

Implementasi YOLOv10 untuk Deteksi Kerapatan dan Transparansi Tajuk Pohon melalui Aplikasi *Mobile*

Alkhadafi Saddam Simparico¹, Rico Andrian², Rahmat Safe'i³, Admi Syarif⁴

^{1,2,4}Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung

³Kehutanan, Fakultas Pertanian, Universitas Lampung

¹alkhadafi.saddam21@students.unila.ac.id, ²rico.andrian@fmipa.unila.ac.id, ³rahmat.safei@fp.unila.ac.id*,

⁴admi.syarif@fmipa.unila.ac.id

Abstract

Crown density and foliage transparency are key indicators of forest health that significantly influence ecosystem balance and biodiversity. This study develops a real-time detection system based on the YOLOv10 model, optimized for mobile devices through conversion to TensorFlow Lite, enabling fast and efficient inference in the field without the need for high-end computing hardware. The dataset comprises 5,000 canopy images across ten classes of varying density and transparency, representing five needle-leaf and five broadleaf species with distinct morphological characteristics and light transmission properties. Data collection was conducted from multiple viewing angles to enhance the model's robustness against real-world variability. The dataset was split into 70% for training, 10% for validation, and 20% for testing. Evaluation results show an accuracy of 97.7% with consistently high precision, recall, and F1-scores across all classes. This system has the potential to accelerate field surveys, improve ecosystem monitoring accuracy, and serve as a decision-support tool for forest management and conservation programs. The proposed approach offers a practical and scalable solution for sustainable forest monitoring by leveraging state-of-the-art computer vision technology on mobile devices.

Keywords: CNN, crown density, foliage transparency, YOLOv10, mobile app

Abstrak

Kerapatan dan transparansi tajuk pohon merupakan indikator penting kesehatan hutan yang berpengaruh terhadap keseimbangan ekosistem dan keanekaragaman hayati. Penelitian ini mengembangkan sistem deteksi *real-time* berbasis model YOLOv10 yang dioptimalkan untuk perangkat mobile melalui konversi ke *TensorFlow Lite*, sehingga memungkinkan inferensi cepat dan efisien di lapangan tanpa memerlukan perangkat komputasi besar. *Dataset* yang digunakan terdiri dari 5.000 citra tajuk pohon yang mencakup sepuluh kelas variasi kerapatan dan transparansi, mewakili lima jenis daun jarum dan lima jenis daun lebar dengan perbedaan morfologi dan karakteristik transmisi cahaya. Pengambilan data dilakukan pada berbagai sudut pandang untuk meningkatkan ketahanan model terhadap kondisi nyata di lapangan. Data dibagi menjadi 70% untuk pelatihan, 10% untuk validasi, dan 20% untuk pengujian. Hasil evaluasi menunjukkan akurasi 97,7% dengan nilai *precision*, *recall*, dan *F1-score* yang tinggi di setiap kelas. Sistem ini berpotensi mempercepat proses survei lapangan, meningkatkan akurasi pemantauan ekosistem, dan menjadi alat pendukung pengambilan keputusan dalam pengelolaan hutan serta program konservasi. Pendekatan ini menawarkan solusi praktis dan terukur untuk pemantauan hutan berkelanjutan dengan memanfaatkan teknologi *computer vision* mutakhir di perangkat mobile.

Kata kunci: CNN, kerapatan tajuk, transparansi tajuk, YOLOv10, aplikasi *mobile*

©This work is licensed under a Creative Commons Attribution - ShareAlike 4.0 International License

1. Pendahuluan

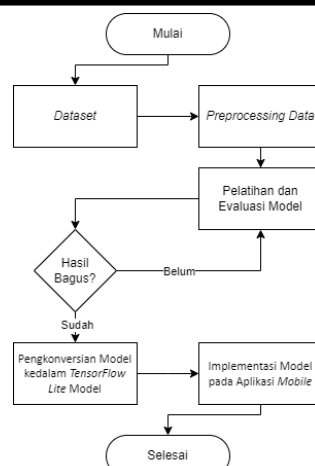
Indikator kesehatan hutan mencakup berbagai aspek yang berperan dalam menjaga keberlanjutan ekosistemnya. Beberapa parameter utama yang digunakan untuk menilai kesehatan hutan meliputi produktivitas, vitalitas, kualitas tapak, serta biodiversitas [1]. Vitalitas pohon merupakan indikator penting yang mencerminkan kemampuan pohon untuk bertahan hidup, tumbuh, dan beradaptasi terhadap tekanan lingkungan, sehingga berpengaruh langsung pada kestabilan ekosistem hutan secara keseluruhan. Vitalitas ini terdiri dari dua komponen utama, yaitu tingkat kerusakan pohon dan kondisi tajuk, yang saling berkaitan dalam menentukan kualitas fungsi ekologis pohon. Tingkat kerusakan pohon dapat memengaruhi luas, bentuk, dan kualitas tajuk, sementara kondisi

tajuk memberikan gambaran visual tentang kesehatan fisiologis pohon. Penelitian ini berfokus pada kondisi tajuk, yang dapat diukur melalui beberapa indikator, yaitu kerapatan tajuk pohon, transparansi tajuk pohon, diameter tajuk pohon, *live crown ratio*, dan mati pucuk. Kerapatan tajuk pohon (*crown density*) dan transparansi tajuk (*foliage transparency*) merupakan dua indikator utama yang banyak digunakan untuk menilai vitalitas dan kesehatan pohon, baik pada ekosistem perkotaan maupun hutan [2]. Pengukuran kerapatan dan transparansi tajuk pohon biasanya dilakukan secara manual menggunakan kartu skala transparansi. Metode ini cenderung kurang efisien dan rentan terhadap kesalahan manusia. Solusi yang lebih optimal adalah memanfaatkan teknologi pemrosesan citra berbasis kecerdasan buatan, yang dapat memberikan hasil lebih cepat, akurat, dan konsisten.

Pemrosesan citra digital mencakup berbagai algoritma, salah satunya adalah YOLO (*You Only Look Once*), yang merupakan arsitektur dari *Convolutional Neural Network* (CNN). YOLO dikenal sebagai salah satu metode deteksi objek paling populer karena efisiensinya yang tinggi, kecepatan pemrosesan yang cepat, serta kemampuannya mendeteksi banyak objek dalam satu gambar hanya melalui satu kali pemindaian. Pendekatan ini berbeda dari metode tradisional yang memerlukan pemindaian berulang, yang umumnya membutuhkan waktu lebih lama dan sumber daya komputasi lebih besar. Pada penelitian ini digunakan YOLOv10, salah satu versi terbaru dari keluarga YOLO, yang dirilis dengan sejumlah peningkatan dibandingkan pendahulunya, termasuk optimasi arsitektur untuk efisiensi komputasi, peningkatan akurasi deteksi, serta kemampuan generalisasi yang lebih baik pada beragam kondisi pencahayaan dan sudut pandang [3]. Pemilihan YOLOv10 dilakukan untuk memanfaatkan keunggulan teknologi terkini yang relevan dengan kebutuhan sistem deteksi objek *real-time* di perangkat *mobile*, sehingga diharapkan dapat memberikan hasil yang lebih optimal dibandingkan versi sebelumnya [4]. Penelitian terdahulu terkait penggunaan arsitektur CNN dipakai untuk mengidentifikasi kerapatan dan transparansi daun jarum menggunakan arsitektur VGG16 menghasilkan akurasi diatas 92% untuk setiap kelasnya [5]. Penelitian terkait lainnya dalam penggunaan arsitektur CNN untuk mengidentifikasi kerapatan dan transparansi daun lebar menggunakan arsitektur MobileNet menghasilkan akurasi diatas 87% untuk setiap kelasnya [6]. Penelitian ini akan menggunakan arsitektur YOLOv10 sebagai model utama. 5000 citra akan digunakan dalam penelitian ini yang terbagi menjadi sepuluh kelas dengan 500 citra untuk setiap kelasnya yang terbagi berdasarkan kerapatan dan transparansi dalam rentang 5% sampai dengan 95%. *Dataset* yang digunakan terdiri dari sepuluh jenis pohon yang terbagi menjadi dua berdasarkan bentuk daun-nya yaitu daun lebar dan daun jarum. Sepuluh jenis pohon yang digunakan yaitu Cemara Norfolk (*Araucaria heterophylla*), Cemara Bundel (*Cupressus retusa*), Tusam Sumatra (*Pinus merkusii*), Damar Mata-Kucing (*Shorea javanica*), Damar Biasa (*Agathis dammara*), Cokelat (*Theobroma cacao*), Durian (*Durio zibethinus*), Karet (*Havea brasiliensis*), Kemiri (*Aleurites moluccana*) dan Sonokeling (*Dalbergia latifolia*).

2. Metode Penelitian

Tahapan dalam penelitian ini akan dijelaskan menggunakan diagram alir mengenai langkah-langkah yang akan dilakukan selama penelitian yang terdapat pada Gambar 1.



Gambar 1. Metodologi Penelitian

2.1. Dataset

Dataset digunakan agar mesin dapat mempelajari pola visual dari setiap kelas. *Dataset* ini berasal dari penelitian terdahulu dan mencakup dua tipe morfologi utama pada ekosistem hutan tropis, yaitu daun jarum dengan kerapatan tajuk tinggi dan transmisi cahaya rendah, serta daun lebar yang memiliki variasi transparansi lebih besar. Pendekatan ini memungkinkan model mempelajari perbedaan struktural dan tekstural antar tipe vegetasi, sehingga relevan untuk pemantauan hutan secara umum. *Dataset* berisi 5.000 citra terbagi dalam 10 kelas, seperti terlihat pada Tabel 1.

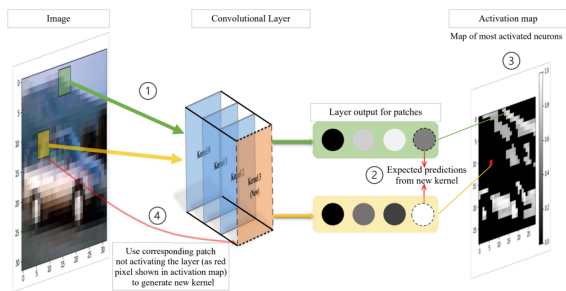
Tabel 1. *Dataset*

No	Nama Kelas	Contoh Citra	No	Nama Kelas	Contoh Citra
1	CD 5% FT 95%		6	CD 55% FT 45%	
2	CD 15% FT 85%		7	CD 65% FT 35%	
3	CD 25% FT 75%		8	CD 75% FT 25%	
4	CD 35% FT 65%		9	CD 85% FT 15%	
5	CD 45% FT 55%		10	CD 95% FT 5%	

2.2. Convolutional Neural Network

Algoritma *deep learning* memiliki berbagai varian yang dirancang untuk menangani beragam permasalahan. *Convolutional Neural Networks* (CNN) merupakan salah satu yang paling umum digunakan

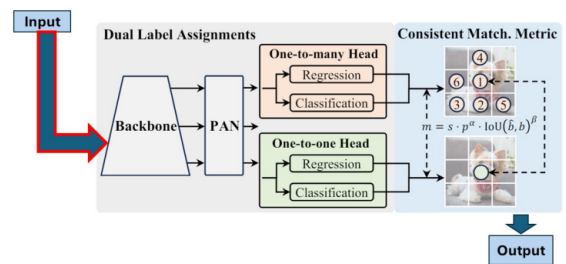
dalam pemrosesan citra dan video [7]. CNN adalah jenis jaringan saraf tiruan yang dirancang khusus untuk memproses data berbentuk gambar dengan menganalisis nilai piksel. Jaringan ini terdiri dari beberapa lapisan tersembunyi yang bertugas mengenali berbagai karakteristik dalam gambar. CNN mendeteksi fitur dasar seperti garis vertikal dan horizontal. Hasil dari lapisan pertama ini kemudian digunakan untuk mengenali pola yang lebih kompleks di lapisan berikutnya. Seiring dengan bertambahnya kedalaman lapisan, CNN mampu mengidentifikasi karakteristik yang lebih rumit, seperti warna dan bentuk, sehingga meningkatkan akurasi dalam pengenalan objek [8]. Arsitektur CNN umumnya terdiri dari beberapa lapisan utama yaitu *Convolutional Layer*, *Pooling Layer*, dan *Fully Connected Layer* [9]. Ilustrasi dari CNN dapat dilihat di Gambar 2.



Gambar 2. Arsitektur CNN

2.3. YOLOv10

YOLOv10 adalah salah satu dari versi terbaru dari rangkaian model deteksi objek YOLO yang terus mengalami pengembangan dari versi sebelumnya, seperti YOLOv8, hingga YOLOv9 [10]. Fitur utama YOLOv10 adalah kemampuannya dalam melakukan deteksi objek secara *end-to-end* tanpa memerlukan langkah pasca-pemrosesan seperti *non-maximum suppression* (NMS) hal ini memungkinkan YOLOv10 mengurangi latensi, sehingga sangat efektif untuk aplikasi *real-time*. Ilustrasi dari arsitektur YOLOv10 dapat dilihat di Gambar 3.



Gambar 3. Arsitektur YOLOv10

2.4. Preprocessing Data

Preprocessing Data adalah langkah awal yang penting dalam pelatihan, di mana data yang mentah dibersihkan dan diolah agar sesuai dengan kebutuhan model [11]. Tahapan ini meliputi pemberian label pada data dan *resizing* citra ke ukuran 224x224 piksel. Selain itu, dilakukan teknik augmentasi data seperti *flip*, *rotate*, dan *zoom* untuk memperkaya variasi *dataset*.

Augmentasi ini bertujuan meningkatkan kemampuan model dalam mengenali berbagai variasi data dan kondisi nyata di lapangan, sehingga dapat meningkatkan akurasi dan generalisasi model. Setelah itu, data dibagi ke dalam tiga subset, yaitu *training* (70%), *validation* (10%), dan *testing* (20%) dari total data.

2.5. Pelatihan Model

Pelatihan model dilakukan melalui delapan percobaan untuk menemukan formula yang tepat dan konfigurasi *hyperparameter* terbaik. *Epoch* sebanyak 100 digunakan berdasarkan pengujian awal dengan 10 *epoch* yang memberikan hasil kurang optimal. Peningkatan epoch ini bertujuan menguji apakah performa model dapat meningkat sesuai harapan. *Batch-size* 8 dipilih untuk mempercepat proses pelatihan tanpa mengorbankan stabilitas konvergensi. *Optimizer Stochastic Gradient Descent* (SGD) digunakan karena karakteristik objek yang dideteksi bersifat abstrak dan beragam, dimana setiap kelas memiliki variasi unik dalam pola visualnya. SGD lebih cocok dalam mengoptimalkan model untuk kasus ini dibandingkan Adam, yang lebih efektif pada objek dengan pola yang relatif konsisten, seperti motif batik, ukiran, wajah, atau benda dengan tekstur seragam. Pengaturan *learning-rate* didasarkan pada pengamatan empiris selama eksperimen, yang menunjukkan kestabilan dan performa optimal model. Kombinasi *hyperparameter* ini dipilih setelah serangkaian eksperimen untuk memastikan konvergensi yang baik dan akurasi tinggi dalam klasifikasi. Eksperimen *hyperparameter* lengkap dapat dilihat pada Tabel 2.

Tabel 2. Pelatihan Model

No	Jumlah Percobaan	Epoch	Batch-size	Optimizer	Learning-rate	Automatic Mixed Precision (AMP)
1	Pertama	10	32	SGD	0.01	True
2	Ke-dua	100	8	SGD	0.01	True
3	Ke-tiga	100	16	SGD	0.01	True
4	Ke-empat	100	8	Adam	0.001	False
5	Ke-lima	100	8	SGD	0.001	False
6	Ke-enam	100	8	Adam	0.01	False
						W
7	Ke-tujuh	100	8	SGD	0.01	False
8	Ke-delapan	100	8	SGD	0.001	True

2.6. Evaluasi Model

Evaluasi model yang akan dilakukan dalam penelitian ini adalah dengan menggunakan *Confusion Matrix*. *Confusion Matrix* berperan dalam memberikan gambaran visual tentang performa model klasifikasi dengan menunjukkan seberapa baik model dalam memprediksi setiap kelas. *Confusion Matrix* membantu mengidentifikasi kesalahan prediksi serta mengukur akurasi model dalam membedakan kelas objek. *Confusion Matrix* menampilkan jumlah prediksi yang benar dan salah untuk setiap kelas serta menyediakan

informasi penting terkait ukuran evaluasi seperti *accuracy* adalah ukuran kinerja yang menunjukkan proporsi prediksi yang benar dari keseluruhan prediksi (1), *precision* mengukur seberapa banyak kategori yang ditemukan oleh sistem benar-benar relevan dibandingkan dengan total kategori yang ditemukan (2), *recall* mengukur seberapa banyak kategori yang relevan atau benar ditemukan oleh sistem dibandingkan dengan total kategori yang sebenarnya benar (3), dan *F1-Score* adalah ukuran kinerja yang digunakan untuk menilai kualitas suatu sistem klasifikasi, terutama ketika ada ketidakseimbangan antara kelas positif dan negatif (4) [12]. Confusion Matrix terdiri dari empat komponen utama yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) [13]. Menghadapi potensi ketidakseimbangan kelas dalam *dataset*, penelitian ini menerapkan teknik *undersampling* untuk memastikan bahwa setiap kelas memiliki jumlah citra yang seimbang. Pendekatan ini bertujuan untuk mencegah model *bias* terhadap kelas mayoritas dan meningkatkan kemampuan generalisasi model terhadap semua kelas. Dengan demikian, metrik evaluasi yang diperoleh menjadi lebih representatif dan sensitif terhadap performa pada kelas minoritas maupun mayoritas.

Tabel 3. Nilai Matriks Utama dari *Confusion Matrix*

Nilai Prediksi	Nilai Sebenarnya		
	<i>Positive</i> <i>Negative</i>	<i>Positive</i> TP FN	<i>Negative</i> FP TN

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

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

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

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

2.7. Pengkonversian Model kedalam *TensorFlow Lite Model*

Tahapan selanjutnya adalah pengkonversian model kedalam *TensorFlow Lite Model*. Tujuan utama dari penggunaan *TensorFlow Lite* adalah untuk memungkinkan proses inferensi yang cepat dan hemat sumber daya, sehingga model dapat bekerja secara *real-time* tanpa ketergantungan pada koneksi internet atau server *cloud* [14]. Konversi model YOLOv10 ke *TensorFlow Lite* memberikan sejumlah keuntungan signifikan untuk aplikasi *mobile*, terutama yang memerlukan kinerja *real-time*. Keuntungan tersebut mencakup pengurangan latensi inferensi, efisiensi penggunaan memori, serta peningkatan responsivitas sistem dalam memproses data secara langsung di perangkat. *TensorFlow Lite* juga dirancang khusus untuk memaksimalkan efisiensi komputasi pada perangkat dengan sumber daya terbatas, sehingga model dapat berjalan dengan cepat dan hemat daya tanpa memerlukan koneksi ke server eksternal. Namun, proses konversi menghadirkan tantangan tersendiri, seperti mengoptimalkan ukuran model agar tetap

ringan tanpa mengorbankan akurasi, memastikan kompatibilitas dengan arsitektur *hardware mobile*, serta mengatasi potensi penurunan performa saat menjalankan deteksi secara *real-time*. Dengan mengatasi tantangan ini, model hasil konversi dapat digunakan langsung di perangkat *mobile* sebagai sistem deteksi yang andal di lapangan.

2.8. Implementasi Model pada Aplikasi *Mobile*

Tahapan ini akan membuat aplikasi *mobile* yang dapat menjalankan model dari *TensorFlow Lite*. Pengembangan dari aplikasi *mobile* ini akan menggunakan bahasa pemrograman Kotlin. Kotlin merupakan bahasa pemrograman modern yang banyak digunakan dalam pengembangan *Android* karena kemampuannya untuk berinteroperasi dengan Java serta efisiensinya dalam perangkat *mobile* [15].

Dengan memanfaatkan YOLOv10 sebagai algoritma deteksi objek yang cepat dan akurat, serta *TensorFlow Lite* untuk memungkinkan inferensi secara *real-time* di perangkat *mobile*, aplikasi ini dirancang untuk secara praktis dan langsung mengidentifikasi kerapatan dan transparansi tajuk pohon. Tahap akhir penelitian meliputi pengujian aplikasi untuk memastikan bahwa prediksi model terhadap gambar tajuk pohon yang diambil melalui aplikasi sesuai dengan kondisi nyata di lapangan. Tujuan utama dari pengembangan ini adalah menghadirkan solusi praktis dan efisien yang dapat digunakan petugas kehutanan dan peneliti dalam memantau kesehatan hutan secara *real-time* tanpa bergantung pada perangkat komputasi berat atau koneksi internet.

3. Hasil dan Pembahasan

Bagian ini akan membahas secara rinci mengenai hasil yang sudah diperoleh selama penelitian dan pengembangan aplikasi *mobile*. Hasil pelatihan dengan performa terbaik juga akan ditampilkan untuk memberikan gambaran jelas mengenai pelatihan yang telah dilakukan menggunakan Arsitektur YOLOv10. Bagian terakhir akan menjelaskan hasil model *TensorFlow Lite* yang telah digunakan didalam aplikasi *mobile* yang telah dikembangkan.

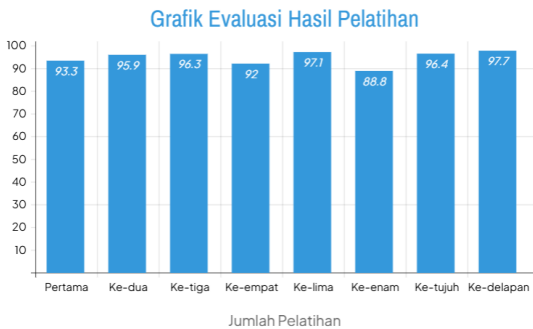
3.1. Evaluasi Hasil Pelatihan

Model dari arsitektur YOLOv10 yang telah dilatih memiliki hasil yang beragam dikarenakan parameter pada *Hyperparameter* yang digunakan berbeda-beda. Hasil dari semua pelatihan dapat dilihat pada Tabel 4.

Tabel 4. Evaluasi Hasil Pelatihan

N o	Jumla h Perco baan	Ep och	Bat ch- size	Optim izer	Learn ing- rate	Auto matic Mixed Precis ion (AMP)	Accu racy
1	Perta ma	10	32	SGD	0.01	True	93.3 %

No	Jumlah Percoobaan	Epoch	Batch-size	Optimizer	Learning-rate	Automatic Mixed Precision (AMP)	Accuracy
2	Ke-dua	100	8	SGD	0.01	True	95.9 %
3	Ke-tiga	100	16	SGD	0.01	True	96.3 %
4	Ke-empat	100	8	Adam	0.001	False	92%
5	Ke-lima	100	8	SGD	0.001	False	97.1 %
6	Ke-enam	100	8	Adam W	0.01	False	88.8 %
7	Ke-tujuh	100	8	SGD	0.01	False	96.4 %
8	Ke-delapan	100	8	SGD	0.001	True	97.7 %



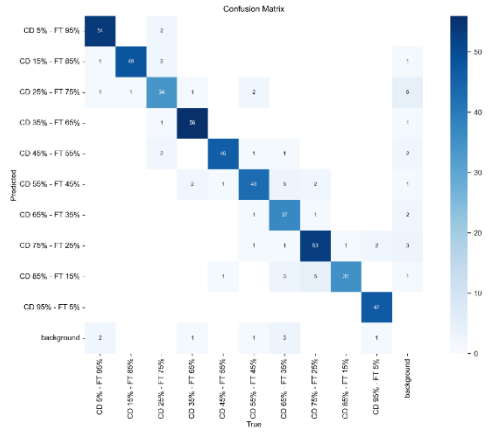
Gambar 4. Grafik evaluasi hasil pelatihan

Setelah menganalisis hasil pelatihan dan melihat grafik pada gambar 4, peneliti menyimpulkan bahwa pelatihan ke-delapan dengan parameter seperti 100 epoch, 8 batch-size, SGD optimzier, 0,001 learning-rate, dan True pada Automatic Mixed Precision (AMP) merupakan hasil pelatihan terbaik dengan menghasilkan akurasi sebesar 97.7%. Pemilihan epoch sebanyak 100 memungkinkan model belajar lebih mendalam dan mengurangi underfitting yang terjadi pada percobaan dengan epoch lebih rendah. Batch size 8 dipilih untuk menjaga keseimbangan antara kecepatan pelatihan dan stabilitas konvergensi model. Penggunaan optimizer SGD mendukung optimasi pada data dengan karakteristik objek yang abstrak dan variatif, sehingga meningkatkan kemampuan model dalam mengenali pola yang kompleks. Learning rate 0,001 merupakan nilai yang cukup kecil untuk memastikan proses pembaruan bobot model berjalan stabil tanpa menyebabkan lonjakan besar yang dapat mengganggu konvergensi. Hasil lengkap dari pelatihan ke-delapan dapat dilihat pada Tabel 5 dan *Confusion Matrix* dari pelatihan ke-delapan dapat dilihat pada Gambar 5.

Tabel 5. Hasil Pelatihan ke-delapan

No	Skala Kerapatan dan Transparansi tajuk pohon	Hasil		
		Precision	Recall	F1-score
1	CD 5% - FT 95%	93.1%	96.4%	94.7%
2	CD 15% - FT 85%	98%	92.3%	95%

3	CD 25% - FT 75%	82.9%	77.3%	80%
4	CD 35% - FT 65%	93.3%	96.6%	94.9%
5	CD 45% - FT 55%	93.9%	88.5%	91.1%
6	CD 55% - FT 45%	87.8%	82.7%	85.1%
7	CD 65% - FT 35%	77.1%	90.2%	83.1%
8	CD 75% - FT 25%	86.9%	85.5%	86.2%
9	CD 85% - FT 15%	97.2%	78.8%	86.4%
10	CD 95% - FT 5%	94%	100%	96.9%
Accuracy		97.7%		



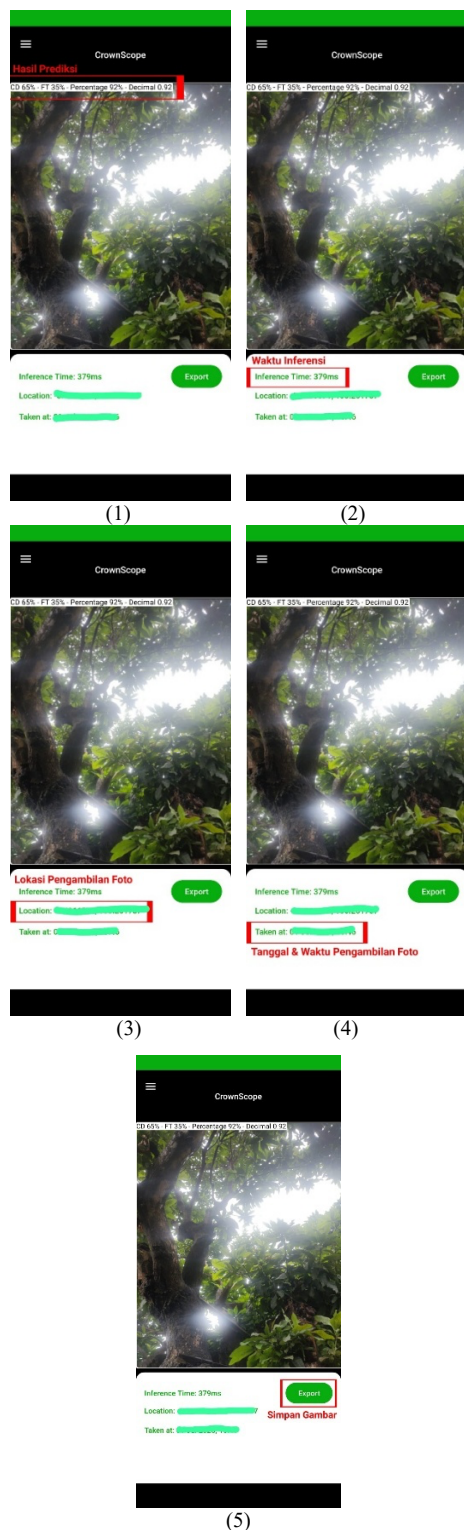
Gambar 5. Confusion Matrix dari Pelatihan ke-delapan

Berdasarkan Gambar 5 terlihat *Confusion Matrix* yang dihasilkan menunjukkan hasil yang cukup baik dengan hanya beberapa kesalahan yang minim dikarenakan kemiripan pola dari beberapa kelas.

3.2. Identifikasi Kerapatan dan Transparansi Tajuk Pohon pada Aplikasi *Mobile*

Tahapan selanjutnya yang akan dilakukan adalah pengkonversian model YOLOv10 yang sudah dipilih menjadi *TensorFlow Lite Model* dan dilanjutkan dengan implementasi pada aplikasi *mobile* yang telah dikembangkan dengan bahasa pemrograman *Kotlin*. Aplikasi *mobile* yang dikembangkan bertujuan untuk mengidentifikasi kerapatan dan transparansi tajuk melalui gambar yang langsung diambil dari aplikasi tersebut secara *real-time*. Aplikasi ini bertujuan untuk mempermudah petugas kehutanan, mahasiswa, dan dosen yang terlibat langsung dalam pemeriksaan kondisi hutan agar dapat melakukan pemantauan lebih cepat dan praktis di lapangan tanpa memerlukan peralatan rumit. Mahasiswa dan dosen kehutanan juga dapat menggunakan aplikasi ini sebagai alat bantu riset yang efisien dan akurat. Hasil dari prediksi model terhadap kasus nyata akan ditampilkan didalam aplikasi dalam bentuk kelas yang diidentifikasi oleh model. Selain menampilkan kelas pada kerapatan dan transparansi tajuk pohon, aplikasi juga akan menampilkan koordinat dimana foto tersebut diambil sehingga pengguna dapat tahu dimana letak foto tersebut diambil melalui koordinat yang tersedia. Adapun fitur lainnya pengguna dapat menyimpan gambar hasil prediksi kedalam perangkat *mobile* sehingga hasil-hasil dari identifikasi dapat dilihat kembali sebagai bahan evaluasi dan pembelajaran. Beberapa fitur yang sudah dijelaskan sebelumnya dapat dilihat di Gambar 6. Hasil prediksi ditampilkan diatas

gambar dalam bentuk persen dan desimal (1). *Inference Time* adalah waktu yang dibutuhkan perangkat *mobile* dalam mengenali objek yang sedang diidentifikasi (2). *Location* menampilkan koordinat dari lokasi foto tersebut diambil (3). *Taken At* berfungsi untuk menampilkan tanggal dan waktu dari saat foto tersebut diambil (4). *Export* berfungsi untuk menyimpan gambar hasil dari prediksi (5).



Gambar 6. Fitur pada Aplikasi

3.3. Pengujian Sistem

Pengujian sistem atau *User Acceptance Testing* (UAT) merupakan tahap validasi perangkat lunak oleh pengguna akhir di lingkungan operasional sebenarnya [16]. Tujuan dari UAT adalah untuk memastikan bahwa perangkat lunak yang dikembangkan benar-benar memenuhi kebutuhan dan harapan pengguna akhir. Evaluasi ini mencakup beberapa aspek utama, yaitu fungsionalitas, reliabilitas, kemudahan penggunaan (*usability*), serta efisiensi operasional. UAT menjadi tahap penting dalam proses pengembangan perangkat lunak karena memberikan kesempatan bagi pengguna dari kalangan yang relevan untuk terlibat secara langsung dalam proses penilaian sebelum aplikasi diluncurkan secara resmi. Melalui keterlibatan ini, berbagai ketidaksesuaian, kekurangan, maupun potensi permasalahan yang mungkin tidak terdeteksi selama tahap pengembangan internal dapat diidentifikasi dan diperbaiki lebih awal, sehingga dapat meningkatkan kualitas serta keandalan aplikasi ketika digunakan di dunia nyata.

Proses pengujian dilakukan secara bertahap, dimulai dari *internal testing* yang dilaksanakan melalui platform *Google Play Store*. Pada tahap awal ini dilakukan *alpha testing* oleh tim pengembang segera setelah aplikasi di-deploy dalam *closed environment*. Pengujian ini bertujuan untuk menilai stabilitas dan fungsionalitas dasar aplikasi dari sisi teknis dan apabila tidak ditemukan masalah signifikan, proses dilanjutkan ke tahap *closed testing*. Pada tahap ini, pengujian aplikasi dilakukan oleh 12 responden yang diminta untuk menggunakan aplikasi secara aktif selama 14 hari berturut-turut. Tujuan pengujian ini adalah untuk mengevaluasi fungsionalitas aplikasi, memastikan kelancaran operasional, serta mendeteksi adanya bug atau masalah teknis yang mungkin muncul selama penggunaan. Responden diminta untuk memberikan *feedback* secara sistematis melalui fitur *review* di *Google Play Store*, termasuk memberikan *rating* bintang dan komentar mengenai pengalaman penggunaan aplikasi. *Feedback* ini kemudian dianalisis untuk menilai efektivitas aplikasi dalam kondisi penggunaan sehari-hari serta mengidentifikasi area yang perlu diperbaiki sebelum aplikasi dirilis secara luas. Pendekatan ini memastikan pengujian yang komprehensif dan relevan dengan kebutuhan pengguna nyata di lapangan. Setelah menyelesaikan *closed testing* dilakukan pengujian terakhir oleh pengguna langsung dari aplikasi ini yaitu para mahasiswa kehutanan dan para praktisi kehutanan. Sebanyak 15 responden dilibatkan, yang terdiri atas 6 mahasiswa dari jurusan kehutanan serta 9 praktisi kehutanan yang memiliki pengalaman langsung di lapangan. Melalui pendekatan UAT, masukan dari para pengguna ini digunakan untuk menilai sejauh mana aplikasi mampu menjawab kebutuhan mereka secara praktis dan aplikatif, sebelum akhirnya diputuskan untuk dirilis ke publik secara luas. Detail lengkap mengenai hasil pengujian terakhir dapat dilihat pada Tabel 6.

Tabel 6. Hasil Pengujian

No	Pertanyaan	SS	S	C	TS	ST	N S *	P
1	Aplikasi mudah dipasang dari <i>Google play store (install)</i>	6	8	1	0	0	65	86.6 %
2	Fungsionalitas aplikasi bekerja sesuai harapan	2	9	4	0	0	58	77.3 %
3	Mudah mengakses fitur yang disediakan	5	7	2	0	0	64	85.3 %
4	Tampilan halaman dapat dengan mudah dimengerti	7	4	4	0	0	63	84%
5	Tampilan konsisten di seluruh aplikasi	2	10	3	0	0	59	78.6 %
6	Tampilan warna nyaman secara visual	4	4	6	1	0	56	74.6 %
7	Aplikasi memberikan pengalaman yang baik	1	11	3	0	0	58	77.3 %
8	Membuat akun dengan mudah	5	9	1	0	0	64	85.3 %
9	Proses <i>login</i> berjalan dengan lancar	6	7	2	0	0	64	85.3 %
10	Semua fitur berjalan sesuai dengan harapan	2	10	3	0	0	59	78.6 %
11	Berhasil mendapatkan data kerapatan dan transparansi tajuk pohon yang di uji	1	6	8			53	70.6
12	Berhasil menyimpan hasil pemindaian ke dalam <i>gallery</i>	6	4	5	0	0	61	81.3 %

Keterangan:

SS = Sangat Setuju (5 poin)
S = Setuju (4 poin)
C = Cukup (3 poin)
TS = Tidak Setuju (2 poin)

STS = Sangat Tidak Setuju (1 poin)

N = Nilai dari Setiap Jawaban

R = Jumlah Responden

$P = ((N * R) / \text{Total Skor Maksimum}) * 100\%$

Setiap pertanyaan dalam pengujian dilengkapi dengan 6 pilihan jawaban yang memiliki bobot penilaian berbeda-beda. Skor yang diperoleh dari masing-masing pertanyaan kemudian dikalikan dengan jumlah total responden. Hasil perkalian ini selanjutnya dibagi dengan *total skor maksimum* yang mungkin dicapai, dan nilai akhir dikalikan dengan 100% untuk memperoleh persentase tingkat kepuasan atau efektivitas berdasarkan setiap pertanyaan.

Analisis hasil UAT menunjukkan bahwa aplikasi mudah dipasang melalui *Google Play Store*, berkat ukuran file yang pas dan pengemasan yang sesuai standar *platform*, sehingga proses instalasi berjalan lancar tanpa hambatan. Dari segi fungsionalitas, seluruh fitur aplikasi, seperti pemindaian tajuk pohon, penyimpanan hasil ke galeri, serta proses pembuatan akun dan login, berjalan dengan baik tanpa kendala teknis. Navigasi dan akses ke berbagai fitur juga sangat mudah, berkat desain antarmuka yang sederhana, konsisten, dan intuitif. Penggunaan warna yang nyaman secara visual membantu mengurangi kelelahan mata dan meningkatkan keterbacaan selama penggunaan aplikasi. Selain itu, hasil prediksi kerapatan dan transparansi tajuk pohon yang dihasilkan dinilai memadai dan konsisten, mendukung kebutuhan pengambilan keputusan di lapangan.

Berdasarkan hasil pengujian, pertanyaan dengan persentase tertinggi adalah “Apakah aplikasi mudah dipasang dari *Google Play Store (install)*?” yang mencapai skor 86.6%, menandakan kemudahan instalasi aplikasi yang sangat baik di mata pengguna. Sebaliknya, nilai persentase terendah diperoleh pada pertanyaan “Apakah berhasil mendapatkan data kerapatan dan transparansi tajuk pohon yang diuji?”, dengan capaian sebesar 70.6%. Adapun rata-rata keseluruhan dari seluruh persentase nilai tiap pertanyaan menunjukkan angka 80.4%, yang mencerminkan tingkat penerimaan dan kepuasan pengguna terhadap aplikasi berada pada kategori baik secara umum.

4. Kesimpulan

Penelitian ini membuktikan bahwa pemanfaatan YOLOv10 untuk mengidentifikasi kerapatan dan transparansi tajuk pohon berbasis aplikasi *mobile* menghasilkan performa sangat baik dengan akurasi mencapai 97.7%. Berbeda dengan penelitian sebelumnya yang sebatas penelitian arsitektur CNN untuk identifikasi citra secara teori, penelitian ini mengambil langkah lebih nyata dengan mengonversi model hasil pelatihan menjadi model yang dapat digunakan pada aplikasi *mobile* yang dapat digunakan langsung di lapangan. Proses penelitian meliputi pengumpulan data, *preprocessing*, pelatihan model, hingga konversi ke *TensorFlow Lite* yang

memungkinkan inferensi cepat pada perangkat *mobile*. Aplikasi ini memudahkan petugas kehutanan, mahasiswa, dan praktisi dalam memantau kondisi hutan secara praktis tanpa metode konvensional yang memakan waktu dan tenaga. Fitur seperti pengenalan kelas kerapatan dan transparansi tajuk, waktu inferensi, serta lokasi pengambilan gambar memberikan nilai tambah signifikan untuk pemantauan lapangan. Penelitian ini berpotensi mendukung konservasi, riset ekologi, dan pemantauan lingkungan dengan lebih efisien dan akurat. Ke depan, pengembangan dapat difokuskan pada perluasan jenis pohon dan variasi kondisi lingkungan guna meningkatkan kualitas pengelolaan hutan.

Daftar Rujukan

- [1] Safe'i, R. (2018). PENILAIAN VITALITASPOHONJATI DENGANFOREST HEALTH MONITORINGDIKPH BALAPULANG. *Jurnal Ecogreen*. <http://repository.lppm.unila.ac.id/id/eprint/8288>
- [2] He, L., Zhang, X., Wang, X., Ullah, H., Liu, Y., & Duan, J. (2023). Tree Crown Affects Biomass Allocation and Its Response to Site Conditions and the Density of *Platycladus orientalis* Linnaeus Plantation. In *Forests*. mdpi.com. <https://www.mdpi.com/1999-4907/14/12/2433>
- [3] Gao, C., Zhang, Q., Tan, Z., Zhao, G., Gao, S., Kim, E., & Shen, T. (2024). Applying optimized YOLOv8 for heritage conservation: enhanced object detection in Jiangnan traditional private gardens. In *Heritage Science* (Vol. 12, Issue 1). Springer. <https://doi.org/10.1186/s40494-024-01144-1>
- [4] da Silva, D. de A., de Freitas, E. D. G., Abud, H. F., & Gomes, D. G. (2024). Applying YOLOv8 and X-ray Morphology Analysis to Assess the Vigor of *Brachiaria brizantha* cv. Xaraés Seeds. In *AgriEngineering* (Vol. 6, Issue 2, pp. 869–880). mdpi.com. <https://doi.org/10.3390/agriengineering6020050>
- [5] Andrian, R., Satriatna, D. A., & Tarigan, F. R. (2024). Classification of Density and Transparency of Needle Leaves Types Using AlexNet and VGG16 Architecture. ... *Des Systèmes d'...* <https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=16331311&AN=178256174&h=7xxYtu0iTV8yiGaXvZTdhXYdG1qnXH9RT0%2B180w3teT7m60ZyKfs9BB4isl8trQ4vQ4jkwOkHL0wQw%2B0mSZDIw%3D%3D&url=c>
- [6] Sofiyana, F., Andrian, R., & Safe'i, R. (2023). MobileNet untuk Identifikasi Skala Kerapatan dan Transparansi Tajuk Pohon Daun Lebar. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, 4(3), 1850–1859. <http://djournals.com/klik/article/view/1476>
- [7] Shiri, F. M., Perumal, T., Mustapha, N., & ... (2023). A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. *ArXiv Preprint ArXiv* <https://arxiv.org/abs/2305.17473>
- [8] Zhu, Y., & Chen, Y. (2023). Adaptive Growth: Real-time CNN Layer Expansion. *ArXiv Preprint ArXiv:2309.03049*. <https://arxiv.org/abs/2309.03049>
- [9] Liu, Y., Pu, H., & Sun, D. W. (2021). Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. *Trends in Food Science & Technology*. <https://www.sciencedirect.com/science/article/pii/S0924224421003022>
- [10] Vijayakumar, A., Vairavasundaram, S., Koilraj, J. A. S., & ... (2024). Real-time visual intelligence for defect detection in pharmaceutical packaging. In *Scientific Reports*. nature.com. <https://www.nature.com/articles/s41598-024-69701-z>
- [11] Long, J., Wang, X., Zhou, W., Zhang, J., Dai, D., & Zhu, G. (2021). A comprehensive review of signal processing and machine learning technologies for UHF PD detection and diagnosis (I): Preprocessing and localization approaches. *IEEE Access*. <https://ieeexplore.ieee.org/abstract/document/9422716>
- [12] Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*. <https://doi.org/10.1023/A:1009982220290>
- [13] Maxwell, A. E., Warner, T. A., & Guillén, L. A. (2021). Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—Part 2: Recommendations and best practices. In *Remote Sensing*. mdpi.com. <https://www.mdpi.com/2072-4292/13/13/2591>
- [14] Dokic, K., Martinovic, M., & ... (2020). Inference speed and quantisation of neural networks with TensorFlow Lite for Microcontrollers framework. *2020 5th South-East* <https://ieeexplore.ieee.org/abstract/document/9221846/>
- [15] Wasilewski, K., & Zabierowski, W. (2021). A comparison of java, flutter and kotlin/native technologies for sensor data-driven applications. In *Sensors*. mdpi.com. <https://www.mdpi.com/1424-8220/21/10/3324>
- [16] Zuluaga, C. A., Aristizábal, L. M., Rúa, S., Franco, D. A., & ... (2022). Development of a modular software architecture for underwater vehicles using systems engineering. In *Journal of Marine* mdpi.com. <https://www.mdpi.com/2077-1312/10/4/464>