

Rancang Bangun Quadcopter Drone Untuk Deteksi Api Menggunakan YOLOv4

Made Radikia Prasanta¹, M. Yoga Pranata², M. Aditya Firnanda³, Siti Sendari⁴

^{1,2,3,4}Jurusan Teknik Elektro, Universitas Negeri Malang

Jl. Semarang No. 5, Kec. Lowokwaru, 65145

email: made.radikia.1805366@students.um.ac.id

Abstrak— Kebakaran hutan dan lahan berdampak pada rusaknya ekosistem. Menurut data dari Kementerian Lingkungan Hidup dan Kehutanan (KLHK), luas hutan Indonesia per tahun 2013 mencapai 128,5 juta hektar. Ironisnya, hutan di Indonesia setiap tahunnya terus menyusut diakibatkan oleh kebakaran hutan. Hal ini dapat dicegah dengan memanfaatkan UAV (Unmanned Aerial Vehicle) untuk melakukan pengamatan secara langsung melalui kamera. Penelitian ini dilakukan untuk mengembangkan prototype quadcopter drone yang dilengkapi dengan sistem deteksi api berbasis YOLOv4. Algoritma YOLOv4 adalah algoritma deep learning yang menerapkan jaringan syaraf konvolusional (CNN). YOLOv4 yang diskalakan adalah jaringan neural yang memiliki akurasi tertinggi dan memiliki kecepatan terhadap akurasi diseluruh rentang akurasi dan kecepatan. Metode penelitian yang dilakukan diawali dengan perancangan hardware quadcopter drone, perancangan sistem pengiriman data citra video, kemudian percancangan sistem deteksi api, dilakukan pengumpulan data citra api, pra processing data, rancangan konfigurasi arsitektur YOLO, training data dan pengujian. Hasil penelitian menunjukkan quadcopter drone yang terintegrasi dengan groundstation menggunakan algoritma YOLOv4 berhasil mendeksi api dengan menghasilkan rata-rata akurasi sebesar 0.92. Akurasi terbaik didapat pada pengujian ke 3 dengan skor precision sebesar 0.96, skor recall sebesar 0.98 dan skor accuracy sebesar 0.96.

Kata kunci: Api, YOLOv4, Quadcopter.

Abstract— Forest and land fires have an impact on ecosystem damage. According to data from the Ministry of Environment and Forestry (KLHK), Indonesia's forest area per year 2013 reached 128.5 million hectares. Ironically, forests in Indonesia continue to shrink every year due to forest fires. This can be prevented by using a UAV (Unmanned Aerial Vehicle) to make observations directly through the camera. This research was conducted to develop a quadcopter drone prototype equipped with a YOLOv4-based fire detection system. The YOLOv4 algorithm is a deep learning algorithm that implements a convolutional neural network (CNN). scalable YOLOv4 is the most accurate neural network and has speed to accuracy across the entire accuracy and speed range. The research method begins with designing quadcopter drone hardware, designing a video image data transmission system, then designing a fire detection system, collecting fire image data, pre-processing data, YOLO architectural configuration design, training data and testing. The results showed that the quadcopter drone integrated with the groundstation using the YOLOv4 algorithm succeeded in detecting fire by producing an average accuracy of 0.92. The best accuracy is obtained on the 3rd test with a precision score of 0.96, a recall score of 0.98 and an accuracy score of 0.96.

Keywords: Fire, YOLOv4, Quadcopter.

I. PENDAHULUAN

Hutan merupakan salah satu pendukung yang sangat penting bagi keseimbangan alam. Berbagai macam kebutuhan manusia sangat bergantung pada hutan, hasil hutan digunakan sebagai bahan baku aneka produk kebutuhan manusia, hutan menjadi sumber paling besar penghasil oksigen dan berperan penting dalam menyerap karbon dioksida, hutan juga secara langsung menjadi rumah bagi flora dan fauna di seluruh dunia. Ironisnya luas hutan Indonesia setiap tahunnya terus menyusut merujuk pada perhitungan Ditjen Planologi KLHK, angka deforestasi Indonesia periode 2014-2015 mencapai 1,09 juta hektar dan 2015-2016 menjadi 0,63 juta hektar [1]. Penyebab hilangnya hutan, salah satunya kebakaran hutan.

Kebakaran hutan dan lahan berdampak pada rusaknya ekosistem dan menyebabkan musnahnya flora dan fauna

yang tumbuh dan hidup di hutan. Dampak lainnya dari asap yang ditimbulkan dapat menyebabkan berbagai macam penyakit. Menurut data dari Kementerian Lingkungan Hidup dan Kehutanan, luas hutan Indonesia per tahun 2013 mencapai 128,5 juta hektar [2]. Ironisnya, hutan di Indonesia setiap tahunnya terus menyusut diakibatkan oleh kebakaran hutan. Merujuk pada perhitungan Ditjen Planologi KLHK, angka deforestasi Indonesia periode 2014-2015 mencapai 1,09 juta hektar dan 2015-2016 menjadi 0,63 juta hektar [3].

Salah satu solusi yang dapat dilakukan untuk mencegah terjadinya kebakaran hutan yaitu pemantauan kawasan hutan yang harus dilakukan secara berkala. Seperti penggunaan kamera yang dipasang pada batang pohon untuk mencakup luasan tertentu di hutan dalam mendeksi dini kebakaran hutan, pendekripsi dilakukan secara realtime berbasis notifikasi menggunakan *raspberry* [4].

Selain itu pemantauan dapat dilakukan dengan memanfaatkan teknologi UAV (*Unmanned Aerial Vehicle*) atau pesawat tanpa awak yaitu pengamatan secara langsung melalui kamera yang ada di pesawat [5]. Dalam hal ini kamera pada pesawat bertujuan untuk mengambil foto atau video kondisi lingkungan atau hutan sekitar melalui udara, kamera terintegrasi dengan program komputer yang selanjutnya akan diproses untuk menentukan apakah terdapat api atau tidak.

Metode yang dapat dilakukan dalam mendeksi api secara otomatis adalah dengan menggunakan pengolahan citra [6]. Dalam penggunaan teknik ini objek api dengan berbagai macam kondisi dapat secara mudah untuk dideteksi oleh sistem yaitu dengan menggunakan framework You Only Look Once (YOLO). YOLO adalah sebuah algoritma yang dikembangkan untuk mendeksi sebuah objek secara realtime. Berbagai macam versi YOLO salah satunya yaitu YOLOv4. Algoritma YOLOv4 adalah algoritma deep learning yang menerapkan jaringan saraf konvolusional (CNN). YOLOv4 yang diskalakan memiliki keakuratan yang tertinggi (55,8% AP) dengan kumpulan data dari semua jaringan neural. YOLOv4 juga merupakan yang terbaik dalam hal kecepatan terhadap akurasi, dengan rentang kecepatan mulai dari 15 FPS hingga 1774 FPS.

II. STUDI PUSTAKA

A. YOLO

Yolo merupakan algoritma pendekatan jaringan saraf tiruan yang digunakan untuk mendeksi objek dalam sebuah citra secara realtime. Algoritma Yolo merupakan pengembangan lanjut dari algoritma *Convolutional Neural Network* (CNN). Metode Yolo memungkinkan proses pengenalan objek lebih cepat jika dibandingkan dengan metode *region-based convolutional neural networks* (R-CNN) [7]. Hal ini karena algoritma Yolo mengenali pola objek dalam sekali lihat tanpa proses klasifikasi ulang.

Prinsip kerja Yolo melibatkan pelatihan jaringan saraf tiruan secara end-to-end menggunakan citra input untuk memprediksi bounding box dan pengelompokan object class setiap bounding box.

B. Quadcopter Drone

Quadcopter merupakan pesawat multirotor/multiwings yang memiliki motor sejumlah 4 buah dengan gerakan yang lebih leluasa dibandingkan dengan helicopter dengan hanya 2 buah rotor [8]. Terdapat empat gerakan dasar pada quadcopter yaitu gerakan altitude (*throttle*), gerakan sudut (*roll, pitch*), dan gerakan sudut yaw. Gerakan throttle merupakan gerak translasi quadcopter sepanjang sumbu z, Gerakan ini dipengaruhi oleh perubahan kecepatan keempat rotor dengan nilai kecepatan putaran yang sama. Dengan pergerakan throttle, quadcopter akan dapat melakukan akselerasi naik atau turun sesuai dengan tingkat kecepatan rotor yang dihasilkan.

C. Dataset Api

Data set merupakan sekumpulan model yang digunakan sebagai bahan pembelajaran mesin [9]. Pada penelitian ini komposisi dataset yang digunakan terdiri dari berbagai jenis dan bentuk citra api dengan format jpg. Jumlah data set yang digunakan berjumlah 3000 gambar.

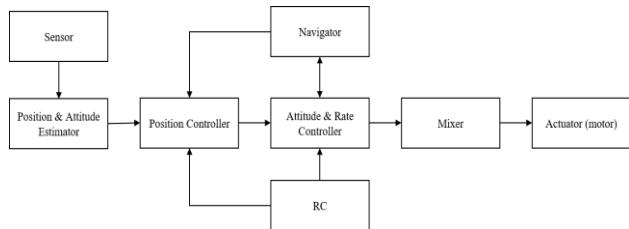
Data yang terkupul kemudian masuk ke proses Anotation. Proses Anotation adalah proses pemberian Bounding Box pada objek yang teridentifikasi sebagai api. Dataset ini kemudian akan ditraining sehingga menghasilkan data latih baru yang dapat mengenali objek api.

III. METODE

Metode pelaksanaan dari penelitian ini dapat digambarkan melalui tahapan *flowchart* sebagai berikut:

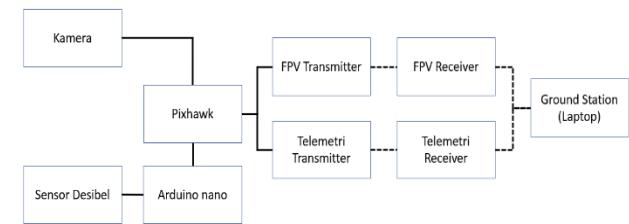
A. Rancangan Hardware Quadcopter

Flight Stack tersusun dari kumpulan algoritma, pedoman, navigasi, dan kontrol untuk drone otonom seperti kontrol untuk lengan sayap, multirotor dan badan pesawat VTOL dan sebagai estimator untuk attitude dan posisi. Blok diagram berikut menunjukkan cara kerja lapisan Flight Stack. Estimator berfungsi untuk membaca satu atau beberapa input lalu menggabungkannya dan menghitung state kenaikan, misalnya menghitung attitude dari sensor IMU. Sedangkan kontroler pada lapisan Flight Stack berfungsi untuk membaca setpoint dan state keadaan sebagai nilai input, dengan tujuan untuk menyesuaikan nilai variabel proses agar mendekati nilai setpoint. Mixer adalah lapisan yang berfungsi untuk menerjemahkan perintah dari kontroler supaya dapat menggerakkan aktuator. Middleware Lapisan Middleware tersusun dari perangkat driver embended sensor, perangkat komunikasi eksternal (seperti PC, CGS) dan bus untuk menyalurkan informasi antar segment. Middleware mencakup lapisan simulasi yang memungkinkan perangkat drone PX4 dijalankan pada sistem operasi desktop.



Gambar 1. Rancangan Sistem Hardware

B. Rancangan Sistem Pengiriman Data Quadcopter



Gambar 2. Rancangan Sistem Pengiriman Data

Penangkapan gambar dilakukan oleh kamera yang terhubung langsung dengan pixhawk dan FPV transmitter. Data citra yang ditangkap akan langsung dikirimkan menggunakan FPV transmitter dan ditangkap oleh ground station melalui FPV receiver. Data yang diterima oleh groundstation akan langsung diolah untuk mendeksi api menggunakan pemrograman bahasa python.

C. Rancangan Sistem Deteksi Api

Alur kerja sistem diawali dengan pengumpulan data (akuisisi data) data dikumpulkan baik dari internet maupun hasil ekstraksi frame video perekaman api. Selanjutnya setiap gambar atau hasil ekstraksi akan mengalami perbaikan dengan mengubah contrast stretching pada tahapan pra processing data dengan melakukan pelabelan. Konfigurasi jaringan YOLO dilakukan untuk persiapan melakukan pelatihan data, kemudian citra yang sudah melalui pelabelan akan ditraining menggunakan model jaringan YOLOv4. Hasil training yang telah dilakukan yang dapat digunakan untuk mendeteksi api



Gambar 3. Alur Rancangan Sistem Deteksi Api

D. Pengumpulan Data

Citra api yang digunakan dalam penelitian dikumpulkan, dijadikan sebagai dataset. Dataset api atau berupa berbagai macam gambar api seperti kebakaran hutan, lilin, kebakaran rumah dikumpulkan hingga berjumlah 2000 gambar. Gambar dikumpulkan dengan format jpg.



Gambar 4. Dataset Api

E. Pra Processing Data

Pada tahap ini data citra akan mengalami perbaikan kualitas citra dengan menggunakan metode contrast stretching atau peregangan kontras. Contrast stretching merupakan teknik untuk mendapatkan citra dengan kontras yang lebih baik dari sebelumnya. Proses contrast stretching dilakukan bergantung pada graylevel satu pixel citra dan tidak bergantung pada piksel lain atau bersifat point operations [6]. Point operations dilakukan pada domain

spasial, yakni operasi pada sumbu x dan y pada bidang kartesian, atau jika dalam citra disebut baris dan kolom [7].

Proses ini juga disebut dengan pelabelan, pelabelan dan perubahan ukuran citra. Pada tahap ini data citra akan mengalami perbaikan kualitas citra. Tahapan pertama setiap gambar akan diberi label untuk menyimpan informasi citra. Selanjutnya citra akan dilakukan perubahan ukuran untuk meningkatkan performa model YOLO.

F. Rancangan Arsitektur YOLO

Konfigurasi layer Yolo digunakan sebagai jaringan konvolusi untuk melakukan proses training dan normalisasi dataset. Arsitektur konfigurasi jaringan Yolo terbentuk dari layer konvolusi dengan ukuran kernel 3 x 3 dan 1x1 sebagai ekstraksi fitur. Serta lapisan max polling dengan kernel ukuran 2 x 2. Pada bagian akhir, output dataset dikecilkan menjadi 26 x 26 x 18, dimana 26 x 26 adalah ukuran grid dan 18 didapat dari penjumlahan filter.

Total keseluruhan layer konvolusi yang digunakan pada training ini berjumlah 36 layer. Kemudian di bagian Filter di tiap [convolutional] layer sebelum [yolo] layer diatur dengan nilai 18 yang diperoleh dari rumus:

$$\text{Filter} = (\text{jumlah kelas dataset} + 5) \times 3 \quad (2)$$

```

conv 32 3 x 3 / 2 416 x 416 x 3 -> 208 x 208 x 32 0.075 BF
  1 conv 64 3 x 3 / 2 208 x 208 x 32 -> 104 x 104 x 64 0.399 BF
  2 conv 64 3 x 3 / 1 104 x 104 x 64 -> 104 x 104 x 64 0.797 BF
  3 route 2
  4 conv 32 3 x 3 / 1 104 x 104 x 32 -> 104 x 104 x 32 0.199 BF
  5 conv 32 3 x 3 / 1 104 x 104 x 32 -> 104 x 104 x 32 0.199 BF
  6 route 5 4
  7 conv 64 1 x 1 / 1 104 x 104 x 64 -> 104 x 104 x 64 0.089 BF
  8 route 2 7
  9 max 2x 2 / 2 104 x 104 x 128 -> 52 x 52 x 128 0.001 BF
  10 conv 128 3 x 3 / 1 52 x 52 x 128 -> 52 x 52 x 128 0.797 BF
  11 route 10
  12 conv 64 3 x 3 / 1 52 x 52 x 64 -> 52 x 52 x 64 0.199 BF
  13 conv 64 3 x 3 / 1 52 x 52 x 64 -> 52 x 52 x 64 0.199 BF
  14 route 13 12
  15 conv 128 1 x 1 / 1 52 x 52 x 128 -> 52 x 52 x 128 0.089 BF
  16 route 10 15
  17 max 2x 2 / 2 52 x 52 x 256 -> 26 x 26 x 256 0.001 BF
  18 conv 256 3 x 3 / 1 26 x 26 x 256 -> 26 x 26 x 256 0.797 BF
  19 route 18
  1/2 -> 26 x 26 x 128
  
```

Gambar 5. Struktur Layer Konvolusi YOLO

G. Training Data

Training data dilakukan menggunakan Google Colaboratory atau Google Colab. Google colab merupakan layanan cloud gratis yang disediakan oleh Google dalam bentuk *executable document* yang dapat digunakan untuk menulis, mengedit, menyimpan serta membagikan program yang telah ditulis melalui Google Drive. Proses training data set dilakukan dengan bantuan Google Colab, karena google colab menyediakan GPU dengan kapasitas 12 GB dengan support N-Vidia. Sehingga memungkinkan proses training yang lebih cepat pada perangkat yang memiliki spesifikasi yang kurang mendukung. Training data dilakukan berkali kali.

```

Learning Rate: 0.00261, Momentum: 0.9, Decay: 0.0005
Detection layer: 38 - type: 28
Create 6 parallel cpu-threads
Loaded 6 parallel threads
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (ID: 0.391839), count: 7, class_loss = 137.592941, iou_loss = 0.253693,
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (ID: 0.308378), count: 1, class_loss = 448.911285, iou_loss = 0.417877,
total_bbox = 8, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 38 Avg (ID: 0.228093), count: 3, class_loss = 136.327438, iou_loss = 0.048477,
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (ID: 0.451495), count: 1, class_loss = 458.877380, iou_loss = 0.418669,
total_bbox = 12, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 38 Avg (ID: 0.286406), count: 6, class_loss = 133.689879, iou_loss = 0.092816,
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (ID: 0.186695), count: 1, class_loss = 447.552924, iou_loss = 0.199482,
total_bbox = 19, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 38 Avg (ID: 0.244618), count: 4, class_loss = 135.219797, iou_loss = 0.050879,
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (ID: 0.242484), count: 6, class_loss = 446.332788, iou_loss = 0.373077,
total_bbox = 29, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 38 Avg (ID: 0.292565), count: 3, class_loss = 133.640844, iou_loss = 0.018997,
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (ID: 0.339722), count: 7, class_loss = 445.855773, iou_loss = 0.397766,
total_bbox = 36, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 38 Avg (ID: 0.479061), count: 4, class_loss = 132.077138, iou_loss = 0.327377,
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (ID: 0.212658), count: 1, class_loss = 448.293242, iou_loss = 0.039429,
total_bbox = 15, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 38 Avg (ID: 0.419497), count: 4, class_loss = 134.598479, iou_loss = 0.164807,
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (ID: 0.358955), count: 2, class_loss = 445.199048, iou_loss = 0.099454,
total_bbox = 50, rewritten_bbox = 0.000000 %
  
```

Gambar 6. Proses Training Data

H. Pengujian

Pada tahapan pengujian adalah menguji data hasil training dapat mendekripsi citra api. Jika citra api dapat terdeteksi dengan adanya bounding box, maka proses training berhasil. Pada proses pengujian spesifikasi PC yang digunakan yaitu seperti pada tabel 1.

Tabel 1. Spesifikasi PC yang Digunakan

Processor	AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz
RAM	16 GB DDR4-3200 SDRAM
Storage	SSD 512 GB PCIe® NVMe™ M.2
Graphic Card	NVIDIA® GeForce® GTX 1650 4GB
Operating System	Windows 10

Pada tahap pengujian dilakukan analisis akurasi menggunakan confusion matrix. Pengujian dilakukan untuk mengetahui sensitifitas dan spesifisitas. Pengujian sensitifitas adalah metode perhitungan akurasi dengan membandingkan jumlah klasifikasi tepat pada suatu class dengan keseluruhan klasifikasi yang terdapat pada class tersebut. Sedangkan pengujian spesifisitas adalah metode perhitungan dengan membandingkan jumlah klasifikasi yang tidak berhubungan dengan sebuah class namun dianggap tepat dengan seluruh klasifikasi yang tidak berhubungan dengan class tersebut. Dalam perhitungan selanjutnya akan digunakan *matrix confusion* dalam perhitungan dari beberapa class.

Berdasarkan tabel *confusion matrix*, apabila terdeteksi api dan sistem dapat mendekripsi benar api maka akan dikelompokan menjadi True Positive (TP) karena hasil klasifikasi sesuai seperti yang diharapkan. Pengelompokan ini didasari oleh hasil deteksi yang menunjukkan benar adanya api (true) dan hasil klasifikasi sesuai dengan yang seharusnya (positive).

Apabila objek bukan api namun sistem mengidentifikasi sebagai api, maka akan dikelompokan menjadi False Positive (FP), karena hasil deteksi tidak seperti yang diharapkan. Pengelompokan ini berdasarkan pada objek bukan api (False) namun teridentifikasi sebagai api (Positive). Ketika objek api dideteksi sebagai bukan api, maka akan dikelompokan menjadi False Negative (FN). Pengelompokan tersebut didasari dari adanya objek api namun sistem mendeteksi bukan api. Serta apabila objek bukan api terdeteksi sebagai bukan api, maka akan dikelompokan ke True Negative (TN). Untuk mendapatkan hasil akurasi maka dari akumulasi tersebut akan dihitung menggunakan rumus sebagai berikut:

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

Untuk menghitung sensitivitas digunakan rumus:

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

Untuk menghitung spesifisitas digunakan rumus:

$$Spesifitas = \frac{TN}{TN+FP} \quad (3)$$

I. Implementasi Desain Sistem

Tahap implementasi dari pengembangan yang telah dilakukan yaitu dengan menggunakan kamera webcam yang

terhubung melalui kabel USB yang akan menangkap citra secara langsung dan terhubung ke PC untuk diproses.

IV. HASIL DAN PEMBAHASAN

Hasil dari penelitian dan pengembangan ini berupa Purwarupa Drone untuk deteksi api berbasis pengolahan citra dan deteksi penebang liar menggunakan sensor suara. Terdapat 3 hasil dalam penelitian dan pengembangan ini. Hasil Pertama yaitu purwarupa quadcopter drone, sistem deteksi api berbasis pengolahan citra, dan sistem deteksi penebang liar melalui suara gergaji mesin.

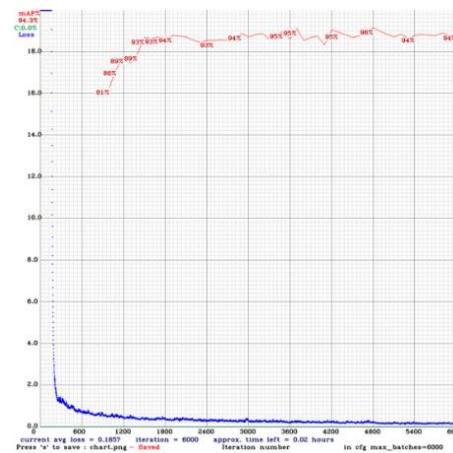
A. Quadcopter Drone

Dalam penelitian ini, purwarupa quadcopter drone berfungsi sebagai alat utama yang akan terbang untuk merekam citra gambar untuk mendeteksi api dan mendeteksi suara gergaji mesin menggunakan sensor suara desibel. Drone terdiri dari beberapa komponen utama seperti brushless motor sebagai motor pengangkat drone, FPV transmitter untuk mengirimkan video ke ground station, telemetri transmitter untuk mengirimkan data ke ground station, ESC sebagai pengatur kecepatan brushless motor, kamera untuk menangkap citra, sensor suara desibel untuk mendeteksi suara gergaji mesin dan pixhawk sebagai flight controller drone. Power supply drone menggunakan baterai dengan kapasitas 5400 mAh yang membuat drone dapat terbang hingga 20 menit. Hasil rancangan quadcopter drone seperti pada gambar 7.



Gambar 7. Hasil Rancangan Quadcopter Drone

B. Hasil Training



Gambar 8. Grafik Hasil Training

Proses training dilakukan sebanyak 6000 batch atau iterasi dan menghasilkan data model baru yang dapat mengenali cita api. Berdasarkan training yang dilakukan didapat mAP (*mean Average Precision*) yaitu 94% dan avg

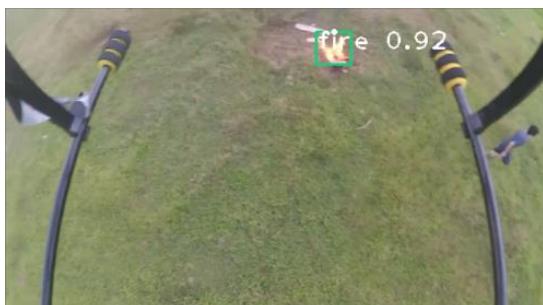
loss 0.1657 dengan waktu training selama 2.4 jam. Hasil dari proses training dalam bentuk grafik dapat dilihat pada gambar 6.

C. Pengujian Deteksi Api Secara Real Time

Pengujian secara real time melalui Action Cam yang terpasang pada badan drone. Action Cam terhubung dengan perangkat FPV transmisi yang dikirim ke FPV receiver untuk kemudian diteruskan ke ground station (PC). Citra yang diperoleh dari Kamera kemudian langsung diolah pada ground station dengan menggunakan program aplikasi Python. Pengujian dilakukan di area lapangan hijau. Kemudian terdapat tong api sebagai sumber titik api yang akan dideteksi.



Gambar 9. Hasil Deteksi Api Dengan Ketinggian 3 meter



Gambar 10. Hasil Deteksi Api Ketinggian 5 meter



Gambar 11. Hasil Deteksi Api Ketinggian 8 meter

Pengujian yang telah dilakukan seperti pada gambar 9, 10, 11 terbukti sistem mampu mendeksi api secara realtime dengan pemberian bounding box jika citra yang ditangkap kamera positif teridentifikasi sebagai api. Sistem mampu mendeksi titik api hingga ketinggian 8 meter.

Pengujian ini dilakukan sebanyak empat kali percobaan. Sehingga diperoleh empat video yang memiliki durasi dan posisi citra api yang berbeda-beda. Keempat percobaan tersebut kemudian dianalisis dengan menggunakan rumus *Confusion Matrix* sehingga diperoleh nilai akurasi, presisi dan recall sebagai berikut :

Tabel 2. Akurasi deteksi api dalam 4 kali percobaan

Video	Durasi	TP	TN	FP	FN	Recall	Precision	Accuracy
1	105	42	55	7	1	0.98	0.86	0.92
2	95	37	12	5	0	1.00	0.88	0.91
3	100	64	32	3	1	0.98	0.96	0.96
4	210	76	118	14	2	0.97	0.84	0.92

Berdasarkan hasil perhitungan pada tabel 2 sistem dapat mengenali objek api dengan akurasi 0.91 hingga tertinggi mencapai 0.96 dengan rata-rata sebesar 0.92. Sistem dapat mendeksi posisi dan objek api dengan tepat. Sistem juga dapat membedakan mana objek yang bukan api dan objek yang termasuk api dengan baik. Hal ini dibuktikan dengan nilai FP dan FN yang sangat kecil. Semakin kecil nilai *False* maka persentase akurasinya semakin tinggi.

Sementara nilai TP merupakan citra api yang dapat dideksi oleh sistem dan TN merupakan citra bukan api yang dikenali sistem. Semakin tinggi nilai *True* maka persentase akurasi, presisi dan recall semakin baik. Pada hasil pengujian diperoleh nilai presisi tertinggi yaitu 0.96 pada video 3, persentasi recall yang sangat baik yaitu 1.00 pada video 2, dan akurasi yang tinggi sebesar 0.96 pada video 3.

V. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan, purwarupa quadcopter drone dapat dibuat dan sukses dalam menangkap citra gambar. Pengolahan citra dirancang menggunakan YOLOv4, dapat mendeksi api dengan baik dengan training sebanyak 6000 epoch mendapatkan akurasi terbaik yaitu sebesar 0.96. Serta berdasarkan pengujian dan analisis menggunakan *confusion matrix* didapat nilai akurasi terbaik yaitu pada pengujian ke 3 sebesar 0.96 dengan nilai recall yaitu 0.98 dan nilai presisi sebesar 0.96.

REFERENSI

- [1] BBC, Kebakaran hutan dan lahan kian meluas dan kabut asap semakin parah, BNPB kewalahan padamkan api, Available: <https://www.bbc.com/indonesia/indonesia-49708970>, 2019.
- [2] S. Andy, Faktor-Faktor Yang Mempengaruhi Kebakaran Hutan Di Indonesia dan Implikasi Kebijakannya, Universitas Gadjah Mada, Indonesia, 2015.
- [3] F. Rasyid, Permasalahan dan Dampak Kebakaran Hutan, Kementerian Lingkungan Hidup dan Kehutanan, 2014.
- [4] S.Y. Pradana, F. Utaminingrum, W. Kurniawan, Deteksi Titik Api Terpusat Menggunakan Kamera Dengan Notifikasi Berbasis Sms Gateway Pada Raspberry Pi, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 2, No. 12, 2018.
- [5] P. Perkasa, Y. Aguswan, Use of Drone for Early Detection Means Land and Forest Fire, Jurnal Pendidikan Teknologi dan Kejuruan BALANGA, 2018.
- [6] K. Firdausy, Y. Saudi, T. Sutikno, Deteksi Api Real-Time Dengan Metode Thresholding Rerata RGB, Jurnal TELKOMNIKA. Vol. 5, No. 2, 2007.
- [7] O. E. Karlina and D. Indarti, "Pengenalan Objek Makanan Cepat Saji Pada Video Dan Real Time Webcam Menggunakan Metode You Look Only Once (Yolo)," *J. Ilm. Inform. Komput.*, vol. 24, no. 3, pp. 199–208, 2019, doi: 10.35760/ik2019.v24i3.2362, 2019.
- [8] Arifin, Fatchul dkk, Rancang Bangun Quadcopter dilengkapi dengan Automatic Navigation GPS Control dan Camera Stabilizer sebagai alat bantu Monitoring Lalu Lintas dengan Live Streaming System. Fakultas Teknik, Universitas Negeri Yogyakarta, 2015.
- [9] O. Masurekar, O. Jadha, P. Kulkarni, S. Patil, Real Time Object Detection Using YOLOv3, International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 03, 2020