

# Implementasi YOLOv8 Pada Deteksi Jenis Pohon Menggunakan Drone

1<sup>st</sup> Muhammad Izzudin Islam  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

islamizzudin@student.telkomuniversity.ac.id

2<sup>nd</sup> Casi Setianingsih  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

3<sup>rd</sup> Marisa W Paryasto  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

marisaparyasto@telkomuniversity.ac.id

**Abstrak** — Penelitian ini menggambarkan pemanfaatan drone dan kecerdasan buatan (AI) khususnya menggunakan algoritma YOLOv8 dalam pendeteksian jenis pohon secara cepat dan efisien. Pohon memiliki peran vital dalam ekosistem dan perubahan iklim, konservasi hutan, dan pengelolaan sumber daya alam. Dengan kemajuan drone, AI, dan teknologi computer vision, proses ini dapat dioptimalkan, menghemat waktu dan sumber daya manusia. Penggunaan drone memungkinkan akses ke wilayah sulit dijangkau, sementara AI dengan algoritma YOLOv8 yang dioptimalkan untuk computer vision dapat digunakan untuk mendeteksi dan mengidentifikasi pohon secara otomatis dalam gambar atau video secara real-time. Tantangan utama adalah keterbatasan dataset untuk melatih model AI, tetapi kemajuan dalam teknologi drone, AI, dan computer vision menawarkan potensi besar untuk meningkatkan efisiensi dan akurasi dalam deteksi jenis pohon. Hasil dari penelitian ini menunjukkan bahwa penggunaan drone dan algoritma YOLOv8 merupakan kombinasi yang efisien dan akurat dalam pengukuran dan klasifikasi tinggi pohon. Model yang dikembangkan memiliki nilai performance sebesar 88.57% precision, 86.14% recall, 93.98% mAP50 dan 68.10% mAP50-90. Serta sistem yang dikembangkan memiliki akurasi yang cukup baik dengan confidence score rata-rata 87%. Teknologi ini memiliki potensi besar dalam mendukung berbagai aplikasi, termasuk pemantauan pertumbuhan hutan, konservasi sumber daya alam, dan penilaian dampak perubahan iklim terhadap ekosistem hutan. Namun, perlu dicatat bahwa perbaikan lebih lanjut pada algoritma dan penyesuaian teknis lainnya dapat terus meningkatkan kinerja dan potensi sistem ini dimasa depan.

**Kata kunci** — YOLOv8, Drone, Kecerdasan Buatan, Klasifikasi, Pohon.

## I. PENDAHULUAN

Pohon memainkan peran yang sangat penting tidak hanya dalam sektor kayu, tetapi juga dalam siklus karbon global dan perubahan iklim. Tanpa pohon, makhluk hidup tidak dapat bertahan hidup karena pohon merupakan sumber dari segala kehidupan bagi makhluk hidup.

Dengan kemajuan teknologi, penggunaan dan aplikasi kendaraan udara tanpa awak (Unmanned Aerial Vehicle/UAV) atau drone menjadi opsi yang membantu manusia dalam menjalani kehidupan sehari-hari. UAV merupakan pesawat terbang yang dapat beroperasi tanpa awak yang dikendalikan secara jarak jauh atau otomatis [1],

[2]. Penggunaan drone semakin berkembang karena drone dapat menjangkau wilayah yang sulit dijangkau dan dapat menyediakan data secara real-time yang pada akhirnya membuka atau mengubah cara kerja dan interaksi dengan lingkungan sekitar.

Dalam beberapa tahun terakhir, perkembangan AI (Artificial Intelligence) telah mengalami kemajuan pesat. AI adalah kemampuan sistem komputer untuk meniru, memahami, dan melakukan tugas yang biasanya secara tradisional menggunakan kecerdasan manusia. AI melibatkan penggunaan algoritma dan teknik komputasi yang kompleks untuk memungkinkan mesin mengumpulkan, menganalisis, dan memproses informasi, serta membuat keputusan atau tindakan berdasarkan pemahaman tersebut.[2].

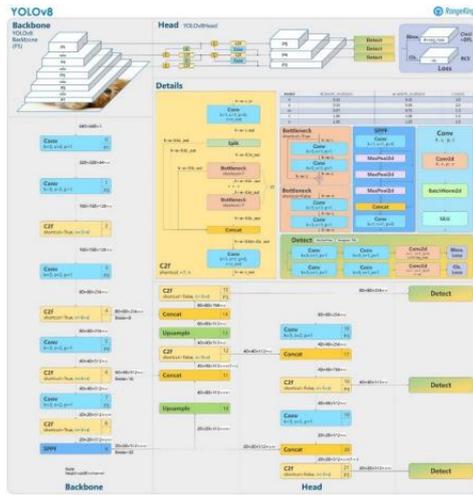
Drone dan AI memiliki potensi besar untuk proses klasifikasi jenis pohon dan juga pengukuran ketinggiannya. Dengan memanfaatkan kedua teknologi tersebut dapat dilakukan klasifikasi dan juga pengukuran ketinggiannya lebih cepat dan akurat. Salah satu tantangannya adalah keterbatasan dataset yang diperlukan untuk proses training. Dengan keterbatasan dari proses deteksi jenis pohon secara manual, maka dirancanglah sistem implementasi YOLOv8 pada deteksi jenis pohon menggunakan drone untuk mengoptimalkan proses tersebut agar lebih cepat dan akurat.

## II. KAJIAN TEORI

### A. YOLO

YOLO (You Only Look Once) adalah algoritma Object Detection secara real-time yang umum digunakan. YOLO terus dikembangkan dari versi pertama dari Joseph Redmon, sampai saat ini yang terbaru adalah YOLOv8 yang dirilis secara resmi pada Januari 2023. Hasil dari riset yang dilakukan terkait perbandingan dengan versi sebelumnya, YOLOv8 menghasilkan mAP yang lebih baik daripada YOLO versi sebelumnya [3]. Algoritma YOLO membagi semua gambar input yang diberikan ke dalam sistem grid  $S \times S$ . Setiap grid bertanggung jawab untuk mendeteksi objek. Sel-sel grid itu memprediksi bounding boxes untuk objek yang terdeteksi. Untuk setiap box memiliki lima atribut utama yaitu  $x$  dan  $y$  untuk koordinat,  $w$  dan  $h$  untuk lebar dan tinggi objek, dan confidence score untuk probabilitas bahwa box tersebut berisi objek [4].

B. YOLOv8



GAMBAR 1  
Arsitektur YOLOv8

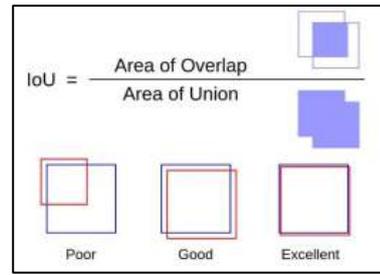
YOLOv8 dirilis pada tahun 2023 oleh Glenn Jocher (Founder & CEO of Ultralytics). YOLOv8 adalah salah satu algoritma object detection terbaru untuk seri YOLO. Sebagai algoritma terbaru dalam seri YOLO saat ini, Struktur jaringan YOLOv8 terdiri dari beberapa lapisan seperti lapisan input, jaringan backbone, jaringan neck dan lapisan output [5]. YOLOv8 memiliki beberapa layer yang membangun struktur jaringannya seperti layer Conv, C2f, Concat, Upsample, SPPF. Layer C2f pada YOLOv8 berisi CSPBottleneck dengan 3 proses konvolusi yang berfungsi untuk memelihara fitur melalui propagasi, mendorong jaringan untuk menggunakan kembali fiturnya, mengurangi jumlah parameter jaringan dan membantu mempertahankan fitur yang halus untuk diteruskan ke lapisan yang lebih dalam dengan lebih efisien [5].

C. Intersection Over Union

Intersection over union (IOU) adalah sebuah istilah yang digunakan untuk mengukur nilai tumpang tindih antara dua kotak. Pada bidang deteksi objek IOU digunakan untuk mengevaluasi bounding box prediksi dan bounding box yang sebenarnya. Hal ini menentukan seberapa benar prediksi yang dilakukan saat melakukan deteksi objek. Nilai IOU dapat dihitung dengan Persamaan [6]

$$IOU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{1}$$

Tujuan utama dari penggunaan IOU adalah untuk meningkatkan prediksinya sampai kedua bounding box bertumpang tindih secara sempurna, yaitu IOU antara dua kotak menjadi sama dengan 1 [19]. Gambar 3.7 merepresentasikan area bounding box yang saling tumpang tindih.



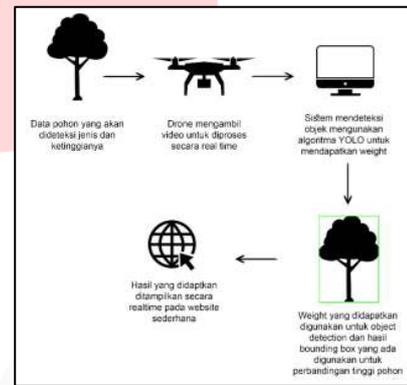
GAMBAR 2  
Representasi IOU

III. PERANCANGAN SISTEM

A. Arsitektur Sistem

1. Gambaran umum sistem

Berikut adalah gambaran umum sistem yang dikembangkan.

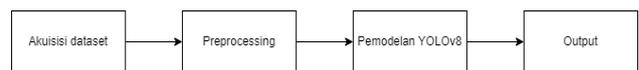


GAMBAR 3  
Gambaran umum system

Sistem diimplementasikan pada drone secara real-time untuk mendeteksi jenis pohon dan mengukur ketinggian pohon, sistem menggunakan algoritma YOLO untuk proses pendeteksian, serta menggunakan drone agar area pengukuran lebih luas dan waktu pengukuran menjadi lebih efisien dibandingkan menggunakan alat ukur manual seperti hagameter dan clinometer ataupun menggunakan kamera manual.

2. Diagram Blok Sistem

Sistem yang dikembangkan menggunakan algoritma YOLOv8 untuk melakukan deteksi jenis pohon. Tahap pertama pada sistem ini adalah akuisisi data. Akuisisi data bertujuan untuk mengumpulkan data berupa citra pohon yang telah ditentukan dari berbagai sumber. Tahap selanjutnya adalah preprocessing yang bertujuan untuk menyesuaikan dataset sebagai input untuk training model. Tahap berikutnya adalah pemodelan YOLOv8 untuk mendeteksi pohon.

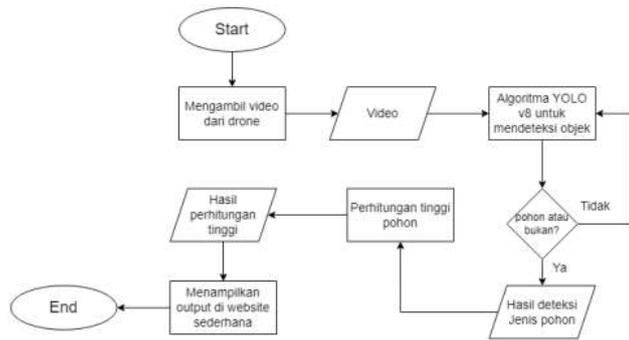


GAMBAR 4  
diagram blok system

3. Flowchart Sistem

Pada flowchart bisa dilihat sistem akan mengambil input berupa video yang didapatkan melalui drone, video yang didapatkan akan diproses oleh algoritma untuk mendeteksi objek. Jika objek terdeteksi maka hasil dari

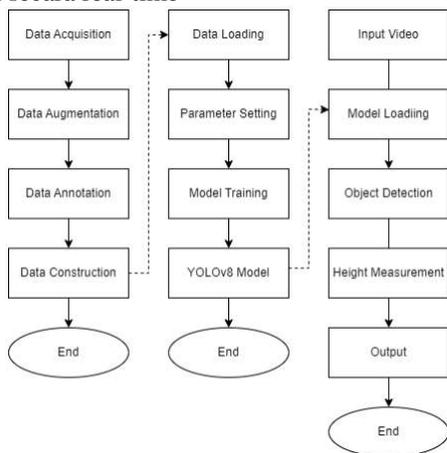
deteksi tersebut akan digunakan untuk proses mengukur tinggi pohon. Jika jenis dan tinggi pohon sudah didapatkan maka sistem akan menampilkannya pada website sederhana dan sistem selesai.



GAMBAR 5 flowchart system

4. Diagram Sistem

Diagram pada gambar 8 menampilkan proses yang berlangsung secara terpisah, dimulai dari dataset construction yang menjelaskan terkait proses pembuatan dataset, Model Training yang menjelaskan terkait proses training model YOLO v8, dan Tree detection yang menjelaskan proses object detection secara real-time



GAMBAR 6 Diagram keseluruhan system

B. Implementasi Sistem

1. Akuisisi dataset

Dataset yang digunakan oleh sistem ini adalah gambar dengan format jpg, jpeg, dan png. Proses akuisisi dataset dilakukan dengan cara mencari gambar pohon yang telah ditentukan yaitu pisang-pisangan, cemara-cemaraan, dan palem-paleman. Pencarian gambar pohon dilakukan melalui berbagai macam website seperti pinterest, facebook, google image, dan lain-lain. Beberapa data gambar juga didapatkan dengan cara mengambil gambar pohon secara langsung dilapangan. Adapun lable pada dataset ini sesuai dengan jenis pohonnya yaitu pisang, palm, dan cemara. Total dataset yang digunakan berjumlah 1600 data yang terdiri dari 400 data pohon palem, 400 data pohon pisang dan 400 data pohon cemara, serta 400 data null.



GAMBAR 7 Contoh data yang digunakan

Pada proses akuisisi dilakukan juga anotasi dataset. Anotasi dataset adalah proses menambahkan label atau metadata ke dalam data mentah untuk memberikan informasi yang lebih spesifik tentang konten data. Dalam konteks machine learning, anotasi dataset adalah langkah krusial untuk pelatihan algoritma pembelajaran mesin. Dengan menerapkan anotasi pada data, algoritma dapat belajar dan mengenali pola-pola yang relevan dalam data tersebut. Pada sistem yang dikembangkan anotasi dilakukan dengan memberikan label sesuai dengan kelasnya yaitu pisang, palm, dan pinus. Hal ini memungkinkan algoritma pembelajaran mesin untuk memahami perbedaan dan persamaan antar kelas, sehingga mampu melakukan prediksi yang tepat pada data baru yang belum pernah dilihat sebelumnya.

Pada sistem yang dikembangkan proses anotasi dilakukan menggunakan platform roboflow. Roboflow adalah platform yang memudahkan pekerjaan terkait computer vision dalam bidang deep learning. Roboflow memudahkan pengguna untuk membangun aplikasi computer vision dengan berbagai macam fitur diantaranya, anotasi dataset, pre-processing dataset dan training model.

2. Preprocessing

Pre-processing adalah proses perbaikan gambar untuk menyesuaikan dataset sebagai input training model. Pre-processing dilakukan pada semua dataset yang telah dibuat sebelumnya. Metode yang digunakan pada sistem ini adalah auto-orient, dan pengaturan brightness. Auto-orient digunakan untuk menjamin citra yang diunggah menjadi dataset yang sesuai dengan orientasi citra orisinalnya, brightness dilakukan dengan menambah atau mengurangi kecerahan gambar.



GAMBAR 8 Proses preprocessing

3. Pemodelan YOLOv8

Agar pengaplikasian model YOLO v8 bekerja dengan optimal pada sistem, terdapat beberapa faktor yang perlu diperhatikan yaitu nilai akurasi yang baik. Untuk mendapatkan nilai akurasi yang cukup baik maka model perlu dilatih menggunakan parameter yang berbeda beda. Parameter yang diuji untuk training antara lain perbandingan rasio data training, validasi, dan test, learning

rate, epoch dan batch size. Jika semua tahap sudah dilakukan akan mendapatkan model terbaik yang akan diimplementasikan pada object detection.

4. Output

Setelah melalui proses pemodelan video yang sudah diproses oleh sistem akan ditampilkan secara real-time pada website sederhana yang dibuat menggunakan framework streamlit. Di dalam website user dapat memantau langsung proses pendeteksian secara real-time dan juga mengatur nilai confidence threshold dan jarak yang diinginkan untuk memaksimalkan hasil pendeteksian oleh sistem. Gambar 9 adalah tampilan hasil akhir output pada website.



GAMBAR 9 Output system

IV. HASIL DAN PEMBAHASAN

Setelah melakukan seluruh implementasi dengan memperhatikan kriteria dan fitur yang akan diimplementasikan, sistem sudah bisa diimplementasikan dengan baik sesuai dengan hasil yang diharapkan. Proses data construction dapat dilakukan dengan baik sehingga mempercepat proses training model untuk selanjutnya diolah pada proses object detection dan pengukuran tinggi. Hasil akhir dari sistem yang dikembangkan user dapat mengukur dan mengetahui jenis pohon melalui website sederhana. Untuk memaksimalkan sistem yang dikembangkan maka diperlukan pengujian sistem lebih lanjut lagi.

Untuk memaksimalkan sistem yang dikembangkan maka diperlukan beberapa pengujian, pada sistem yang dikembangkan terdapat dua pengujian, yaitu pengujian model terbaik, dan pengujian implementasi sistem. Pengujian model terbaik dilakukan untuk mendapatkan model dengan performa paling maksimal yang bisa dikembangkan, sedangkan pengujian implementasi sistem dilakukan untuk mengetahui kemampuan maksimal yang dapat dilakukan oleh sistem.

A. Pengujian Model Terbaik

1. Pengujian partisi data

Dataset yang sudah dikumpulkan dan dianotasi sesuai dengan kelas "pisang", "palm", dan "pinus". akan dibagi partisi menjadi data train, validate, dan test dalam 4 kondisi. Sementara parameter batch size, learning rate, dan epoch menggunakan nilai default yang direkomendasikan oleh YOLOv8. Terdapat empat kondisi pembagian partisi seperti pada tabel 1, masing-masing partisi tersebut akan diuji menggunakan parameter default.

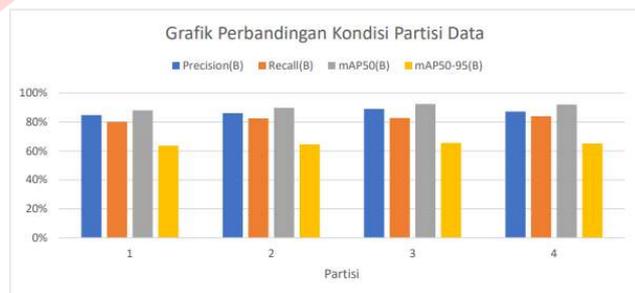
TABEL 1 partisi data

Kondisi	Partisi (%)		
	Train	Val	Test
1	70	20	10
2	75	15	10
3	80	10	10
4	85	10	5

Setelah dilakukan pengujian maka didapatkan hasil seperti pada tabel 2.

TABEL 2 pengujian partisi

Partisi (%)			Precision	Recall	mAP50	mAP50-95
Train	Val	Test				
70	20	10	84.78%	80.14%	88.01%	63.65%
75	15	10	86.11%	82.46%	89.82%	64.54%
80	10	10	89.07%	82.77%	92.39%	65.51%
85	10	5	87.24%	83.99%	92.08%	65.10%



GAMBAR 10 grafik partisi data

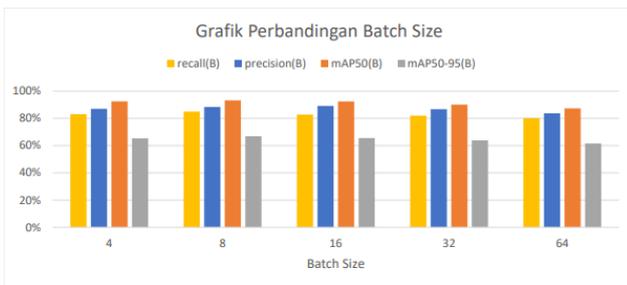
Berdasarkan tabel 2 dan gambar 10 didapatkan hasil partisi dengan kondisi 80% data train, 10% data validation dan 10% data train adalah kondisi dengan nilai performance terbaik. Dengan nilai 89.07% precission, 82.77% recall, 92.39% mAP50, dan 65.51% mAP50-95. Hasil tersebut yang digunakan untuk melakukan pengujian selanjutnya, yaitu pengujian batch size

2. Pengujian Batch Size

Setelah mendapatkan hasil terbaik pada pengujian partisi data, selanjutnya dilakukan pengujian batch size untuk menentukan batch size dengan performa model terbaik. Parameter lain menggunakan nilai default yang direkomendasikan oleh sistem. Nilai batch size yang digunakan pada pengujian ini adalah 4, 8, 16, 32, dan 64. Tabel 3 merupakan hasil dari pengujian yang dilakukan.

TABEL 3 pengujian batch size

Batch	precision	recall	mAP50	mAP50-95
4	87.00%	83.08%	92.46%	65.25%
8	88.36%	85.00%	93.23%	66.81%
16	89.07%	82.77%	92.39%	65.51%
32	86.72%	81.91%	90.08%	63.83%
64	83.69%	80.14%	87.29%	61.53%



GAMBAR 11 grafik batch size

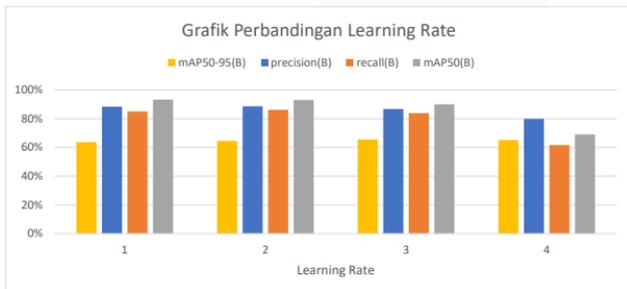
Berdasarkan tabel 3 hasil pengujian batch size 8 adalah hasil terbaik dengan nilai precision 88.36%, recall 85.00%, mAP50 93.23%, dan mAP50-90 66.81%. Hasil tersebut selanjutnya digunakan untuk pengujian learning rate.

3. Pengujian Learning Rate

Hasil terbaik dari proses pengujian sebelumnya yaitu pengujian partisi data dan pengujian batch size selanjutnya diuji kembali dengan pengujian learning rate untuk meningkatkan lagi performa model yang akan digunakan. Nilai learning rate yang akan di uji adalah 0.01, 0.001, 0.0001, dan 0.00001. tabel 4 merupakan hasil dari pengujian yang dilakukan

TABEL 4 pengujian learning rate

Learning Rate	precision(B)	recall(B)	mAP50(B)	mAP50-95(B)
0.01	88.36%	85.00%	93.23%	66.81%
0.001	88.57%	86.14%	92.98%	68.10%
0.0001	86.70%	83.79%	90.00%	63.94%
0.00001	79.88%	61.66%	68.97%	48.05%



GAMBAR 12 pengujian learning rate

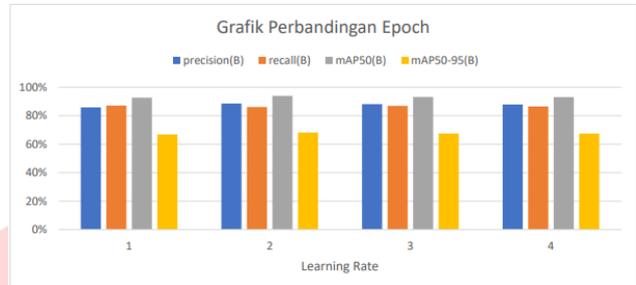
Hasil pengujian learning rate pada tabel 4 menunjukkan learning rate dengan nilai 0.001 memiliki nilai performance yang lebih baik dibandingkan dengan learning rate lainnya. Learning rate dengan nilai 0.001 mendapatkan nilai performance 88.57% precision, 86.14% recall, 92,98% mAP50, dan 68.10% mAP 68.10%. Hasil yang didapatkan akan digunakan untuk pengujian epoch.

4. Pengujian Epoch

Pengujian epoch adalah pengujian yang terakhir, pengujian epoch dilakukan setelah selesai melakukan tiga pengujian sebelumnya, nilai epoch yang dipakai pada pengujian ini adalah 50, 100, 150, dan 200. Tabel xxx adalah hasil dari pengujian yang dilakukan.

TABEL 5 pengujian epoch

Epoch	precision(B)	recall(B)	mAP50(B)	mAP50-95(B)
50	85.92%	87.08%	92.71%	66.82%
100	88.57%	86.14%	93.98%	68.10%
150	88.17%	86.94%	93.23%	67.45%
200	87.88%	86.55%	93.10%	67.43%

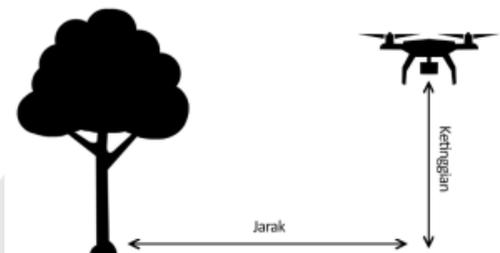


GAMBAR 13 pengujian epoch

Pengujian epoch menunjukkan pada epoch 100 hasil training model mendapatkan nilai performance paling tinggi dengan precision 88.57%, recall 86.14%, mAP50 93.98% dan mAP50-90 68.10%

B. Pengujian Jarak dan Ketinggian Drone

Pengujian ini dilakukan untuk membandingkan jarak optimal deteksi tiap kelas pohon. Proses pengujian yang dilakukan adalah mengukur nilai confidence, serta membandingkan ketinggian hasil pendeteksian dengan ketinggian pohon sesungguhnya untuk mendapatkan nilai eror dan juga akurasi dari sistem yang dikembangkan.



GAMBAR 14 ilustrasi pengujian jarak dan tinggi drone

TABEL 6 pengujian tinggi dan jarak drone

Jarak (m)	Ketinggian (m)	Confidence Score
0	5	29%
5	5	69%
10	5	85%
15	5	80%
20	5	70%
0	7	31%
5	7	65%
10	7	69%

Jarak (m)	Ketinggian (m)	Confidence Score
15	7	46%
20	7	0%
0	10	0%
5	10	60%
10	10	64%
15	10	44%
20	10	31%

Pada kelas pisang jarak terbaik adalah 10 meter dan tinggi terbaik adalah 5 meter dengan nilai confidence score 79%, dengan error rata rata pada kelas tersebut 41%, dan akurasi rata-rata 59%. Pada kelas pinus jarak terbaik adalah 10 meter dan tinggi terbaik adalah 5 meter dengan nilai confidence score 88%, dengan error rata rata pada kelas tersebut 37%, dan akurasi rata-rata 63%. Pada kelas palm jarak terbaik adalah 15 meter dan tinggi terbaik adalah 7 meter dengan nilai confidence score 79%, dengan error rata rata pada kelas tersebut 30%, dan akurasi rata-rata 70%. Sehingga rata rata jarak dan ketinggian terbaik untuk implementasi sistem ini adalah 11.67 meter dan 5.67 meter, dengan confidence score rata rata 87%, error 36%, dan akurasi 64%.

#### V. KESIMPULAN

Pada implementasi sistem tree height detection using drone dilakukan menggunakan model YOLOv8s. Model terbaik yang digunakan adalah model hasil training dengan partisi data sebesar 80% data train, 10% data validation, dan 10% data test. Model tersebut ditraining menggunakan nilai parameter 8 batch size, 0.001 learning rate, dan 100 epoch. Model terbaik tersebut mendapatkan nilai performance sebesar 88.57% precision, 86.14% recall, 93.98% mAP50 dan 68.10% mAP50-90. Adapun beberapa kesimpulan yang diperoleh dari hasil pengujian dan analisis sistem yang dikembangkan adalah sebagai berikut:

1. Telah dikembangkan Implementasi YOLOv8 pada sistem deteksi pohon menggunakan drone.
2. Pada sistem yang dikembangkan confidence threshold default yang digunakan adalah 50% karena memiliki nilai error paling rendah dan akurasi paling tinggi.
3. Rata-rata Jarak dan ketinggian paling optimal pada implementasi sistem ini adalah adalah 11.67 meter dan 5.67 meter, dengan confidence score rata rata 87%, error 36%, dan akurasi 64%. Walaupun pengguna dapat mengatur jarak perbandingan agar mendapatkan hasil yang maksimal.

#### REFERENSI

- [1] H.-J. Hsu and K.-T. Chen, "Face recognition on drones: Issues and limitations," in *Proceedings of the first workshop on micro aerial vehicle networks, systems, and applications for civilian use*, 2015, pp. 39–44.

- [2] S. J. Russell, *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.
- [3] M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," *Machines*, vol. 11, no. 7, p. 677, 2023.
- [4] Mohit, "Guide to Yolov5 for Real-Time Object Detection." <https://analyticsindiamag.com/yolov5/> (accessed Aug. 04, 2023).
- [5] G. Jocher, "ultralytics/ultralytics: NEW - YOLOv8 in PyTorch > ONNX > OpenVINO > CoreML > TFLite." <https://github.com/ultralytics/ultralytics> (accessed Aug. 04, 2023).
- [6] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.

