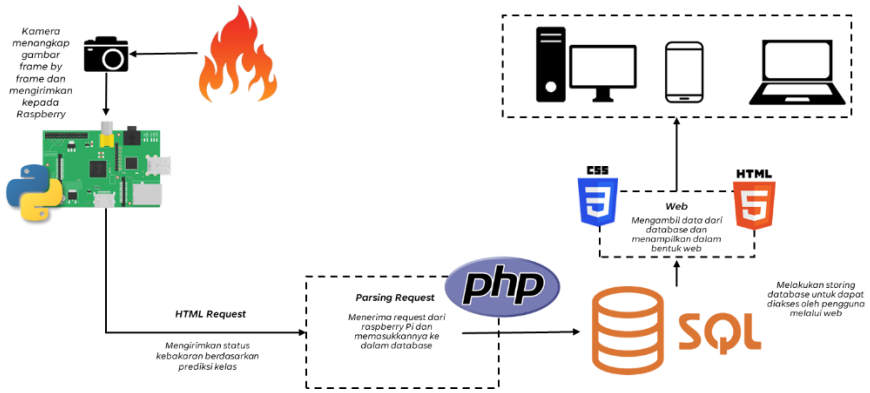
Gambar 4.10 Tampilan halaman *streaming* dari web



Gambar 4.11 Tampilan *mobile site*

4.4 Perangkat

Pengujian dari purwarupa dilakukan dengan menggunakan kamera Logitech C270 yang terhubung dengan Raspberry Pi 4. Kamera ini digunakan untuk merekam gambar dan video dari objek yang akan dideteksi, yang kemudian akan diolah oleh model YOLOv4 untuk mendeteksi kebakaran. Raspberry Pi 4 digunakan sebagai pengolah data yang diterima dari kamera dan menjalankan model YOLOv4.



Gambar 4.12 Desain sistem dan alur kerja dari alat

Selain itu, Raspberry Pi 4 juga terhubung dengan indikator lampu dan piezzo sebagai indikator apabila terjadi kebakaran. Lampu akan menyala dan piezzo akan berbunyi sebagai sinyal bahwa sistem telah mendeteksi kebakaran. Dengan cara ini, pengujian purwarupa dapat dilakukan dengan efektif dan akurat.

Gambar 4.13 Tampilan purwarupa sistem deteksi api

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang di lakukan melalui proses *training* dari dataset yang telah di kumpulkan, dapat di ketahui bahwa hasil deteksi sistem deteksi api pada sistem tertanam menggunakan alogritma YOLOv4 sebagai berikut:

1. Desain monitoring dari sistem deteksi api berhasil di lakukan dengan menerapkan model YOLOv4 pada webserver yang di jalankan pada sistem tertenam Raspberry Pi 4.
2. Desan sirkuit dari sistem deteksi api berhasil di lakukan dengan menghubungkan indikator lampu LED dan *piezzo buzzer* sebagai indikator suara yang berfungsi sebagai sistem alerting pada saat gambar terdeteksi api.
3. Model yang dilatih memiliki kemampuan generalisasi mampu mendeteksi api berukuran besar dan berukuran kecil, dan mampu memprediksi api di dalam ruangan dan di luar ruangan.
4. Model yang di latih menggunakan dataset api dan asap, memiliki nilai hasil mAP@0.50 sebesar 0.81, *precission* sebesar 0.83, *recall* 0.79 *dan F1-Score* sebesar 0.81, yang memenuhi parameter kriteria awal yang ditentukan.

5.2 Saran

Setelah dilakukan penelitian, di dapatkan evaluasi yang dapat di lakukan sebagai bahan pengembangan lebih lanjut, untuk memperoleh sistem yang lebih efektif dan interaktif:

1. Proses pengumpulan data perlu dilakukan kembali untuk mendapatkan variasi data lebih besar, sehingga model mampu melakukan generalisasi dalam mendeteksi terjadinya kebakaran.
2. Membuat tampilan website yang lebih *user friendly* dan interaktif, yang memiliki fitur yang mampu melakukan deteksi dari gambar atau video yang di *upload* oleh pengguna.