

DETEKSI DAN PERHITUNGAN POHON KELAPA SAWIT BERBASIS UAV DENGAN YOLO VERSI 11

Muhammad Fadly Syabana, Erwin Hermawan, Sahid Agustian Hurjimartsu

Teknik Informatika, Universitas Ibn Khaldun Bogor

mhmmfdlysybn@gmail.com

ABSTRAK

Kelapa sawit (*Elaeis guineensis jacq*) salah satu dari beberapa palma yang menghasilkan minyak dengan tujuan komersil. Minyak sawit selain digunakan sebagai minyak makanan margarine, dapat juga digunakan untuk industri sabun, lilin dan dalam pembuatan lembaran-lembaran timah serta industri kosmetik. Indonesia adalah penghasil minyak kelapa sawit kedua dunia setelah Malaysia. Di Indonesia penyebarannya di daerah Aceh, pantai timur Sumatra, Jawa, dan Sulawesi. Perluasan areal perkebunan pohon kelapa sawit terus berlanjut akibat meningkatnya harga minyak bumi sehingga peran minyak nabati meningkat sebagai energi alternatif. Proses perhitungan pohon kelapa sawit menjadi tantangan tersendiri jika dilakukan secara manual, sehingga membutuhkan waktu yang cukup lama. Oleh karena itu, ada kebutuhan untuk metode yang lebih efisien dan akurat untuk menghitung pohon kelapa sawit dari foto citra udara. Penelitian ini telah melakukan perhitungan pohon kelapa sawit otomatis yang lebih efisien dan akurat yaitu menggunakan model deteksi YOLO (*You Only Look Once*) dengan penambahan filter DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*). Hasil yang didapatkan pada penelitian ini model YOLOv11 dapat mendeteksi objek pohon kelapa sawit dengan akurasi 94,8%, dengan nilai *precision* 94,5% dan nilai *recall* 97,3%. Dari hasil akurasi yang didapatkan menunjukkan model YOLOv11 berhasil mendeteksi dan menghitung pohon kelapa sawit dengan baik.

Kata kunci : Deteksi Objek, Deep Learning, Pohon Sawit, UAV, YOLOv11

1. PENDAHULUAN

Kelapa sawit adalah salah satu komoditas perkebunan yang memiliki peran krusial dalam perekonomian Indonesia karena kemampuannya menghasilkan minyak sawit mentah (CPO) serta minyak inti sawit (PKO), yang diharapkan sebagai bahan baku industri makanan serta non makanan. Indonesia adalah negara pembuat serta pengeksport minyak kelapa sawit terbesar di dunia. Lebih dari lima puluh persen kebutuhan kelapa sawit dunia bisa dipenuhi oleh Indonesia sehingga menempatkan Indonesia pada sepuluh top negara pembuat dan pengeksport kelapa sawit dunia[1]. Luas areal perkebunan pohon kelapa sawit pada tahun 2020 sudah mencapai 14.621.693 Ha dengan total produksi CPO sebesar 45.741.845 ton dan produksi inti sawit 9.148.369 ton [2]. Dari data pusat Statistik, 2021 tercatat lonjakan ekspor minyak sawit paling signifikan pada lima tahun terakhir, mencapai nilai US\$ 27,6 miliar, dengan pertumbuhan 58,79% dibandingkan tahun sebelumnya [3]. Inilah yang menjadikan kelapa sawit sebagai faktor krusial pada perkembangan positif sektor pertanian, serta sudah menggambarkan diri sebagai komoditas utama yang memajukan sektor ekspor Indonesia.

Dengan meningkatnya permintaan global akan minyak sawit, khususnya di Indonesia, diperlukan upaya untuk meningkatkan dan mengembangkan di sektor kelapa sawit. Seperti yang sudah umum diketahui, dalam pertanian kelapa sawit saat ini masih umum menggunakan metode manual atau tradisional dalam menghitung jumlah pohon kelapa sawit pada suatu lahan. Seperti digitalisasi pohon kelapa sawit di lahan satu per satu, atau dengan survei lahan kelapa

sawit menggunakan GPS [4]. Oleh karena itu, untuk mengurangi hal-hal yang telah dijelaskan, penulis mengambil langkah untuk melakukan penelitian guna mencari solusi agar perhitungan pohon kelapa sawit dapat dilakukan secara otomatis. Dalam penelitian yang dilakukan oleh penulis mengambil lokasi penelitian di area lahan perkebunan pohon kelapa sawit yang terletak di provinsi Sumatera Utara, tepatnya di kecamatan Pengajahan, Serdang Bedagai. Metode tradisional penghitungan pohon sering kali memerlukan survei lapangan yang memakan waktu dan biaya. Oleh karena itu, ada kebutuhan untuk metode yang lebih efisien dan akurat untuk menghitung pohon kelapa sawit dari foto citra udara [5].

Dalam proses pengolahan, penelitian ini untuk deteksi objek adalah dengan menggunakan YOLO (*You Only Look Once*). YOLO adalah algoritma deteksi objek satu tahap (*one-stage object detection*), algoritma ini mengubah pendekatan konvensional deteksi objek dengan melakukan deteksi secara langsung hanya dalam satu kali pemrosesan gambar, sehingga jauh lebih cepat dan efisien [6]. Dalam penelitian ini, metode YOLOv11 digunakan untuk melakukan deteksi awal terhadap objek pohon kelapa sawit pada citra udara. Dalam praktiknya sering dijumpai adanya deteksi ganda (*multiple detection*) pada satu kanopi pohon, terutama ketika ukuran resolusi citra tinggi. Untuk mengatasi hal tersebut, algoritma DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) diterapkan sebagai tahap pasca-pemrosesan. DBSCAN digunakan untuk mengelompokkan titik-titik deteksi yang memiliki kedekatan spasial tinggi, dan kemudian

menghapus titik-titik duplikat yang berada dalam satu cluster, sehingga hanya satu titik per pohon yang dipertahankan [7][8].

2. TINJAUAN PUSTAKA

2.1. Kelapa Sawit

Kelapa sawit (*Elais Guineensis*) adalah tanaman tahunan yang berasal dari Afrika dan mampu tumbuh serta berbuah bahkan di ketinggian hingga 500 meter di atas permukaan laut. Untuk keperluan budidaya, tanaman ini memiliki umur ekonomis selama 25 hingga 35 tahun, dengan batang yang biasanya tumbuh mencapai 10 hingga 11 meter. Bagian utama yang memiliki nilai ekonomi tinggi adalah buahnya yang tersusun dalam bentuk tandan buah segar (TBS). Dari buah inilah diperoleh *Crude Palm Oil* (CPO) sekitar 20-40% dan *Palm Kernel Oil* (PKO) sekitar 3-4% [1].

2.2. UAV

UAV atau (*Unmanned Aerial Vehicle*) adalah pesawat yang terbang tanpa pilot di dalamnya. Pesawat ini bisa dikendalikan secara jarak jauh oleh seorang operator melalui perangkat seperti stasiun kontrol darat, ponsel pintar, atau komputer, atau bahkan dapat mengoperasikan dirinya secara otomatis melalui sensor dan prosesor *onboard*. Karena kemampuannya untuk terbang di area yang berbahaya atau sulit dijangkau, UAV telah banyak dimanfaatkan dalam berbagai aplikasi, mulai dari pengawasan hingga pemetaan dan operasi penyelamatan [5].

2.3. YOLO V11

YOLOv11 adalah evolusi terkini dalam keluarga algoritma YOLO yang menandai terobosan besar dalam teknologi deteksi objek secara *real-time*. Membangun dari keberhasilan versi-versi sebelumnya, YOLOv11 mengintegrasikan perbaikan arsitektural inovatif dan metode pelatihan yang lebih terperinci. Secara khusus, model ini memperkenalkan komponen baru, seperti blok C3k2, modul SPPF, dan modul C2PSA yang meningkatkan kemampuan ekstraksi fitur sekaligus menjaga ukuran model agar tetap ringan. Secara keseluruhan, YOLOv11 menawarkan keseimbangan yang baik antara performa dan efisiensi, sehingga sangat adaptif untuk berbagai aplikasi computer vision [7].

2.4. DBSCAN

DBSCAN atau *Density-Based Spatial Clustering of Applications with Noise* adalah algoritma klasterisasi berbasis kepadatan yang dirancang untuk menemukan kumpulan data (*clusters*) berdasarkan kepadatan titik-titik di ruang data, dan dapat menangani data yang mengandung *noise* [14].

2.5. Visual Studio Code

Visual Studio Code bukan hanya sekadar notepad yang telah berevolusi dengan fitur pewarnaan sintaks dan pengaturan indentasi otomatis. Melainkan, ia adalah sebuah lingkungan pengembangan yang

berfokus pada kode yang sangat kuat, yang dirancang khusus untuk memudahkan penulisan aplikasi *web*, *mobile*, dan *cloud* menggunakan bahasa-bahasa yang tersedia di berbagai *platform*, serta mendukung seluruh siklus pengembangan aplikasi dengan adanya debugger bawaan dan integrasi dengan sistem kontrol versi Git yang populer [15].

2.6. ArcGIS

ArcGIS merupakan sistem informasi geografis terpadu yang menggabungkan beberapa komponen utama. Pengguna dapat mengoperasikan sistem ini melalui perangkat lunak ArcGIS Desktop, yakni kumpulan aplikasi SIG canggih dan terintegrasi. Selain itu, Esri menyediakan antarmuka untuk pengelolaan basis data melalui sistem manajemen database (DBMS). Tersedia pula versi SIG berbasis internet yang didesain untuk mendistribusikan data dan layanan spasial. Esri juga mengembangkan ekstensi seperti ArcPad, yang mendukung aplikasi SIG pada perangkat tablet atau *palmtop*. Karena fleksibilitasnya, ArcGIS dikenal sebagai produk yang scalable, artinya sistem ini dapat diterapkan dalam berbagai skala. Seorang pengguna dapat memilih konfigurasi yang berjalan hanya pada satu komputer, yang dikenal dengan istilah SIG pribadi, dimana pengguna dapat melihat dan memilih data spasial secara sederhana, meskipun tidak memungkinkan pengeditan atau analisis secara mendalam. Di sisi lain, versi SIG multipengguna sudah dirancang untuk memenuhi kebutuhan seluruh perusahaan atau instansi pemerintah, menyediakan seluruh kemampuan dari perangkat lunak tersebut. Di antara kedua ekstrim tersebut, terdapat pula opsi dengan fungsionalitas menengah yang biasa disebut sebagai SIG kelompok kerja atau SIG departemen [16].

2.7. Confusion Matrix

Confusion matrix digunakan untuk mengevaluasi kinerja model. *Confusion matrix* adalah matriks yang menampilkan deteksi klasifikasi aktual dan klasifikasi yang dideteksi, terdapat 4 klasifikasi pada *confusion matrix* yaitu *True Negative* (TN) jumlah sampel negatif yang diklasifikasikan benar, *True Positive* (TP) yaitu jumlah sampel positif yang diklasifikasikan dengan benar, *False Positive* (FN) adalah jumlah sampel positif yang diklasifikasikan sebagai negatif dan *False Negative* (FP) yaitu jumlah sampel negatif yang diklasifikasikan sebagai positif [17].

2.8. Akurasi

Akurasi merupakan metrik digunakan untuk mengukur seberapa tepat hasil deteksi dibandingkan dengan nilai aktual. Dalam konteks model deteksi atau klasifikasi, akurasi dihitung sebagai rasio antara jumlah deteksi yang benar dengan total deteksi yang dilakukan. Nilai akurasi yang tinggi menunjukkan bahwa model mampu mendeteksi dengan tepat sebagian besar kasus, sehingga mencerminkan kinerja model yang baik. Akurasi menjadi indikator penting

untuk menilai efektivitas metode yang digunakan dalam berbagai aplikasi, seperti klasifikasi citra, deteksi anomali, atau deteksi tren. Dengan kata lain, semakin tinggi nilai akurasi, semakin andal dan efektif metode tersebut dalam menghasilkan deteksi yang mendekati nilai aktual [18].

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

2.9. Precision

Precision adalah salah satu teknik pengujian yang membandingkan jumlah informasi relevan yang dihasilkan oleh sistem dengan total informasi yang diambil oleh sistem, termasuk informasi yang relevan dan tidak relevan [19].

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

2.10. Recall

Recall adalah metode pengujian yang membandingkan jumlah informasi relevan yang ditemukan oleh sistem dengan total jumlah informasi relevan yang ada dalam koleksi informasi (baik yang ditemukan maupun yang tidak ditemukan oleh sistem) [19].

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

2.11. Uji Akurasi

Pada tahap uji akurasi ini diperlukan pengambilan sampel pada citra udara yang sudah di deteksi oleh model. Salah satu metode yang digunakan untuk mengambil sampel yaitu *purposive sampling* adalah teknik pengambilan sampel yang dilakukan dengan menetapkan kriteria-kriteria atau seleksi tertentu untuk memilih sampel yang sesuai dengan tujuan penelitian. Dengan menggunakan metode *purposive sampling* dan *simple random sampling*, diharapkan pengujian dapat menghasilkan data yang lebih spesifik, karena sampel ini dipilih berdasarkan karakteristik atau target tertentu yang relevan dengan penelitian. Hal ini memungkinkan peneliti untuk fokus pada subkelompok yang memiliki ciri-ciri penting yang ingin diteliti, sehingga analisis bisa lebih mendalam dan relevan [20].

3. METODE PENELITIAN

3.1. Waktu dan Lokasi

Penelitian ini dimulai pada bulan Agustus 2024 yang telah melakukan observasi langsung di lapangan. Citra drone diambil dari perkebunan kelapa sawit Kec. Pengajahan, Kab. Serdang Bedagai, Sumatera Utara.

3.2. Bahan dan Alat

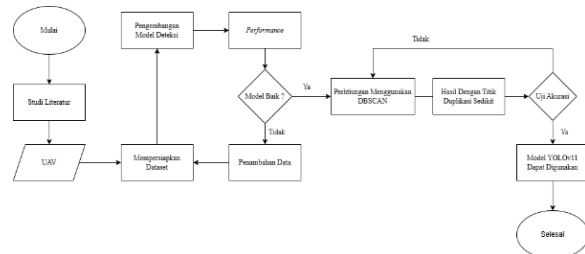
Alat yang digunakan pada penelitian perhitungan pohon kelapa sawit ini terbagi menjadi dua yaitu bahan dan alat.

- Bahan
 - Citra udara UAV kawasan perkebunan pohon kelapa sawit
- Alat
 - Laptop

- ArcGIS
- Roboflow
- Google Colab
- Microsoft Office 2024

3.3. Diagram Alir Penelitian

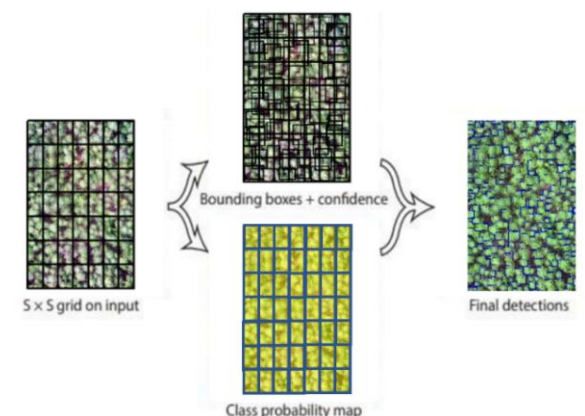
Pada penelitian ini terdapat beberapa tahapan yang dilakukan, dapat dilihat pada Gambar 1.



Gambar 1. Diagram alir penelitian

3.4. Metode Yang Digunakan

Pada metode YOLO (*You Only Look Once*) untuk mendeteksi pohon kelapa sawit melibatkan beberapa tahap kunci. Pertama, dataset citra UAV dikumpulkan. Selanjutnya, gambar diproses melalui pemotongan, perubahan ukuran (*resize*), dan augmentasi untuk meningkatkan kegunaan dan variasi data. Konfigurasi model kemudian ditetapkan, mencakup detail arsitektur YOLO (kelas objek “Sawit”) dan jalur dataset. Model YOLO (dalam penelitian ini *YOLOv11*) dilatih menggunakan dataset tersebut guna mengoptimalkan bobotnya untuk deteksi pohon kelapa sawit. Setelah evaluasi model, deteksi dilakukan dan model yang telah dilatih akan menampilkan bounding box pada lokasi pohon kelapa sawit terdeteksi. Selanjutnya deteksi pohon kelapa sawit dilakukan secara lokal menggunakan *YOLOv11* dengan format *PyTorch* (.pt) yang telah dilatih. Hasil deteksi berupa *dictionary* dikonversi menjadi koordinat sudut *bounding box*. Selanjutnya, titik-titik pusat *bounding box* diolah dengan DBSCAN untuk menghilangkan duplikasi deteksi pada area berhimpitan (*overlapping points*). Titik yang telah terklusterisasi kemudian diimpor kedalam *Geodataframe* dan diekspor sebagai berkas *shapefile*.

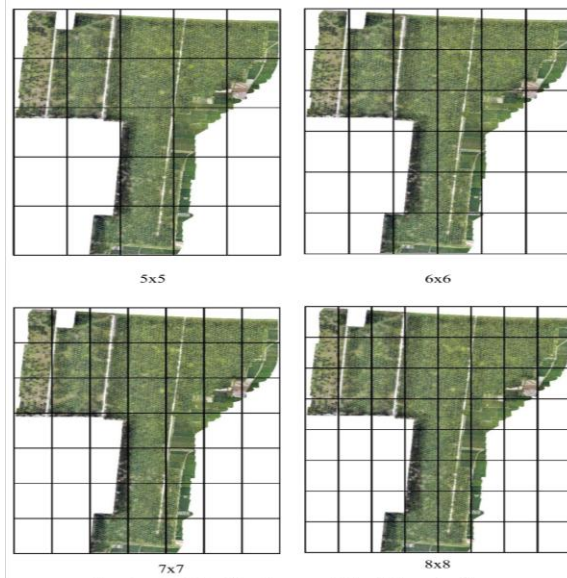


Gambar 2. Lapisan deteksi YOLO

4. HASIL DAN PEMBAHASAN

4.1. Pengolahan Data

Penelitian ini dimulai dengan pengumpulan dataset UAV dan memotong citra menggunakan *fishnet* dengan grid 5x5, 6x6, 7x7, dan 8x8. Sehingga menghasilkan 98 bagian citra yang lebih kecil. Hasil pemotongan citra dapat dilihat pada Gambar 3.



Gambar 3. Citra pohon kelapa sawit

4.2. Anotasi Citra

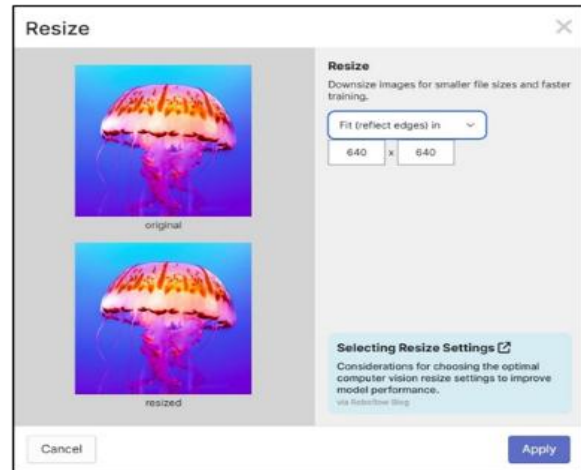
Proses anotasi citra yaitu memberikan *bounding box* pada objek yang akan diidentifikasi dengan memberikan kelas pada objek pohon kelapa sawit. Tahap anotasi ini dilakukan pada Roboflow, dengan mengunggah citra yang telah di potong sebelumnya. Hasil anotasi dapat dilihat pada Gambar 4.



Gambar 4. Anotasi pohon kelapa sawit

4.3. Resize Citra

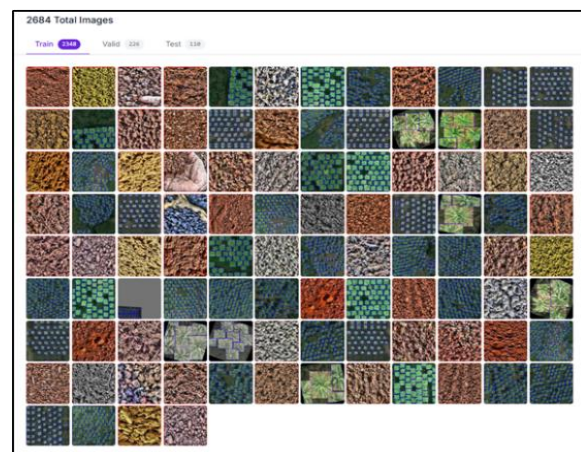
Proses *resize* citra dilakukan untuk menyeragamkan ukuran menjadi 640 x 640 piksel menggunakan metode *fit reflects edges* (*padding letterbox*). Tahapan ini diimplementasikan melalui platform Roboflow. Proses *resize* citra dapat dilihat pada Gambar 5.



Gambar 5. Resize citra

4.4. Augmentasi

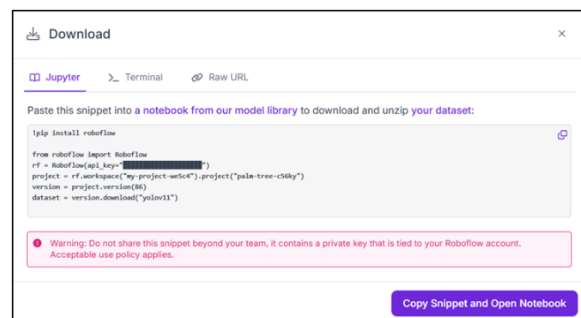
Dengan menggunakan Augmentasi seperti *Flip* 90°, *Rotate*, *Filter Null* sehingga menghasilkan citra yang lebih bervariasi. Hasil augmentasi dapat dilihat pada Gambar 6.



Gambar 6. Augmentasi

4.5. Pembuatan Model Deteksi

Setelah didapatkan API kode, selanjutnya model di *train* sebanyak 50 *epoch* di *google collaboratory*. Dalam proses ini, model akan disesuaikan dengan data pelatihan sebanyak 50 *epoch* dan validasi untuk setiap *epoch*, di mana setiap *epoch* mewakili satu iterasi melalui seluruh *dataset* pelatihan. Dibawah ini dapat dilihat untuk membuat dataset pada Gambar 7.

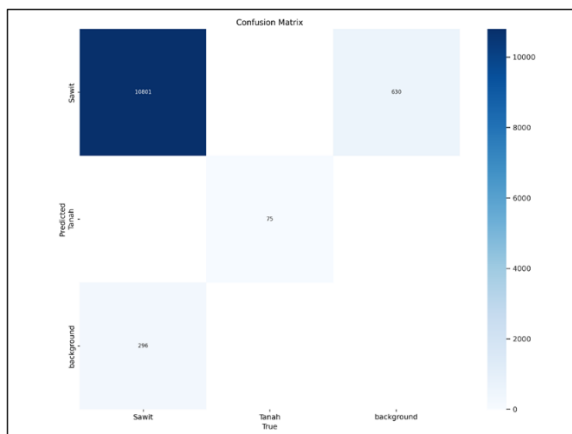


Gambar 7. Membuat dataset

4.6. Performance

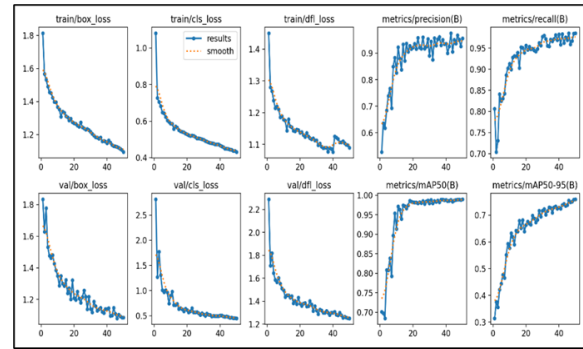
Setelah melakukan *training* dan validasi dilakukan *testing* model yaitu menguji model dengan data yang belum pernah dilihat sebelumnya sebanyak 110 citra dari data *testing*, yang kemudian digunakan sebagai sampel untuk menguji kemampuan model dalam membuat hasil deteksi dan menghitung *confusion matrix* sehingga mendapatkan hasil grafik, akurasi, *precision*, dan *recall*. Dari tahapan ini memperoleh hasil performa model yang lebih spesifik mengenai kinerja model dan memastikan model tidak *overfitting* serta model dapat mengenali objek pohon kelapa sawit yang sudah dipelajari dari *dataset*.

Dari hasil evaluasi model, *confusion matrix* dihasilkan untuk memvisualisasikan performa model deteksi objek pada kelas sawit. Evaluasi menunjukkan model secara akurat mengidentifikasi 10.851 objek sawit (*true positive*), namun 630 objek tidak terdeteksi (*false negative*). Di sisi lain, 296 area *background* salah dikenali sebagai sawit (*false positive*), mengindikasikan kesalahan deteksi pada area non-sawit. Hasil gambar *confusion matrix* dapat dilihat pada Gambar 8.



Gambar 8. *Confusion matrix*

Selain mendapatkan hasil *Confusion Matrix*, terdapat juga grafik evaluasi model sebagai memvisualisasikan grafik pelatihan dan evaluasi model deteksi objek yang mengukur metrik kinerja dan nilai *loss*. Terdapat pola penurunan konsisten pada nilai *loss* (*box loss*, *classification loss*, dan *objectness loss*) di data latih dan validasi seiring peningkatan epoch, mengindikasikan peningkatan kemampuan model dalam mengidentifikasi objek kelapa sawit dan membedakannya dari *background*. Secara paralel, metrik presisi dan recall menunjukkan tren kenaikan stabil mendekati 1, merefleksikan akurasi dan sensitivitas deteksi yang optimal. Peningkatan signifikan juga terlihat pada *mean Average Precision* (*mAP*) di kedua *threshold* (0.5 dan 0.5:0.95), yang stabil pada nilai tinggi di akhir pelatihan. Konvergensi seluruh metrik ini mengonfirmasi efektivitas proses pelatihan dan pencapaian kinerja deteksi optimal. Berikut ini dapat dilihat grafik evaluasi model pada Gambar 9.



Gambar 9. Grafik evaluasi model

Tabel 1. Akurasi, Precision, dan Recall

Kelas	Precision	Recall
Sawit	94,49%	97,33%
Akurasi Model	92,15%	

Hasil akurasi, presisi dan recall diatas didapatkan dari perhitungan persamaan berikut ini:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Akurasi = \frac{10.801 + 75}{10.801 + 75 + 630 + 296} = \frac{10.876}{11.802} = 0.9215 \times 100 = 92,15\%$$

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{10.801}{10.801 + 630} = \frac{10.801}{11.431} = 0.9449 \times 100 = 94,49\%$$

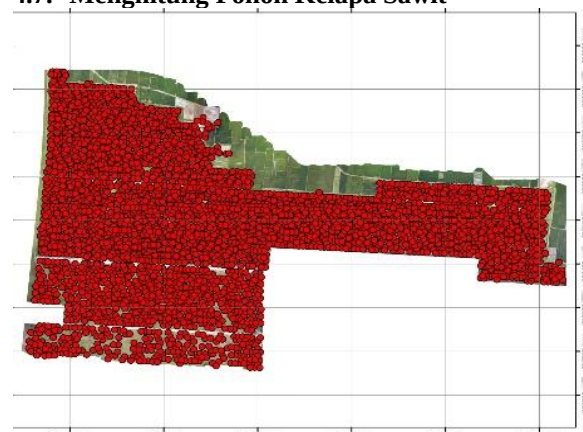
$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{10.801}{10.801 + 296} = \frac{10.801}{11.097} = 0.9773 \times 100 = 97,33\%$$

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$F1score = 2 \times \frac{0,9449 \times 0,9773}{0,9449 + 0,9773} = \frac{0,9236}{1,9222} = 2 \times 0,4804 = 0,9608 = 96,08\%$$

4.7. Menghitung Pohon Kelapa Sawit



Gambar 10. Hasil perhitungan dengan DBSCAN

Pada Gambar 10 menunjukkan hasil dari sebuah *shapefile* yang telah diproses untuk menghitung jumlah pohon kelapa sawit serta DBSCAN diterapkan untuk menghilangkan duplikasi deteksi pada area

berhimpitan (*overlapping points*). Dalam gambar ini, setiap pohon kelapa sawit yang terhitung diwakili oleh poin berwarna merah yang tersebar di seluruh area. Berdasarkan perhitungan yang dilakukan, total pohon kelapa sawit yang berhasil dihitung adalah sebanyak 4212 pohon. Hasil perhitungan pohon kelapa sawit disertai penerapan DBSCAN dapat dilihat pada Gambar 9.

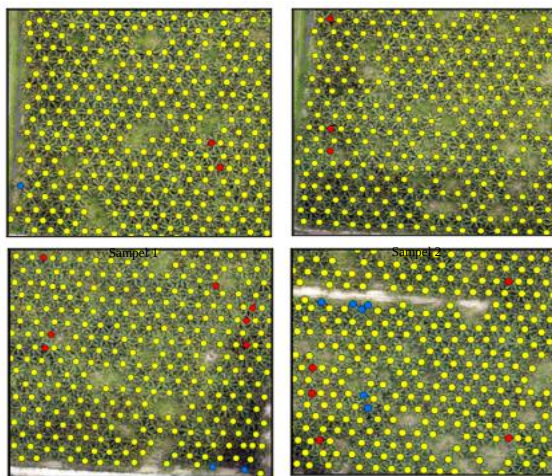
4.8. Uji Akurasi

Setelah titik *error* lebih dan *error* kurang didapatkan kemudian dihitung menggunakan persamaan akurasi, *commission error* dan *omission error* untuk memastikan seberapa akuratnya model dari YOLOv11. Berikut adalah tabel uji akurasi dari setiap sampel dari metode YOLOv11, adapun hasil uji akurasi terdapat pada Tabel 2.

Tabel 2. Hasil Uji Akurasi

Model Deteksi Pohon Kelapa Sawit								
Area	Total Referensi	Hasil Klasifikasi	Error Lebih	Error Kurang	Correct	Commission Error	Omission Error	Accuracy
Sampel 1	280	277	2	1	275	0,7%	0,4%	97,5%
Sampel 2	276	273	3	0	270	1,1%	0%	96,8%
Sampel 3	286	277	7	2	270	2,5%	0,7%	92,2%
Sampel 4	282	271	5	6	266	1,8%	2,2%	92,7%
Nilai Rata-Rata						1,5%	0,8%	94,8%

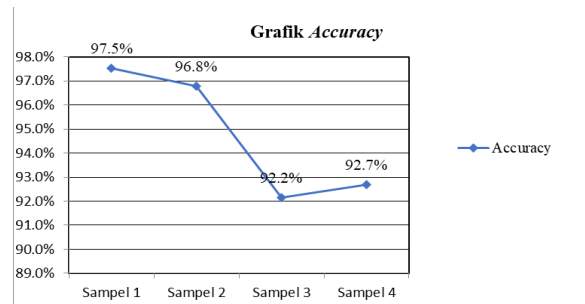
Berdasarkan dari hasil uji akurasi *commission error* dan *omission error* dengan 4 sampel yang ditunjukkan pada Tabel 2 dapat disimpulkan hasil nilai keseluruhan akurasi dari rata-rata sampel yaitu 94,8%, *Commission error* 1,5% dan *Omission error* 0,8%. Gambar sampel, hasil grafik *accuracy* dan grafik *error* dapat dilihat pada Gambar 12 dan Gambar 13.



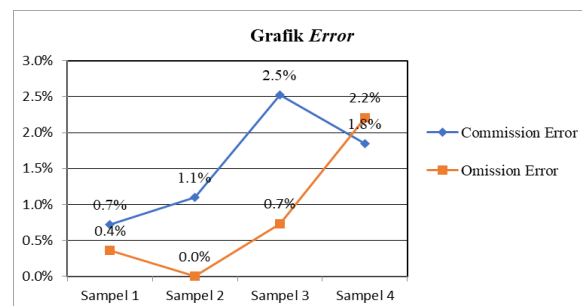
Legenda: Pohon Error Kurang Error Lebih

Gambar 11. Sampel uji akurasi

Grafik ini memvisualisasikan hasil dari 4 sampel yang diuji menggunakan *commission error* dan *omission error*, dapat dilihat sampel 1 memperoleh akurasi 97,5%, *commission error* 0,7% dan *omission error* 0,4%, sampel 2 memperoleh akurasi 96,8%, *commission error* 1,1% dan *omission error* 0%, sampel 3 memperoleh akurasi 92,2%, *commission error* 2,5% dan *omission error* 0,7% dan sampel 4 memperoleh akurasi 92,7%, *commission error* 1,8% dan *omission error* 2,2%. Secara keseluruhan grafik ini menunjukkan bahwa model dapat melakukan deteksi pohon kelapa sawit dengan nilai akurasi yang tinggi dan *error* yang lebih sedikit.



Gambar 12. Grafik *accuracy*



Gambar 13. Grafik *error*

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian mengenai deteksi dan perhitungan jumlah pohon sawit menggunakan YOLOv11, dapat disimpulkan bahwa model deteksi objek pohon kelapa sawit menggunakan YOLOv11 dapat mendeteksi objek pohon kelapa sawit dengan akurasi mencapai lebih dari 92,15%. Nilai akurasi ini menunjukkan bahwa model yang dibangun berhasil mendeteksi pohon kelapa sawit dengan baik dan efisien, memanfaatkan citra udara dari UAV. Uji akurasi pada 4 sampel yang diambil, didapatkan *commission error* dan *omission error* yang bervariasi pada setiap sampel. Pada Sampel 1, *commission error* tercatat sebesar 0,7% dan *omission error* sebesar 0,4%, dengan akurasi 97,5%. Sampel 2 menunjukkan nilai *commission error* 1,1% dan *omission error* 0%, dengan akurasi 96,8%. Sampel 3 memiliki *commission*

error 2,5% dan omission error 0,7%, dengan akurasi 92,2%, sedangkan Sampel 4 memiliki commission error 1,8% dan omission error 2,2%, dengan akurasi 92,7%. Secara keseluruhan, nilai akurasi rata-rata dari 4 sampel yang diuji adalah 94,8%, yang menunjukkan bahwa model YOLOv11 sangat efektif dalam mendeteksi pohon kelapa sawit. Berdasarkan hasil penelitian yang telah dilakukan pada perhitungan pohon sawit menggunakan YOLOv11, diharapkan dapat dikembangkan ke tahap selanjutnya yaitu: Agar pelatihan model dapat dilakukan secara optimal, disarankan untuk menggunakan komputasi yang lebih memadai, seperti penggunaan GPU dengan kapasitas lebih besar saat menangani dataset yang lebih besar atau kompleks agar dapat dilakukan *training* model dengan maksimal. Disarankan untuk menguji dan membandingkan YOLOv11 dengan pustaka deteksi objek lain. Hal ini dapat memberikan wawasan tentang performa masing-masing model dalam konteks deteksi pohon kelapa sawit, baik dari sisi akurasi, kecepatan deteksi, maupun kemampuan deteksi objek pada citra dengan kualitas rendah.

DAFTAR PUSTAKA

- [1] D. Irawati, *Merancang Kelapa Sawit Sebagai Komoditi Unggulan Nasional*, vol. 1. Malang: PT. Literasi Nusantara Abadi Grup, 2023.
- [2] M. Siahaan and O. Darnius, "Klasifikasi Provinsi Di Indonesia Berdasarkan Luas Pengusahaan Dan Produktivitas Tanaman Kelapa Sawit Menggunakan Analisa Klaster," *J. Agro Estate*, vol. 7, no. 2, pp. 1–10, 2023, doi: 10.47199/jae.v7i2.164.
- [3] D. S. Prasvita, M. M. Santoni, R. Wirawan, and N. Trihastuti, "Klasifikasi Pohon Kelapa Sawit Pada Data Fusi Citra Lidar Dan Foto Udara Menggunakan Convolutional Neural Network," *JIPi (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 6, no. 2, pp. 406–415, 2021, doi: 10.29100/jipi.v6i2.2437.
- [4] A. Ammar, A. Koubaa, and B. Benjdira, "Deep-learning-based automated palm tree counting and geolocation in large farms from aerial geotagged images," *Agronomy*, vol. 11, no. 8, pp. 1–24, 2021, doi: 10.3390/agronomy11081458.
- [5] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif, "Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review," *Drones*, vol. 6, no. 6, 2022, doi: 10.3390/drones6060147.
- [6] X. Cong, S. Li, F. Chen, C. Liu, and Y. Meng, "A Review of YOLO Object Detection Algorithms based on Deep Learning," vol. 4, no. 2, pp. 2–5, 2023, doi: 10.54097/fcis.v4i2.9730.
- [7] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," vol. 2024, pp. 1–9, 2024, doi: 10.48550/arXiv.2410.17725.
- [8] D. Prasetia, T. Wardana, S. S. Indonesia, R. S. Sianturi, and F. Ramdani, "Detection of Oil Palm Trees Using Deep Learning Method with High-Resolution Detection of Oil Palm Trees Using Deep Learning Method with High-Resolution Aerial Image Data YOLOv8-Based Deep Learning for Accurate and Efficient Oil Palm Tree Detection," no. October, 2023, doi: 10.1145/3626641.3626667.
- [9] H. Wibowo, I. S. Sitanggang, M. Mushthofa, and H. A. Adrianto, "Large-Scale Oil Palm Trees Detection from High-Resolution Remote Sensing Images Using Deep Learning," *Big Data Cogn. Comput.*, vol. 6, no. 3, 2022, doi: 10.3390/bdcc6030089.
- [10] I. Nurhabib, K. B. Seminar, and Sudradjat, "Recognition and counting of oil palm tree with deep learning using satellite image," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 974, no. 1, 2022, doi: 10.1088/1755-1315/974/1/012058.
- [11] M. R. Rahman, R. Kusumawati, and F. Fatimah, "Deteksi Objek Menghitung Pohon Kelapa Sawit Menggunakan Metode Deep Learning Object Detection Tree Counting Palm Oilusing Deep Learning Method," *Bina J. Pembang. Drh.*, vol. 2, no. 1, pp. 45–51, 2023, doi: 10.62389/bina.v2i1.51.
- [12] D. S. Prasvita, M. M. Santoni, and N. Falih, "Deteksi Pohon Kelapa Sawit dengan Pendekatan Deep Learning pada Citra Multispectral di Indonesia," *Indones. J. Comput. Sci.*, vol. 13, no. 2, pp. 3111–3125, 2024, doi: 10.33022/ijcs.v13i2.3753.
- [13] I. M. Shaikh, M. N. Akhtar, A. Aabid, and O. S. Ahmed, "Enhancing sustainability in the production of palm oil: creative monitoring methods using YOLOv7 and YOLOv8 for effective plantation management," *Biotechnol. Reports*, vol. 44, no. June, p. e00853, 2024, doi: 10.1016/j.btre.2024.e00853.
- [14] A. Fahim, "An Extended DBSCAN Clustering Algorithm," vol. 13, no. 3, pp. 245–258, 2022, doi: 10.14569/IJACSA.2022.0130331.
- [15] A. Del Sole, *Visual Studio Code Distilled*. Cremona: Apress, 2023.
- [16] [M. K. Dr. Joseph Teguh Santoso, S.Kom., *GIS Sistem Informasi Geografis*. Semarang: Yayasan Prima Agus Teknik, 2021.
- [17] I. P. Sary, S. Andromeda, and E. U. Armin, "Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images," *Ultim. Comput. J. Sist. Komput.*, vol. 15, no. 1, pp. 8–13, 2023, doi: 10.31937/sk.v15i1.3204.
- [18] N. J. Hayati, D. Singasatia, and M. R. Muttaqin, "Object Tracking Menggunakan Algoritma You Only Look Once (YOLO) v8 Untuk Menghitung Kendaraan," vol. 12, no. 2, pp. 91–99, 2023, doi: 10.34010/komputa.v12i2.10654.
- [19] A. Ma'aruf and M. Hardjianto, "Application of the You Only Look Once Version 8 Algorithm for Indonesian Sign Language Alphabet," *Semin. Nas. Mhs. Fak. Teknol. Inf.*, vol. 2, no. September, pp. 567–576, 2023.
- [20] I. Lenaini, "Teknik Pengambilan Sampel Purposive Dan Snowball Sampling," *Hist. J. Kajian, Penelit. Pengemb. Pendidik. Sej.*, vol. 6, no. 1, pp. 33–39, 2021, doi: <https://doi.org/10.31764/historis.v6i1.4075>.