

Integrasi Streamlit pada Aplikasi Berbasis Web dengan Algoritma YOLO V8 dan Teknologi Drone untuk Identifikasi Jenis dan Estimasi Tinggi Pohon

Alwi Zulfauzi Toscana
Computer Engineering, Faculty of
Electrical Engineering
Telkom University
Bandung, Indonesia
toscahr@student.telkomuniversity
.ac.id

Casi Setianingsih
Computer Engineering, Faculty of
Electrical Engineering
Telkom University
Bandung, Indonesia
setiacasie@telkomuniversity.ac.id

Marisa W. Paryasto
Electrical Engineering, Faculty of
Electrical Engineering
Telkom University
Bandung, Indonesia
marisa.paryasto@gmail.com

Abstrak — Penelitian ini mencerminkan penerapan drone dan kecerdasan buatan (AI), khususnya menggunakan algoritma YOLOv8, dalam mendeteksi jenis pohon dengan cepat dan efisien. Pohon memiliki peran penting dalam ekosistem dan perubahan iklim, serta dalam upaya konservasi hutan dan pengelolaan sumber daya alam. Dengan kemajuan teknologi drone, AI, dan computer vision, proses ini dapat dioptimalkan untuk menghemat waktu dan tenaga manusia. Penerapan drone memungkinkan akses ke daerah yang sulit dijangkau, sementara AI dengan algoritma YOLOv8 yang dioptimalkan untuk computer vision dapat secara otomatis mendeteksi dan mengidentifikasi pohon dalam gambar atau video. Meskipun tantangan utamanya adalah keterbatasan dataset untuk melatih model AI, perkembangan dalam teknologi drone, AI, dan computer vision membawa potensi besar untuk meningkatkan efisiensi dan akurasi dalam deteksi jenis pohon. Temuan dari penelitian ini menunjukkan bahwa penerapan drone dan algoritma YOLOv8 efektif dan akurat dalam mengukur dan mengklasifikasikan pohon tinggi. Model yang dibangun mencapai kinerja dengan nilai presisi sebesar 88,57%, recall 86,14%, mAP50 93,98%, dan mAP50-90 68,10%. Sistem yang dikembangkan juga memiliki akurasi yang cukup baik dengan skor kepercayaan rata-rata 87%. Teknologi ini memiliki potensi besar dalam mendukung berbagai aplikasi, termasuk pemantauan pertumbuhan hutan, konservasi sumber daya alam, serta penilaian dampak perubahan iklim terhadap ekosistem hutan.

Kata kunci— Algoritma YOLOv8, Deteksi Pohon, Drone, Kecerdasan Buatan (AI), Optimisasi Teknologi Drone

I. PENDAHULUAN

Pohon memiliki peran yang sangat vital tidak hanya dalam bidang kayu, tetapi juga dalam lingkaran karbon global dan perubahan iklim. Kehadiran pohon menjadi kunci bagi kelangsungan makhluk hidup, karena pohon adalah sumber kehidupan bagi semua makhluk.

Dalam era teknologi yang semakin maju, pemanfaatan dan pengaplikasian kendaraan udara tanpa awak (Unmanned Aerial Vehicle/UAV) atau drone menjadi pilihan yang membantu manusia dalam menjalani rutinitas sehari-hari.

UAV merupakan pesawat terbang yang beroperasi tanpa pilot dan bisa dikendalikan dari jarak jauh atau secara otomatis [1], [2]. Penggunaan drone terus berkembang karena kemampuannya menjangkau daerah yang sulit diakses dan menyediakan data secara real-time, yang mengubah paradigma kerja dan interaksi dengan lingkungan.

Dalam beberapa tahun terakhir, kemajuan dalam bidang Kecerdasan Buatan (Artificial Intelligence/AI) berkembang dengan pesat. AI merujuk pada kemampuan sistem komputer untuk meniru, memahami, dan menjalankan tugas yang umumnya memerlukan kecerdasan manusia. Ini 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].

Kolaborasi antara drone dan AI memiliki potensi besar dalam proses klasifikasi jenis pohon serta estimasi tingginya. Dengan memanfaatkan kedua teknologi ini, klasifikasi dan estimasi ketinggian pohon dapat dilakukan dengan lebih efisien dan akurat. Salah satu tantangannya adalah keterbatasan dataset yang diperlukan dalam pelatihan model. Mengingat hambatan dalam proses deteksi jenis pohon secara manual, dirancanglah sistem implementasi YOLOv8 pada deteksi jenis pohon menggunakan drone guna mengoptimalkan efisiensi dan akurasi proses tersebut.

II. KAJIAN TEORI

A. Algoritma YOLOv8

Pada tahun 2023, Glenn Jocher, pendiri dan CEO Ultralytics, merilis YOLOv8, sebuah algoritma deteksi objek terbaru dalam rangkaian YOLO. Sebagai inovasi terbaru dalam seri YOLO saat ini, YOLOv8 memiliki struktur jaringan yang terdiri dari beberapa lapisan, termasuk lapisan input, jaringan backbone, jaringan neck, dan lapisan output [1]. Dalam struktur jaringannya, YOLOv8 memanfaatkan beberapa jenis layer seperti Conv, C2f, Concat, Upsample, dan SPPF. Lapisan C2f dalam YOLOv8 mengandung CSPBottleneck yang melibatkan 3 proses konvolusi. Fungsi lapisan ini adalah untuk menjaga fitur melalui propagasi,

mendorong penggunaan kembali fitur jaringan, mengurangi jumlah parameter jaringan, serta membantu mempertahankan ciri-ciri yang halus untuk disampaikan ke lapisan yang lebih dalam dengan efisiensi yang lebih baik.

B. Streamlit

Streamlit adalah sebuah framework berbasis Python dan bersifat *open-source* yang dibuat untuk memudahkan dalam membangun aplikasi web di bidang sains data dan *machine learning* yang interaktif. Salah satu hal menarik dari framework ini adalah kita tidak perlu mengetahui banyak hal tentang teknologi *web development*. Kita tidak perlu dipusingkan tentang bagaimana mengatur tampilan *website* dengan CSS, HTML, atau Javascript. Untuk menggunakan Streamlit, kita cukup memiliki modal dasar mengetahui bahasa Python saja. Secara umum, Streamlit memiliki beberapa fitur yang cukup lengkap. Fitur-fitur tersebut dapat digunakan untuk melakukan kontrol terhadap aplikasi web kita seperti *slider*, *text input*, *checkbox*, *selectbox*, dan sebagainya [2].

III. METODE

A. Dataset Construction

Data yang dikumpulkan mencakup beragam variasi dan representatif ketiga kelas yaitu pisang, palm, dan pinus untuk menggambarkan berbagai skenario dan kondisi pada kehidupannya. Oleh karena itu data yang digunakan untuk sistem ini sebanyak 1600 data dengan pembagian data seimbang tiap kelasnya yaitu 400 data pisang, 400 data pinus, dan 400 data palm, serta 400 data null. Tiap data merepresentasikan kelasnya sedangkan data *null* digunakan untuk representasi ketika tidak ada objek yang diinginkan dalam frame pendeteksian. Analisis yang dilakukan adalah melakukan pengujian partisi data *training*, *validation* dan *test*. Pada implementasi sistem yang dikembangkan proses ini berjalan sesuai dengan yang diharapkan.

B. Training Model YOLOv8

Pada implementasi *training* model YOLOv8 diperlukan beberapa pengujian untuk mendapatkan model terbaik. Model terbaik harus mendapatkan nilai *performance* yang bagus. Nilai *performance* yang diukur antara lain *precision*, *recall*, mAP50, dan mAP50-95. Pada sistem yang dikembangkan untuk mendapatkan hasil *training* model YOLOv8 terbaik diperlukan *training* berulang ulang dengan mengubah parameter yang digunakan seperti partisi data, *batch size*, *learning rate*, dan juga *epoch*. Pada implementasinya proses ini berjalan sesuai dengan yang diharapkan, dari total 1600 dataset yang digunakan dibagi menjadi 80% data *train*, 10% data *validation*, dan 10% data *test*. Parameter yang digunakan adalah *batch size* 8, *learning rate* 0.001, dan *epoch* 100.

IV. HASIL DAN PEMBAHASAN

A. Proses Pengujian Website

Proses pengujian *website* akan dilakukan melalui metode pengujian *alpha* dan *beta*. Pada tahap pengujian *alpha*,

website akan diuji secara internal oleh pengembang untuk mengidentifikasi dan mengatasi potensi masalah atau *bug* sebelum dirilis kepada pengguna akhir. Pengujian ini akan melibatkan uji fungsionalitas, dan kinerja *website* secara mendalam. Setelah tahap pengujian *alpha* selesai dan perbaikan dilakukan, *website* akan masuk ke tahap pengujian *beta*. Pada tahap ini, *website* akan diuji oleh sekelompok pengguna yang mewakili pengguna akhir sesungguhnya. Tujuan dari pengujian *beta* adalah untuk mendapatkan umpan balik dari pengguna terhadap pengalaman penggunaan *website*, mengidentifikasi masalah atau perbaikan yang mungkin belum terdeteksi, serta memastikan kesesuaian dan kepuasan pengguna dalam penggunaan *website*. Kedua tahap pengujian ini akan memastikan bahwa *website* telah siap untuk diimplementasikan dan digunakan secara luas dengan kualitas yang optimal.

1. Pengujian alpha

Pengujian *Alpha* dilakukan sebelum sistem diuji oleh pengguna yang sebenarnya agar sistem nantinya dapat digunakan oleh pengguna dengan baik. Pengujian *alpha* dilakukan dengan metode *Black Box*. Pengujian ini difokuskan pada masukan dan keluaran dari sistem yang sedang diuji. Penguji membandingkan apakah keluaran sesuai dengan yang direncanakan sebelumnya. Pada Proses pengujian *website* akan dilakukan melalui metode pengujian *alpha* dan *beta*. Pada tahap pengujian *alpha*, *website* akan diuji secara internal oleh pengembang untuk mengidentifikasi dan mengatasi potensi masalah atau *bug* sebelum dirilis kepada pengguna akhir. Pengujian ini akan melibatkan uji fungsionalitas, dan kinerja *website* secara mendalam. Tabel 3.1 merupakan skenario pengujian *alpha*.

Tabel 3.1 Pengujian Alpha

No	Aksi yang Diuji	Detail Pengujian	Jenis Pengujian
1	Menggeser tombol slider <i>confidence threshold</i>	Pengujian dilakukan dengan mengatur nilai <i>confidence threshold</i> pada hasil outputnya.	<i>Black Box</i>
2	Menggeser tombol slider jarak <i>drone</i>	Pengujian dilakukan dengan mengatur jarak <i>drone</i> dari objek pada hasil outputnya.	<i>Black Box</i>
3	Menampilkan hasil deteksi pohon.	Pengujian dilakukan untuk memastikan bahwa hasil <i>output</i> tampil sesuai pada halaman.	<i>Black Box</i>
4	Menekan tombol <i>Stop</i>	Pengujian dilakukan untuk memeriksa hasil waktu saat program berjalan	<i>Black Box</i>

No	Aksi yang Diuji	Detail Pengujian	Jenis Pengujian
		dan tombol "Stop" ditekan.	

2. Pengujian Beta

Pengujian *Beta* dilakukan untuk mendapatkan tanggapan dan penilaian langsung dari pengguna yang sebenarnya mengenai sistem yang sudah dibuat. Penguji dalam pengujian ini adalah pengguna yang sebenarnya. Tabel 3.2 merupakan keterangan nilai skala likert yang digunakan.

Tabel 3.2 Keterangan skala likert

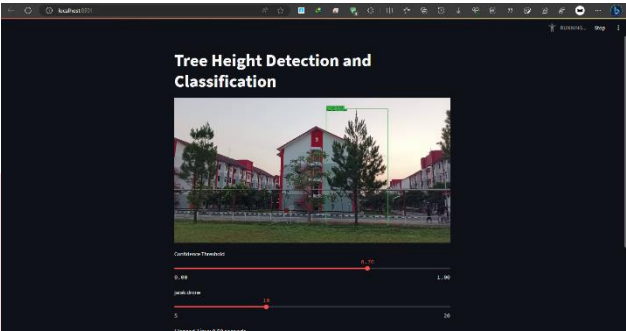
Kode	Keterangan
A	Sangat Setuju
B	Setuju
C	Netral
D	Tidak Setuju
E	Sangat Tidak Setuju

Tabel 3.3 Hasil pengujian beta

NO	Pertanyaan	Jumlah Responden				
		A	B	C	D	E
1	Website memiliki tampilan antar muka (user interface) yang menarik?	2	18	10	0	0
2	Apakah laporan atau output yang dihasilkan oleh aplikasi akurat dan sesuai dengan harapan?	7	15	5	3	0
3	Apakah antarmuka pengguna (UI) mudah dipahami dan nyaman digunakan?	8	10	10	2	0
4	Apakah aplikasi ini berhasil menjalankan fungsi utama sesuai dengan yang diharapkan?	7	15	5	3	0
5	Tidak terdapat kendala dalam penggunaan website secara keseluruhan?	8	14	4	4	0
6	Tulisan pada website dapat dibaca dan dipahami dengan mudah?	11	14	3	2	0
7	Website dapat menampilkan hasil data deteksi tinggi pohon dan jenis pohon	9	20	1	0	0

B. Output pada website

Setelah melalui proses *Object Detection* dan pengukuran tinggi, 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 1 adalah tampilan hasil akhir *output* pada *website*.



Gambar 1 Output pada website

C. KESIMPULAN

Penerapan framework Streamlit pada sistem pendeteksian jenis dan estimasi ketinggian pohon merupakan langkah penting dalam mengembangkan solusi yang efisien dan interaktif dalam mengelola dan menganalisis data terkait pohon. Dalam penelitian ini, penggunaan framework Streamlit telah berhasil memungkinkan integrasi yang lancar antara algoritma pendeteksian objek dan tampilan antarmuka pengguna. Kemampuan interaktif Streamlit memungkinkan pengguna untuk dengan cepat memasukkan input, seperti gambar atau data baru, dan mendapatkan hasil deteksi dan estimasi secara *real-time*. Dengan adanya fitur ini, sistem menjadi lebih adaptif dan dapat digunakan dalam berbagai konteks, seperti pemantauan lingkungan, manajemen hutan, atau penelitian ilmiah.

REFERENSI

[1] G. Jocher, “ultralytics/ultralytics: NEW - YOLOv8 in PyTorch > ONNX > OpenVINO > CoreML > TFLite.” <https://github.com/ultralytics/ultralytics> (accessed Aug. 04, 2023).

[2] R. Agarwal, “How to Write Web Apps Using Simple Python for Data Scientists - KDnuggets,” 2019. <https://www.kdnuggets.com/2019/10/write-web-apps-using-simple-python-data-scientists.html> (accessed Aug. 22, 2023).

