

## Deteksi *Landing Pad* pada *Unmanned Aerial Vehicle* (UAV) Menggunakan YOLO

**Ifa Nirmala Laili**

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya  
e-mail: ifa.20034@mhs.unesa.ac.id

**Parama Diptya Widayaka, I Gusti Putu Asto Buditjahjanto, Lilik Anifah**

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya  
e-mail: paramawidayaka@unesa.ac.id, asto@unesa.ac.id, lilikanifah@unesa.ac.id

### Abstrak

Teknologi robot terbang di Indonesia telah mengalami kemajuan yang signifikan, terbukti dengan penyelenggaraan Kontes Robot Terbang Indonesia (KRTI) setiap tahun. Salah satu kategori yang dilombakan adalah VTOL (*Vertical Take-off and Landing*) dengan salah satu misinya pendaratan secara otonom yang mengharuskan *multicopter* mampu mendeteksi lokasi *landing* dengan tepat. Oleh karena itu, diperlukan pengembangan sistem *computer vision* untuk mendeteksi objek *landing pad* dengan keakuratan yang tinggi, sehingga *multicopter* mampu melakukan misi selanjutnya dengan baik. Penelitian ini menggunakan sebuah UAV *Multicopter* yang dilengkapi dengan *webcam* Logitech C920 sebagai *platform computer vision*. Metode deteksi yang diterapkan adalah YOLO Darknet, sebuah pendekatan yang menggunakan *deep learning*. Model YOLO yang digunakan adalah YOLOv3 Tiny, dan proses pelatihan melibatkan *framework* Darknet. Selama tahap pelatihan, didapatkan nilai *average loss* sebesar 0,0609 dan *mean Average Precision* (mAP) sebesar 99,4%. Selama pengujian jaringan, sistem berhasil mendeteksi objek dengan label "landing pad" dan menampilkan *bounding box* serta nilai *confidence* pada *frame* gambar yang dihasilkan. Pada pengujian menggunakan *validation set* menggunakan 200 gambar sebagai *test set* didapatkan nilai performa jaringan dengan rata-rata nilai *accuracy* sebesar 0.95, *precision* 0.98, *recall* 0.96, dan *F1-score* sebesar 0.96.

Kata Kunci: UAV, *multicopter*, *computer vision*, deteksi objek, YOLO, darknet

### Abstract

The flying robot technology in Indonesia is developing quite rapidly, evidenced by the Indonesian Flying Robot Contest (KRTI) held annually. One of the categories competed is VTOL (Vertical Take-off and Landing) with one of its missions being autonomous landing, requiring multicopters to accurately detect the landing location. Therefore, the development of a computer vision system to detect landing pad objects with high accuracy is necessary, so that multicopters can perform subsequent missions effectively. This research utilizes a UAV Multicopter equipped with a Logitech C920 webcam as a computer vision platform. The detection system used is YOLO Darknet, a method of deep-learning. The YOLO network used is YOLOv3 Tiny, and the training process involves the Darknet framework. During the network training phase, an average loss value of 0.0609 and a mean Average Precision (mAP) of 99.4% were obtained. During network testing, the system successfully detected objects labeled as "landing pad" and displayed bounding boxes as well as confidence values on the resulting image frames. In testing using a validation set with 200 images as a test set, the network performance values were obtained with an average accuracy of 0.95, precision of 0.98, recall of 0.96, and F1-score of 0.96.

Keywords: UAV, multicopter, computer vision, object detection, YOLO, darknet

### PENDAHULUAN

Teknologi robot terbang di Indonesia telah mengalami perkembangan yang signifikan karena adanya dorongan dan dukungan dari pemerintah (Ashari & Achmadi, 2021). Bukti nyata dari perkembangan ini adalah diselenggarakannya Kontes Robot Terbang Indonesia (KRTI) setiap tahunnya (Abbiyansyah dkk, 2022). Salah satu kategori yang diperlombakan adalah

*Vertical Take-off and Landing* (VTOL). Berdasarkan peraturan, KRTI mensyaratkan agar sebuah wahana *multicopter* dapat terbang secara otonom menuju objek yang telah ditentukan, dalam hal ini wahana dapat menggunakan baik GPS (*Global Positioning System*) maupun *vision* untuk navigasi (Firdaus dkk, 2023).

Misi divisi VTOL pada KRTI 2023, yaitu *multicopter* mampu mengambil dan menjatuhkan *object dropping* pada lokasi yang telah ditentukan. Misi ini dilaksanakan dengan cara menyelesaikan misi *indoor* terlebih dahulu kemudian melanjutkan misi *outdoor* dan *landing pada* posisi HOME di lapangan sendiri dalam sekali perintah (Nuryadi, et al., 2023). Pendaratan otonom pada VTOL merupakan masalah yang kompleks karena apabila terjadi sedikit kesalahan pembacaan antara posisi *landing pad* yang sebenarnya dan posisi *landing pad* yang diestimasi oleh sistem navigasi, dapat menyebabkan kegagalan saat proses *landing* (Medeiros, et al., 2018).

Sistem yang telah diterapkan saat ini pada prototipe *multicopter* untuk mendeteksi *landing pad* yaitu menggunakan pendeteksian warna *Hue*, *Saturation*, *Value* (HSV) untuk mendapatkan kontur warna dari objek dengan menggunakan Mini PC Intel NUC I3. Metode HSV mengartikan warna menggunakan istilah *Hue* (warna sebenarnya), *Saturation* (kemurnian warna), dan *Value* (kecerahan warna) (Fauzi dkk, 2018). Kekurangan dari metode ini yaitu sensitivitas terhadap variasi cahaya yang kurang stabil ketika terjadi perubahan pencahayaan yang signifikan yang mengharuskan *vision* melakukan pengambilan data setiap saat untuk mendapatkan kontur warna yang sesuai (Ikhsan & Akbar, 2020), ditambah dengan adanya keterbatasan prosesor yang digunakan mengakibatkan sering terjadi *delay* saat memproses kontur.

Oleh karena itu, diperlukan pengembangan sistem *computer vision* untuk mendeteksi objek yang dalam hal ini adalah *landing pad* dalam citra digital yang ditangkap oleh kamera *multicopter*, dengan tujuan mengurangi kesalahan antara posisi pusat *landing pad* dan posisi *multicopter* serta untuk efisiensi waktu (Nugroho & Anifah, 2023).

Pendeteksian objek atau *object detection* merupakan salah satu penerapan visi komputer untuk mengidentifikasi objek tertentu (Nugroho, 2023). Tujuan dari visi komputer adalah untuk mengekstrak informasi penting yang terdapat dalam sebuah gambar. Menggunakan algoritma dan *learning*, visi komputer mengajarkan komputer untuk mengenali, memahami, menangkap, menginterpretasikan, dan mengolah informasi dari data citra, video, dan visual lainnya (Narlan, 2023). Hal ini terjadi melalui proses pengenalan pola yang dilatih menggunakan dataset dalam jumlah besar, dimana terdapat pemodelan untuk mengenali berbagai objek. Metode deteksi objek yang digunakan pada penelitian ini adalah YOLOv3 Tiny dengan Darknet yang memungkinkan menghasilkan tingkat keakuratan deteksi objek yang tinggi.

Algoritma YOLO (*You Only Look Once*) merupakan sebuah metode *deep learning* yang menjadi salah satu bagian dari *machine learning* yang dapat mendeteksi suatu objek secara *real-time* yang memberikan hasil deteksi dengan kecepatan tinggi dan akurat, serta menggunakan klasifikasi *multi-purpose* dalam melakukan sebuah pendeteksian (Salamah dkk, 2022).

Dengan beberapa kelebihan metode YOLO yang telah disebutkan tersebut, dengan dilakukan penelitian ini diharapkan dengan mengimplementasikan metode YOLO pada sistem deteksi *landing pad* untuk *Unmanned Aerial Vehicle* (UAV) *multicopter* dapat menghasilkan hasil yang akurat.

## METODE

### Desain Mekanik

Desain *multicopter* yang digunakan pada penelitian kali ini adalah konfigurasi *frame quadcopter* yaitu dengan menggunakan empat propeller dengan model X *frame* dan juga empat kaki sebagai tahanan wahana ketika berada di atas permukaan tanah. Desain mekanik wahana menggunakan konfigurasi 4 motor BLDC membentuk sudut 45 derajat dengan sumbu X dan Y pada wahana. Berikut desain wahana *multicopter* yang digunakan pada penelitian ini, seperti yang ditunjukkan pada Gambar 1.



(a)



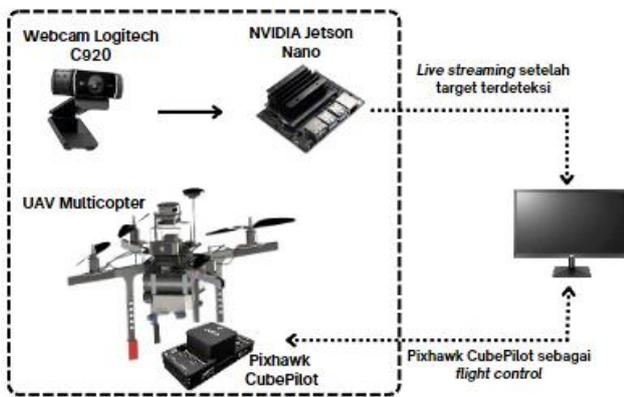
(b)

Gambar 1. (a) Desain *multicopter* tampak samping, (b) Desain *multicopter* tampak atas

Fungsi *webcam* sebagai sensor *vision* pada penelitian kali ini adalah untuk mengambil citra rekaman udara dan mengirimkan data ke NVIDIA Jetson Nano sebagai Mini PC seperti yang ditunjukkan pada Gambar 2. Sistem menghasilkan *frame* hasil deteksi dengan nama dan presentase nilai *confidence*.

### Desain Elektrikal Pendeteksian

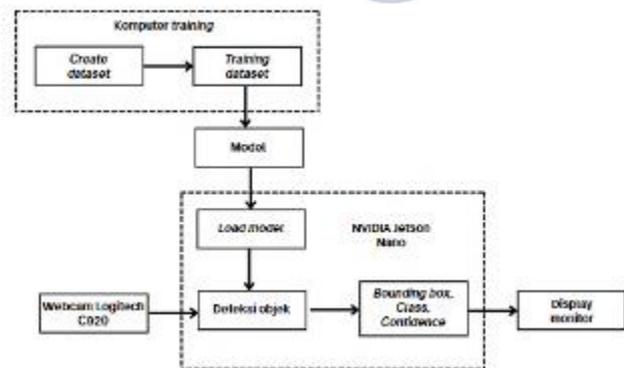
Dalam proses ini, NVIDIA Jetson Nano bertanggung jawab untuk memproses seluruh algoritma dan mengirimkan data melalui jaringan. *Ground station* yang terhubung dengan jaringan serupa melalui *receiver* menerima data yang dikirimkan dan menampilkan hasilnya pada layar monitor. *Flight control multicopter* menggunakan *Flight Control Unit (FCU)* dengan jenis CubePilot yang berfungsi sebagai pengontrol pusat pada *multicopter*. Kegunaan FCU disini adalah sebagai penanggung jawab untuk mengontrol dan mengatur pergerakan UAV *Multicopter* berdasarkan informasi yang telah diterima dari sensor *vision*.



Gambar 2. Skema desain elektrikal pendeteksian

### Perencanaan Sistem Deteksi *Landing pad*

Perencanaan sistem deteksi pada penelitian ini dapat dilihat pada Gambar 3.



Gambar 3. Diagram blok perencanaan *software*

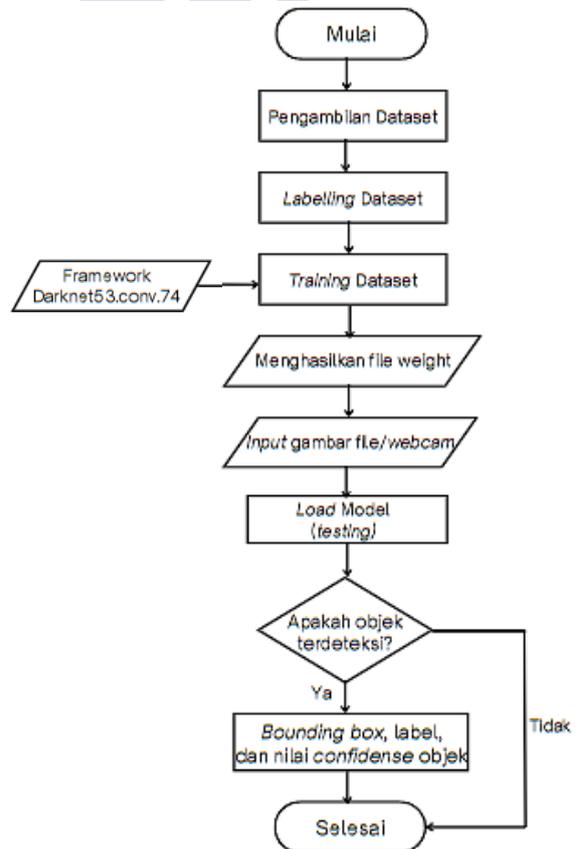
Pada penelitian ini menggunakan bahasa pemrograman Python dengan Google Colaboratory sebagai *platform* pada komputer *training*, *vision* yang digunakan yaitu *webcam* Logitech C920 yang telah terpasang pada prototipe yang terhubung dengan NVIDIA Jetson Nano. Program Python yang dibuat nantinya memuat algoritma YOLO yang kemudian

akan memproses citra dari input yang diperoleh dari *webcam* secara real-time. Unit atau perangkat yang digunakan dalam proses pendeteksian objek *landing pad* menggunakan Mini PC NVIDIA Jetson Nano.

Algoritma YOLO akan membagi gambar menjadi grid dengan ukuran  $s \times s$ . Kemudian, di setiap grid, algoritma akan memprediksi *bounding box* dan peta klasifikasi untuk setiap kelas grid tersebut (Nazilli & Rahmat, 2020). Setelah melalui pemrosesan oleh algoritma YOLO, layar monitor yang telah terhubung dengan NVIDIA Jetson Nano akan menampilkan *display window* berupa *frame* citra yang di dalamnya terdapat objek yang telah terdeteksi dengan ditandai oleh adanya *bounding box* serta nilai *confidence*.

### Implementasi Metode YOLO

Desain sistem deteksi objek berupa *landing pad* menggunakan algoritma YOLO dapat dilihat pada Gambar 4.



Gambar 4. Diagram alir implementasi metode YOLO

Dari diagram alir pada Gambar 4. dapat ditunjukkan beberapa tahapan dalam penerapan dari metode Darknet YOLO, diantaranya sebagai berikut. Pertama, pengambilan dataset, dilakukan pada ketinggian (*altitude*) 1 meter, 2 meter, 4 meter, dan 6 meter. Dan menggunakan resolusi  $320 \times 240$  *pixel* dan kondisi pencahayaan pada pukul 10.00-15.00 WIB. Dataset yang digunakan berjumlah 1000 gambar,

dengan rasio 80% untuk *training set* dan 20% untuk *validation set*.

*Landing pad* yang digunakan berukuran diameter 50cm x 50cm dengan huruf "H" ditengahnya. Gambar 5. merupakan gambar objek *landing pad* yang akan digunakan dalam pengambilan dataset dan pengujian pada sistem deteksi kali ini.



Gambar 5. *Landing pad*

Setelah itu, dataset gambar yang telah terkumpul akan melalui proses pelabelan di setiap *frame* gambar. Pelabelan dataset, merupakan proses penambahan informasi pada citra dalam dataset tersebut. Pada penelitian kali ini menggunakan aplikasi YOLO-Label untuk proses *labelling*. Setiap gambar dataset diberi label satu per satu untuk mendapatkan koordinat *bounding box* sesuai dengan keadaan sebenarnya (*ground-truth bounding box*), yang kemudian akan dibandingkan dengan *bounding box* yang diprediksi. Dengan membandingkan kedua jenis *bounding box* ini, maka akan didapatkan nilai *Intersection over Union* (IoU).

Langkah selanjutnya, *Training* atau pelatihan Dataset, pelatihan dilakukan menggunakan algoritma *You Only Look Once* (YOLO) dengan *framework* Darknet. Proses pelatihan dilakukan menggunakan Google Colaboratory. Model YOLO yang diadopsi adalah YOLOv3 Tiny. *Weight* untuk layer konvolusi menggunakan bobot konvolusi yang telah dilatih sebelumnya dalam *Imagenet* dari model Darknet53.

Selama proses *training*, *hyperparameter* seperti *learning rate* dan momentum dapat disesuaikan untuk meningkatkan konvergensi model. Evaluasi kinerja model pada dataset validasi membantu mengontrol *overfitting* dan memastikan generalisasi yang baik. Iterasi diulang hingga model YOLOv3 Tiny mencapai kinerja yang sesuai yang memungkinkan deteksi objek yang cepat dan akurat pada perangkat keras yang lebih ringan.

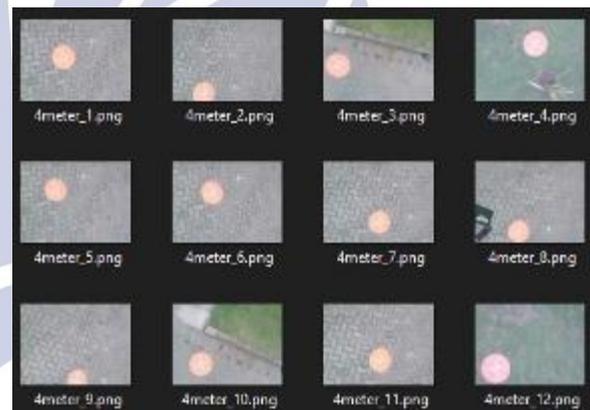
Setelah menyelesaikan tahap *training* dataset, dilakukan pengujian sistem deteksi dengan menggunakan file *weights* yang dihasilkan pada proses *training* sebelumnya. Pengujian sistem ini

dilakukan secara *offline* dengan memberikan *input* gambar ke sistem deteksi. Hasil dari pengujian tersebut akan mencakup *bounding box* yang sesuai dengan prediksi koordinat (x, y, w, h), nilai *confidence*, dan probabilitas kelas dari objek yang terdeteksi.

## HASIL DAN PEMBAHASAN

### Perancangan Sistem Deteksi

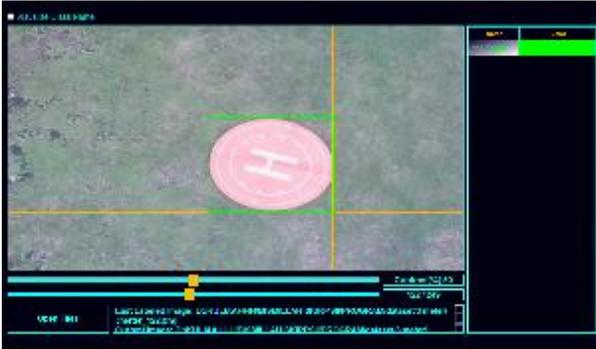
Pada tahap pengambilan dataset, diambil gambar objek berupa *landing pad*. Gambar diambil menggunakan *webcam* yang telah dipasang pada prototipe UAV *Multicopter*. Jumlah dataset yang digunakan yaitu terdapat total 1000 gambar, dengan rasio 80% untuk *training set* dan 20% untuk *validation set*. Pada *training set* digunakan sebanyak 800 gambar dataset dengan pembagian masing-masing ketinggian sebanyak 200 gambar. Pada *validation set* digunakan sebanyak 200 gambar dataset dengan pembagian untuk setiap ketinggian sebanyak 50 gambar. Seperti ditunjukkan pada Gambar 6.



Gambar 6. Hasil pengambilan gambar dataset

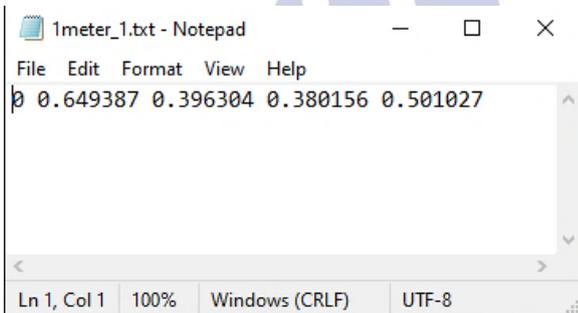
Dataset diambil dengan menggunakan program Python yang dikembangkan dengan *library* OpenCV. Proses pengambilan dataset dilakukan secara *real-time* menggunakan *webcam* untuk menampilkan video pada *form windows*, kemudian program secara otomatis mengambil dan menyimpan *frame* gambar per 1.25 detik-nya pada folder yang telah ditentukan. Tahap selanjutnya yaitu dengan melakukan pelabelan (*labelling*) gambar dataset.

Proses pelabelan objek *landing pad* dilakukan sesuai format anotasi YOLO, yaitu nomor kelas objek, koordinat x dan y nomor objek, serta lebar dan tinggi objek kelas. *Labelling* pada penelitian kali ini menggunakan *software* YoloLabel yang merupakan *software* GUI untuk membuat *bounding box* pada objek dalam gambar yang nantinya akan digunakan untuk *training* YOLO. Seperti yang ditunjukkan pada Gambar 7.



Gambar 7. Pelabelan objek pada gambar dataset

Pelabelan dilakukan dengan melabeli objek *landing pad* satu persatu pada setiap gambar dataset. Hasil dari pelabelan akan disimpan dengan format “.txt” ke dalam folder yang sama yang juga berisi file gambar dataset objek *landing pad*. Gambar 8. ditunjukkan isi file hasil dari pelabelan pada gambar dataset menggunakan YoloLabel.



Gambar 8. Hasil file anotasi pelabelan objek

Diketahui dari isi file hasil pelabelan tersebut terdapat beberapa nilai angka sesuai dengan anotasi objek kelas keluaran jaringan YOLO yang berisi informasi sesuai urutan dari kiri ke kanan yaitu:

- “0” menunjukkan nomor kelas objek yang diberi label, yang dimana dalam penelitian ini hanya menggunakan satu kelas objek yang dideteksi yaitu *landing pad*.
- “0.649387” dan “0.396304” menunjukkan koordinat x dan y dari objek yang terdeteksi pada gambar.
- “0.380156” dan “0.501027” menunjukkan width dan height atau lebar dan tinggi merepresentasikan pusat kotak relatif terhadap gambar.

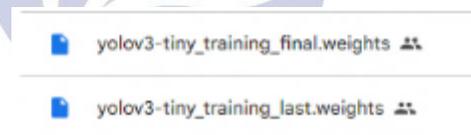
Kemudian langkah selanjutnya adalah melakukan pelatihan atau *training* jaringan deteksi *landing pad* menggunakan YOLOv3 Tiny. Tahap *training* menggunakan sebanyak 800 gambar yang terdiri dari 200 gambar dengan pengambilan pada ketinggian 1 meter, 200 gambar pada 2 meter, 200 gambar pada 4 meter, dan 200 gambar pada 6 meter, serta menggunakan bobot *pretrained* “darknet53.conv.74”. Sebelum melanjutkan ke tahap

pelatihan jaringan, penyesuaian diperlukan untuk mengatur *hyperparameter* agar sesuai dengan jaringan YOLOv3 Tiny. Konfigurasi nilai *hyperparameter* dapat dilihat pada Gambar 9.

```
1 !cp cfg/yolov3-tiny.cfg cfg/yolov3-tiny_training.cfg
2 !sed -i 's/batch=1/batch=64/' cfg/yolov3-tiny_training.cfg
3 !sed -i 's/subdivisions=1/subdivisions=16/' cfg/yolov3-tiny_training.cfg
4 !sed -i 's/max_batches = 10000/max_batches = 5000/' cfg/yolov3-tiny_training.cfg
5 !sed -i 's/0@classes=0@classes=10/' cfg/yolov3-tiny_training.cfg
6 !sed -i 's/177@filters=255@filters=128/' cfg/yolov3-tiny_training.cfg
7 !sed -i 's/127@filters=255@filters=128/' cfg/yolov3-tiny_training.cfg
8 !sed -i 's/171@filters=255@filters=128/' cfg/yolov3-tiny_training.cfg
9 !sed -i 's/140@random=1@random=0/' cfg/yolov3-tiny_training.cfg
10 !sed -i 's/182@random=1@random=0/' cfg/yolov3-tiny_training.cfg
```

Gambar 9. Konfigurasi *hyperparameter* YOLOv3Tiny

Proses pelatihan menggunakan *platform* Google Collaboratory dengan *framework* Darknet. Setelah tahap pelatihan jaringan selesai, akan menghasilkan file yang disimpan di Google Drive dengan nama “yolov3-tiny\_training\_final.weights”, yang berisi file *weight* akhir dari jaringan yang telah dilatih. Selama tahap pelatihan berlangsung, bobot atau *weight* akan disimpan dalam file “yolov3-tiny\_training\_last.weights” setiap 1000 kali *epoch* ke dalam *path backup* yang telah ditentukan dalam file “obj.data”. Berikut ditunjukkan file *weight* yang dihasilkan dari proses *training* pada Gambar 10.

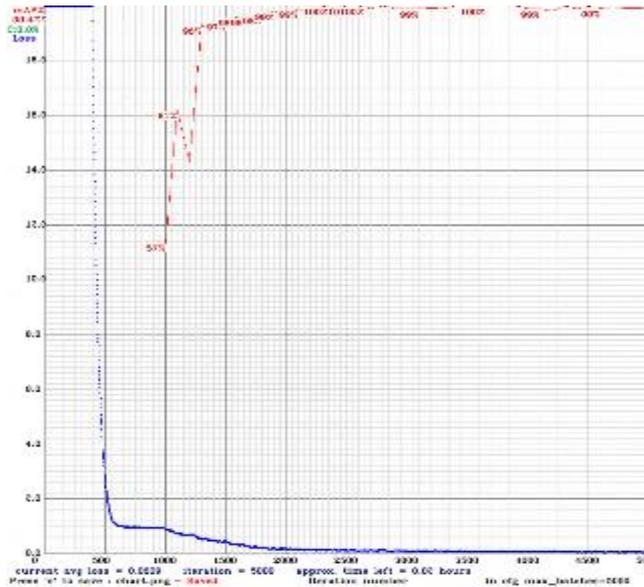


Gambar 10. File *weight* proses pelatihan jaringan

Proses *training* ini memerlukan waktu sekitar 3 jam tanpa gangguan. Hal ini dipengaruhi oleh jumlah data, total kelas, dan nilai “max\_batches” yang digunakan. Pengaturan *hyperparameter* “random” menjadi “0” juga berpengaruh untuk mempercepat proses pelatihan jaringan. Pada Gambar 11. menunjukkan grafik pada saat proses pelatihan jaringan dengan nilai *average loss* sebesar 0.0609 dan nilai mAP sebesar 99.4%.

### Pengujian Sistem Deteksi *Landing pad*

Sistem deteksi objek *landing pad* dengan YOLOv3 Tiny menggunakan beberapa file yang didapatkan pada proses *training* atau pelatihan sebelumnya. File tersebut diantaranya file bobot akhir jaringan YOLOv3 Tiny dengan nama file “yolov3-tiny\_training\_final.weights”, kemudian file konfigurasi untuk pendeteksian objek “yolov3-tiny\_training.cfg”, serta file nama kelas objek “classes.txt”.



Gambar 11. File chart yang dihasilkan proses pelatihan jaringan

Untuk pengujian kali ini, dilakukan perhitungan *confusion matrix* untuk mengevaluasi nilai *accuracy*, *recall*, *precision*, dan *F1-score* dari sistem deteksi. *Confusion matrix* sendiri merupakan metode untuk mengevaluasi kinerja dari sistem klasifikasi multikelas dimana metode ini menampilkan jumlah prediksi benar dan salah untuk setiap kelas target (Nugroho & Anifah, 2023).

Dalam metode *confusion matrix* ini, TP (*True positive*) mengacu pada jumlah data yang sebenarnya positif dan diprediksi dengan benar sebagai positif oleh sistem. FP (*False positive*) merujuk pada jumlah data yang sebenarnya negatif namun salah diprediksi sebagai positif oleh sistem. TN (*True negative*) yaitu jumlah data yang sebenarnya negatif dan diprediksi dengan benar sebagai negatif oleh sistem. FN (*False negative*) mengindikasikan jumlah data yang sebenarnya positif tetapi salah diprediksi sebagai negatif oleh sistem (Nugroho & Anifah, 2023).

Nilai *accuracy* merupakan ukuran seberapa sering sistem deteksi memberikan prediksi yang benar secara keseluruhan. *Recall* adalah ukuran kemampuan model dalam menemukan semua objek yang relevan dengan persentase positif yang benar dan terdeteksi dari semua data *ground truth*. *Precision* adalah ukuran kemampuan model dalam mengenali objek yang relevan dengan persentase prediksi yang benar. *F1-score* merupakan ukuran kombinasi antara nilai *precision* dan nilai *recall* (Nugroho & Anifah, 2023).

### Pengujian Peforma YOLOv3 Tiny

Pada pengujian peforma sistem deteksi *landing pad* dilakukan menggunakan dataset *validation set*, terdiri

dari total 200 gambar dataset yang diambil pada pukul 15.00 WIB dengan rasio pembagian 50 gambar dengan pengambilan gambar dari ketinggian 1 meter, 50 gambar dari ketinggian 2 meter, 50 gambar dari ketinggian 4 meter, dan 50 gambar dari ketinggian 6 meter. Perhitungan *confusion matrix* dilakukan menggunakan gambar langsung dari dataset folder *validation set*. Gambar 12. ditunjukkan hasil pengujian sistem deteksi menggunakan *validation set*.



Gambar 12. Hasil pengujian sistem menggunakan *validation set*

Berikut ditunjukkan hasil perhitungan *confusion matrix* dari pengujian peforma jaringan YOLOv3 Tiny.

Tabel 1. *Confusion matrix* pada gambar ketinggian 1 meter

		1 meter	
		<i>Landing pad</i>	Bukan <i>Landing pad</i>
Prediksi	Aktual <i>Landing pad</i>	44	1
	Bukan <i>Landing pad</i>	2	3

Berdasarkan Tabel 1. Didapatkan tabel *confusion matrix* yang diperoleh dari pengujian sistem menggunakan *validation set* dengan gambar ketinggian 1 meter. Dari tabel menunjukkan *landing pad* yang terdeteksi untuk Nilai TP (*True Positive*) sebanyak 44 menunjukkan bahwa 44 objek *landing pad* benar-benar diprediksi dengan benar oleh sistem sebagai positif. Nilai FN (*False Negative*) sebanyak 2 menunjukkan bahwa 2 objek *landing pad* yang sebenarnya positif namun diprediksi sebagai negatif oleh sistem. Nilai FP (*False Positive*) sebanyak 1 menunjukkan bahwa 1 objek *landing pad* yang sebenarnya negatif namun diprediksi sebagai positif oleh sistem. Nilai TN (*True Negative*) sebanyak 3 menunjukkan bahwa 3 objek *landing pad* benar-benar diprediksi dengan benar oleh sistem sebagai negatif.

Tabel 2. Confusion matrix pada gambar ketinggian 2 meter

		2 meter	
		Landing pad	Bukan Landing pad
Prediksi	Aktual Landing pad	44	0
	Aktual Bukan Landing pad	1	5

Berdasarkan Tabel 2. menunjukkan landing pad yang terdeteksi untuk Nilai TP (True positive) sebanyak 44. Nilai FN (False negative) sebanyak 1. Nilai FP (False positive) sebanyak 0. Dan nilai TN (True negative) sebanyak 5.

Tabel 3. Confusion matrix pada gambar ketinggian 4 meter

		4 meter	
		Landing pad	Bukan Landing pad
Prediksi	Aktual Landing pad	40	1
	Aktual Bukan Landing pad	1	8

Berdasarkan Tabel 3. menunjukkan landing pad yang terdeteksi untuk Nilai TP (True positive) sebanyak 40. Nilai FN (False negative) sebanyak 1. Nilai FP (False positive) sebanyak 1. Dan nilai TN (True negative) sebanyak 8.

Tabel 4. Confusion matrix pada gambar ketinggian 6 meter

		6 meter	
		Landing pad	Bukan Landing pad
Prediksi	Aktual Landing pad	40	0
	Aktual Bukan Landing pad	3	7

Berdasarkan Tabel 4. menunjukkan landing pad yang terdeteksi untuk Nilai TP (True positive) sebanyak 40. Nilai FN (False negative) sebanyak 3. Nilai FP (False positive) sebanyak 0. Dan nilai TN (True negative) sebanyak 7.

Berikut adalah persamaan untuk menghitung nilai accuracy, precision, recall, dan F1-score pada pengujian validation set pada persamaan 1, 2, 3, dan 4.

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

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

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

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

Tabel 5. Peforma hasil pengujian validation set secara keseluruhan

KP	Peforma							
	T P	F N	F P	T N	A	P	R	F
1m	44	2	1	3	0.94	0.98	0.96	0.93
2m	44	1	0	5	0.98	1.00	0.98	0.98
4m	40	1	1	8	0.96	0.97	0.97	0.97
6m	40	3	0	7	0.94	1.00	0.93	0.96

Keterangan: KP = Ketinggian Pengujian

A = Accuracy

P = Precision

R = Recall

F = F1-score

Berdasarkan Tabel 5. diatas, banyaknya jumlah FN dan FP yang terdeteksi selain disebabkan karena kesalahan pembacaan pada sistem juga disebabkan proses pengambilan gambar validation set yang dilakukan secara random sehingga mendapatkan hasil yang juga random yang didalamnya ada yang terdapat objek landing pad maupun gambar yang tidak terdapat objek landing pad. Untuk nilai accuracy, precision, recall, F1-score yang bervariasi pada pengujian ini, dikarenakan pengaruh dari karakteristik lingkungan, kondisi pengambilan data (Hadi, 2022), dan juga stabilitas dari gerak drone multicopter saat melakukan pengambilan gambar.

Jarak yang lebih tinggi atau jauh antara multicopter dan objek juga memberikan sudut pandang yang lebih luas yang dapat menyediakan banyak informasi tentang objek yang dideteksi sehingga membantu algoritma untuk mendeteksi lebih akurat.

## PENUTUP

### Simpulan

Dari penelitian yang telah dilakukan, didapatkan kesimpulan sebagai berikut. Rancang bangun sistem pendeteksian landing pad telah diimplementasikan pada UAV Multicopter menggunakan metode YOLO Darknet. Pada tahap training, sistem menggunakan framework Darknet dengan konfigurasi YOLOv3 Tiny. Dalam tahap pelatihan, model dilatih untuk mendeteksi kelas objek "landing pad", yang menghasilkan nilai average loss sebesar 0.0609 dan

nilai mAP sebesar 99.4%. Setelah tahap pelatihan, file *weights* yang dihasilkan diuji menggunakan program Python dengan *library* OpenCV dan algoritma YOLO. Hasil pengujian menggunakan *validation set* didapatkan nilai peforma jaringan dengan rata-rata nilai *accuracy* sebesar 0.95, *precision* 0.98, *recall* 0.96, dan *F1-score* sebesar 0.96.

### Saran

Meningkatkan jumlah dan variasi dari dataset gambar, seperti perbedaan sudut pengambilan gambar, resolusi citra, dan penggunaan filter pada gambar. Hal ini bertujuan agar jaringan dapat mengenali berbagai variasi objek *landing pad* pada berbagai kondisi dan lapangan.

### DAFTAR PUSTAKA

- Abbiyansyah, Mochammad Zava, Eko Setiawan, dan Agung Setia Budi. 2022. *Navigasi Halang Rintang Quadcopter Tanpa Awak menggunakan RPLIDAR dan Algoritma Bug2 berbasis Raspberry Pi*. JPTIIK, Vol 6 No 11, 5413-5421.
- Ashari, Faisal, dan Fuad Achmadi. 2021. *Desain Pengembangan Produk Robot Terbang Tanpa Awak dengan Metode QFD (Quality Function Deployment)*. Jurnal SENOPATI: Sustainability, Ergonomics, Optimization, and Application of Industrial Engineering, 2(2), 66-73.
- Fauzi, Julian Fuad, Herman Tolle, dan Ratih Kartika Dewi. 2018. *Implementasi Metode RGB To HSV pada Aplikasi Pengenalan Mata Uang Kertas Berbasis Android untuk Tuna Netra*. JPTIIK, 2(6), 2319-2325.
- Firdaus, A. R., Tranganu, M. A., Hutagalung, A., Syahputra, A., dan Analia, R. 2023. *Camera Based Auto Positioning Drone*. In AIP Conference Proceedings (Vol. 2665, No. 1), AIP Publishing.
- Hadi, Muhammad Irfaan, Desri Kristina Silalahi, dan Ig Prasetya Dwi Wibawa. 2022. *Pengaturan Lampu Lalu Lintas Berdasarkan Deteksi Volume Kendaraan Menggunakan Metode YOLOv3*. eProceedings of Engineering, 9(5).
- Ikhsan, Ismawan Noor, dan Son Ali Akbar. 2020. *Aplikasi Machine Vision pada Hexacopter untuk Deteksi Survival Kits di Bidang Mitigasi Bencana*. Jurnal Teknik Elektro, 12(2), 72-79.
- Medeiros, Felipe Leonardo Lôbo, dkk. 2015. *A Computer vision System for Guidance of VTOL UAVs Autonomous Landing*. In 2015 Brazilian Conference on Intelligent Systems (BRACIS) (pp. 333-338). IEEE.
- Narlan, Ryandra. 2023. *Penerapan Visi Komputer Dalam Bidang Konstruksi*. Proceeding Civil Engineering Research Forum, Vol. 2, No. 2.
- Nazilly, Mada Lazuardi, Basuki Rahmat, dan Eva Yulia Puspaningrum. 2020. *Implementasi Algoritma YOLO (You Only Look Once) untuk Deteksi Api*. Jurnal Informatika dan Sistem Informasi, Vol. 1, No.1, 81-91.
- Nugroho, Dwi Novianto, dan Lilik Anifah. 2023. *Perancangan Sistem Deteksi Objek Bola Dan Gawang Pada Robot Sepakbola Menggunakan Metode Darknet YOLO*. JIEET (Journal of Information Engineering and Educational Technology), 7(1), 22-29.
- Nuryadi, I. H. 2023. *Pedoman Kontes Robot Terbang Indonesia (KRTI) Tahun 2023*. Jakarta: Balai Pengembangan Talenta Indonesia Pusat Prestasi Nasional Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi.
- Salamah, Irma, M. Redho Ali Said, dan Sopian Soim. 2022. *Perancangan Alat Identifikasi Wajah Dengan Algoritma You Only Look Once (YOLO) Untuk Presensi Mahasiswa*. Jurnal Media Informatika Budidarma, 6(3), 1492-1500.