

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kelapa sawit (*Elaeis guineensis*) merupakan komoditas perkebunan yang berperan penting dalam menunjang perekonomian Indonesia. Produk utama kelapa sawit, yaitu minyak kelapa sawit (*Crude Palm Oil*, CPO), memiliki permintaan tinggi di pasar domestik maupun internasional. Pada tahun 2023 ekspor minyak kelapa sawit mencapai 5,44 juta ton ke Tiongkok dan 1,98 juta ton ke Amerika Serikat (Badan Pusat Statistik, 2023). Data ini menggambarkan besarnya kontribusi kelapa sawit terhadap perekonomian nasional sekaligus urgensi untuk menjaga kualitas produksi yang tinggi.

Penentuan tingkat kematangan Tandan Buah Segar (TBS) menjadi faktor yang sangat penting untuk menjaga mutu minyak yang dihasilkan. Kematangan buah sawit berpengaruh langsung pada kualitas minyak akhir, dimana buah yang matang optimal memiliki potensi menghasilkan rendemen minyak yang lebih tinggi (Islamiah et al., 2021). Umumnya, kematangan buah kelapa sawit dapat diamati dari jumlah brondolan yang jatuh dari tandannya (Lai et al., 2023) serta perubahan warna pada brondolan tersebut (Herman et al., 2020). Buah yang terlalu matang berisiko meningkatkan kadar asam lemak bebas (*Free Fatty Acid*, FFA) yang berdampak negatif terhadap kualitas minyak sawit secara keseluruhan (Sitio et al., 2022). Sehingga, pemanenan buah dalam keadaan lewat matang akan meningkatkan FFA dan menurunkan mutu minyak, sedangkan panen saat buah belum matang akan menghasilkan FFA rendah tetapi akan menghasilkan rendemen minyak sawit yang rendah sehingga menurunkan produksi minyak sawit (Sirait & Riyan Artha, 2023). Oleh karena itu, penentuan waktu panen yang tepat harus dilakukan secara cermat agar hasil produksi optimal dan nilai ekonomis minyak sawit tetap tinggi (Akbar & Wibowo, 2023).

Selama ini, penilaian tingkat kematangan tandan sawit umumnya masih dilakukan secara manual oleh pekerja perkebunan. Pekerja biasanya harus memanjat untuk menentukan kematangan berdasarkan pengamatan visual perubahan warna atau menghitung jumlah brondolan yang jatuh dari tandan (Lai, 2022). Namun, metode ini bersifat subjektif dan memiliki risiko ketidakakuratan tinggi yang berdampak pada penurunan efisiensi produksi dan kualitas minyak sawit. Gunawan (2023) menyebutkan bahwa penurunan rendemen minyak (*Oil Extraction Rate*, OER) sebesar 0,13% akibat kesalahan panen buah yang belum matang dapat menimbulkan kerugian finansial yang besar hingga mencapai RM 340 juta. Kesulitan penilaian tersebut akan semakin meningkat seiring dengan banyaknya jumlah pohon sawit dan tingginya batang pohon yang menyulitkan pekerja untuk mengecek kematangan tandan secara langsung satu per satu. Kondisi ini tidak hanya mengurangi efisiensi kerja, tetapi juga meningkatkan risiko

kelelahan fisik dan potensi kecelakaan kerja akibat aktivitas pemanjatan yang berulang.

Seiring dengan perkembangan teknologi, penelitian berbasis *computer vision* dan *deep learning* mulai dilakukan untuk membantu proses deteksi kematangan TBS kelapa sawit secara otomatis. Penelitian oleh Hidayatullah et al. (2022) menggunakan pendekatan YOLOv4 berhasil mendeteksi tingkat kematangan dengan sangat baik pada kondisi ideal, mencapai nilai *mean Average Precision* (mAP) sebesar 99%. Akan tetapi, penelitian tersebut dilakukan pada citra yang diseleksi ketat, sehingga kurang merepresentasikan kondisi nyata di perkebunan sawit.

Penelitian lanjutan dilakukan oleh Genoveva dan Syah (2024) menggunakan metode YOLOv8 dengan dataset yang lebih besar. Penelitian ini mampu mencapai nilai mAP sebesar 98%. Namun, dalam pengujian kondisi nyata menggunakan video, terjadi penurunan performa deteksi secara signifikan ketika objek TBS tertutup sebagian oleh pelepah atau tandan buah lainnya. Ini menunjukkan bahwa YOLOv8, meskipun efektif di kondisi ideal, masih memiliki keterbatasan dalam kondisi *occlusion*.

Tantangan deteksi objek tertutup (*occlusion*) secara khusus ditangani oleh Sapkota dan Karkee (2025) yang membandingkan kinerja model YOLO11-seg dengan YOLOv8-seg untuk segmentasi objek apel muda yang tertutupi daun di kebun. Penelitian tersebut menunjukkan bahwa YOLO11-seg memiliki performa terbaik dalam menangani objek tertutup sebagian dengan mAP@0,50 sebesar 81%, dibandingkan dengan YOLOv8 yang lebih cepat dalam pemrosesan. Keunggulan YOLO11-seg terletak pada arsitektur *Backbone* yang lebih dalam untuk ekstraksi fitur, penggunaan *prototype masks* yang lebih presisi, serta *loss function* yang mampu mengurangi *false positive* dan *false negative* secara lebih efektif. Meskipun pada kasus ini proses inferensinya relatif lebih lambat, akurasi yang lebih tinggi dalam situasi *occlusion* menjadikan model ini relevan untuk kondisi kebun yang kompleks.

Namun, efektivitas model tersebut perlu diuji lebih lanjut dalam konteks berbeda, seperti deteksi kematangan tandan buah segar (TBS) kelapa sawit di kebun, yang memiliki tantangan visual tersendiri, terutama jika citra diambil dari jarak jauh. Kondisi lapangan yang curam dan tidak mudah dijangkau menuntut metode akuisisi citra yang aman dan efisien. Oleh karena itu, penelitian ini mengusulkan evaluasi dan perbandingan kinerja YOLOv8-seg dan YOLO11-seg dalam mendeteksi kematangan TBS menggunakan kamera teropong beresolusi tinggi dengan kemampuan zoom optik hingga 4x dan digital hingga 12x, yang memungkinkan observasi tanpa perlu mendekati objek secara langsung.

Penelitian ini bertujuan untuk menganalisis dan membandingkan performa YOLOv8-seg dan YOLO11-seg dalam mendeteksi kematangan TBS kelapa sawit secara langsung di pohon. Hasil dari perbandingan ini diharapkan dapat memberikan rekomendasi teknologi deteksi yang akurat dan efisien untuk mendukung kemajuan teknologi pertanian serta meningkatkan produktivitas dan mutu hasil produksi minyak kelapa sawit di Indonesia.

1.2 Landasan Teori

1.2.1 Tandan Buah Segar (TBS) Kelapa Sawit

Tandan Buah Segar (TBS) kelapa sawit (*Elaeis guineensis* Jacq.) adalah komponen penting dari tanaman kelapa sawit yang bernilai ekonomis tinggi. Bergantung pada varietas dan usia tanaman, satu tandan dapat mengandung ratusan buah sawit dengan bobot rata-rata 10 hingga 25 kilogram (Idris et al., 2020). Gambar 1 menunjukkan contoh TBS kelapa sawit yang telah dipanen di mana setiap tandan terdiri atas ratusan buah kecil yang tersusun rapat. Setiap buah sawit terdiri dari empat bagian penting, yaitu eksokarp (kulit luar), mesokarp (daging buah), endokarp (cangkang), dan kernel (inti). Mesokarp menghasilkan minyak sawit mentah, dan kernel menghasilkan minyak inti sawit.



Gambar 1. TBS Kelapa Sawit

Tingkat kematangan TBS sangat mempengaruhi kualitas minyak yang dihasilkan. Tandan dalam kondisi mentah menghasilkan rendemen minyak yang lebih rendah, sementara tandan yang terlalu matang meningkatkan kadar asam lemak bebas (*Free Fatty Acid*) yang tinggi, yang mengakibatkan penurunan kualitas CPO (*Crude Palm Oil*) (Sitio et al., 2022). Penurunan tersebut dapat menimbulkan kerugian finansial hingga menjapai ratusan juta rupiah (Gunawan et al., 2023). Oleh karena itu, sistem penentuan kematangan yang tepat sangat penting agar produktivitas optimal dan kualitas minyak tetap terjaga. Metode paling umum untuk mengetahui tingkat kematangan TBS Adalah dengan melakukan perhitungan jumlah brondolan yang jatuh secara alami dari tandan dan perubahan warna buah dari kehitaman menjadi jingga kemerahan yang disebabkan oleh akumulasi pigmen karotenoid (Herman et al., 2020; Lai et al., 2023).

1.2.2 Visi Komputer

Computer vision merupakan bidang ilmu yang berfokus pada bagaimana komputer dapat meniru kemampuan manusia dalam melihat dan menafsirkan informasi visual dari gambar maupun video. Tujuan utama dari bidang ini adalah mengembangkan metode dan algoritma yang memungkinkan komputer untuk mengenali objek,

memahami adegan, serta melakukan analisis visual secara otomatis (Wu et al., 2018).

Secara historis, *computer vision* berakar dari penggabungan antara pengolahan citra (*image processing*) dan kecerdasan buatan (*artificial intelligence*) (Bowden, 2015). Dalam perkembangannya, *computer vision* telah digunakan untuk mendukung berbagai aplikasi, mulai dari sistem keamanan seperti pengenalan wajah, bidang transportasi seperti kendaraan tanpa pengemudi, kesehatan seperti analisis citra medis, hingga hiburan dan industri kreatif seperti *augmented reality* (Jiang et al., 2023).

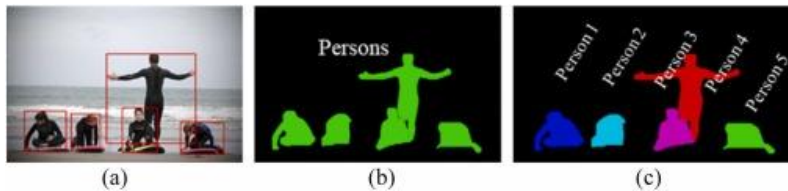
Keterkaitan *computer vision* dengan bidang lain seperti robotika, otomasi industri, serta pemrosesan sinyal menjadikannya salah satu teknologi inti di era digital. Dengan semakin banyaknya data visual yang dihasilkan setiap hari, kebutuhan akan sistem yang mampu mengolah dan menafsirkan citra secara otomatis semakin meningkat. Hal ini menjadikan *computer vision* sebagai salah satu fondasi penting dalam mendukung transformasi teknologi modern di berbagai sektor kehidupan (Ali, 2023).

1.2.3 Segmentasi Objek

Segmentasi objek (*Object Segmentation*) merupakan salah satu cabang utama dalam bidang *computer vision* yang berfokus pada pemisahan objek dalam citra hingga tingkat piksel. Tidak hanya mendeteksi keberadaan dan posisi objek seperti pada object detection, object segmentation juga berusaha memberikan batas yang lebih detail sehingga setiap piksel dalam citra dapat diklasifikasikan sesuai dengan kelas objek yang direpresentasikan (Wu et al., 2019).

Segmentasi objek memiliki peran penting karena menghasilkan representasi visual yang lebih kaya. Dengan adanya informasi pada level piksel, sistem komputer dapat memahami struktur citra dengan lebih akurat, sehingga hasil analisis visual lebih informatif dibandingkan hanya menggunakan kotak pembatas. Informasi detail ini sangat bermanfaat pada bidang-bidang yang membutuhkan ketelitian tinggi, seperti kesehatan, kendaraan otonom, maupun penginderaan jauh. Secara umum, object segmentation terbagi menjadi dua kategori, yaitu *semantic segmentation* dan *instance segmentation*. *Semantic segmentation* bertujuan untuk memberikan label kelas pada setiap piksel dalam citra, sehingga semua objek dengan kelas yang sama akan dilabeli secara seragam tanpa membedakan identitas individunya. Sebagai contoh, pada citra yang menampilkan beberapa manusia, semua piksel yang merepresentasikan manusia akan diberi label "manusia" tanpa perbedaan antara manusia satu dengan lainnya. Sementara itu, *instance segmentation* merupakan pengembangan lebih lanjut karena tidak hanya memberikan label kelas, tetapi juga membedakan setiap objek individual meskipun berasal dari kelas yang sama. Dengan demikian, *instance segmentation* dapat dianggap sebagai kombinasi antara deteksi objek dan segmentasi semantik, karena mampu menunjukkan lokasi objek sekaligus batas detail setiap individu (Molina et al., 2025). Perbedaan visual ketiganya dapat dilihat pada Gambar 2. Sementara

ringkasan perbedaan kunci antara *semantic segmentation* dan *instance segmentation* disajikan pada Tabel 1.



Gambar 2. Perbedaan antara (a) object detection, (b) semantic segmentation, (c) instance segmentation

Tabel 1. Perbandingan Kedua Jenis Object Segmentation

Aspek	<i>Semantic Segmentation</i>	<i>Instance Segmentation</i>
Definisi	Memberikan label kelas pada setiap piksel dalam citra	Memberikan label kelas sekaligus membedakan setiap objek individu
Output	Peta piksel per kelas (semua objek kelas sama dianggap seragam)	Peta piksel per kelas dengan identitas berbeda untuk tiap objek
Tingkat Detail	Fokus pada kategori objek, tanpa pembedaan antar objek individu	Detail lebih tinggi, dapat menghitung dan membedakan tiap objek
Kelebihan	Lebih sederhana, cocok untuk analisis distribusi kelas	Lebih informatif, cocok untuk aplikasi yang butuh jumlah dan posisi spesifik
Keterbatasan	Tidak dapat membedakan antar objek dalam kelas yang sama	Lebih kompleks, membutuhkan komputasi lebih tinggi

1.2.4 Python

Python adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido van Rossum pada tahun 1991 dan terus dikembangkan oleh Python Software Foundation. Python memiliki sintaks yang sederhana dan jelas, sehingga mudah dipelajari dan digunakan oleh pemula maupun oleh pengembang profesional. Python dapat digunakan untuk berbagai kebutuhan pengembangan perangkat lunak, mulai dari aplikasi web, pengolahan data, sampai machine learning dan kecerdasan buatan. Ketersediaan pustaka (*library*) dan modul *built-in* yang sangat lengkap juga menjadi keunggulan Python, sehingga *programmer* dapat dengan

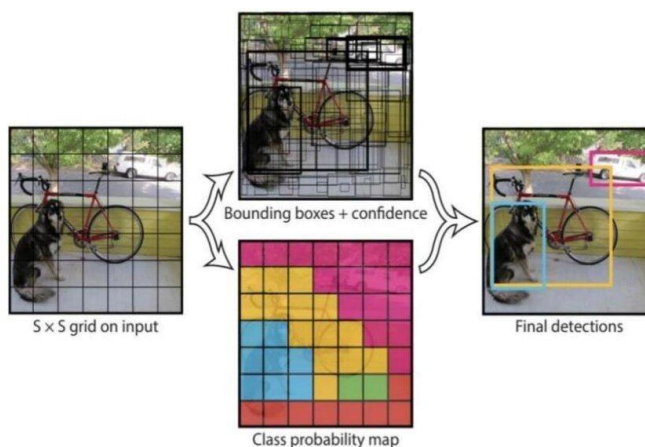
mudah memanfaatkan berbagai fungsi dan fitur tanpa harus menulis kode dari awal (Dhruv et al., 2022).

Python banyak digunakan di bidang pengembangan web, kecerdasan buatan, big data, analisis statistik, dan otomasi sistem. Python juga populer di kalangan akademisi karena kemampuannya untuk menyelesaikan berbagai masalah dengan efisiensi dan fleksibilitas tinggi (Cervantes-Barraza, 2025).

1.2.5 YOLO (*You Only Look Once*)

YOLO (*You Only Look Once*) adalah sebuah metode deteksi objek berbasis *deep learning* yang dikembangkan untuk mengatasi keterbatasan metode deteksi objek sebelumnya yang biasanya menggunakan pendekatan dua tahap, seperti R-CNN. YOLO mengubah permasalahan deteksi objek menjadi masalah regresi tunggal yang memprediksi kotak pembatas (*bounding box*) dan kelas objek secara langsung dari gambar input dalam satu kali proses jaringan saraf konvolusional (CNN) yang terpadu dan *end-to-end*.

Pada dasarnya, YOLO membagi gambar input menjadi *grid* $S \times S$, di mana setiap sel *grid* bertanggung jawab mendeteksi objek yang pusatnya berada di dalam sel tersebut. Setiap sel memprediksi beberapa *bounding box* dengan koordinat posisi relatif serta skor kepercayaan yang mencerminkan probabilitas keberadaan objek dan kualitas prediksi kotak tersebut. Selain itu, setiap sel juga memprediksi probabilitas kelas objek secara kondisional. Dengan cara ini, deteksi objek dilakukan secara simultan dalam satu proses tanpa memerlukan tahap-tahap terpisah, sehingga menghasilkan kecepatan prediksi yang sangat tinggi dan efisiensi komputasi yang baik, terutama untuk aplikasi *real-time* (Redmon et al., 2016). Skema alur ini divisualisasikan pada Gambar 3, yang memperlihatkan pembagian *grid*, prediksi kotak dan kelas oleh tiap sel, serta konsolidasi hasil deteksi.



Gambar 3. Gambaran umum cara kerja YOLO

YOLO terus mengalami perkembangan dari versi awal hingga versi terkini, dengan peningkatan akurasi dan efisiensi yang signifikan sambil mempertahankan kecepatan deteksi yang *real-time*. Arsitektur dasarnya mengandalkan jaringan konvolusional dalam framework Darknet, dengan berbagai optimasi pada lapisan konvolusional dan fungsi aktivasi untuk meningkatkan performa model (Zhao & Chen, 2024).

1.2.6 APEXEL NV008 *Night Vision Binoculars*

Kamera teropong Apexel NV008 *Infrared Night Vision Binoculars* adalah perangkat penglihatan malam berbasis inframerah yang mengusung sistem dua saluran (*dual-channel night vision system*) untuk menghasilkan tampilan stereoskopis dengan kedalaman visual yang realistis. Contoh wujud perangkat pada Gambar 4. Desainnya menggunakan material komposit yang ringan namun tahan lama, serta didukung oleh sistem optik berlapis ganda (*multi-coated optics*) untuk meningkatkan kejernihan dan kontras gambar. Selain itu, fitur *manual gain control* memungkinkan pengguna mengatur tingkat kecerahan gambar agar tetap optimal pada berbagai kondisi pencahayaan malam hari.



Gambar 4. APEXEL NV008 *Night Vision Binoculars*

Pengambilan citra dilakukan menggunakan kamera digital ber-perbesaran tinggi (kamera teropong) dengan zoom optik 4× dan zoom digital hingga 12×, yang mampu menghasilkan foto beresolusi maksimum 1280×720 piksel. Dengan kemampuan tersebut, perangkat ini sangat efektif digunakan untuk aktivitas pengamatan malam hari, seperti studi perilaku satwa liar atau pemantauan area terbuka dengan visibilitas terbatas. (Apexel, 2025).

1.2.7 Evaluasi Model

Evaluasi model adalah tahap penting untuk mengukur kinerja model yang telah dilatih. Berikut ini adalah metrik evaluasi yang digunakan dalam penelitian ini, yang menunjukkan seberapa baik model tersebut dalam memprediksi data yang belum pernah dilihat sebelumnya:

Confusion Matrix. *Confusion Matrix* adalah tabel untuk menilai kinerja model klasifikasi dengan membandingkan prediksi model terhadap label aktual pada data uji. Tabel ini membantu mengidentifikasi pola keberhasilan dan kesalahan prediksi secara lebih terperinci dengan merincikan hasil per kelas. *Confusion Matrix* mencakup:

- *True Positive* (TP): Prediksi positif yang benar.
- *True Negative* (TN): Prediksi negatif yang benar.
- *False Positive* (FP): Prediksi positif yang keliru (seharusnya negatif).
- *False Negative* (FN): Prediksi negatif yang keliru (seharusnya positif).

F1-score. *F1-score* merupakan metrik evaluasi yang membandingkan rata-rata antara nilai *precision* dan *recall*. *Precision* merupakan rasio antara jumlah prediksi positif yang benar (*true positive* / TP) terhadap total prediksi positif yang dihasilkan oleh model, yaitu kombinasi antara *true positive* (TP) dan *false positive* (FP). Metrik ini menunjukkan tingkat ketepatan model dalam menghasilkan prediksi positif yang benar (Clara dkk., 2021).

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

Recall merupakan rasio antara jumlah prediksi positif yang benar (*true positive* / TP) terhadap keseluruhan data yang sebenarnya positif, yaitu gabungan antara *true positive* (TP) dan *false negative* (FN). Metrik ini menunjukkan kemampuan model dalam mengenali atau menemukan kembali seluruh data yang termasuk dalam kelas positif (Clara dkk., 2021).

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

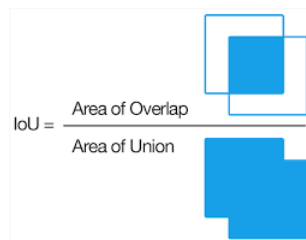
Nilai *F1-score* berada pada rentang antara 0 hingga 1, di mana nilai 1 menunjukkan kinerja model yang sangat baik, sedangkan nilai 0 menunjukkan kinerja yang buruk. Semakin mendekati nilai 1, semakin baik kemampuan model dalam menyeimbangkan antara *precision* dan *recall* (Kusuma dkk., 2021). Berikut persamaan F1 score:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Akurasi. Akurasi merupakan metrik evaluasi yang menunjukkan proporsi prediksi model yang benar dibandingkan dengan keseluruhan jumlah data yang diuji. Dengan kata lain, akurasi dihitung dengan membagi jumlah sampel yang berhasil diklasifikasikan dengan benar oleh sistem dengan total jumlah sampel. Metrik ini umum digunakan karena memberikan gambaran sederhana dan intuitif mengenai kinerja model. Berikut persamaan akurasi:

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

Intersection over Union (IoU). *Intersection over Union* (IoU) merupakan metrik evaluasi yang digunakan untuk mengukur tingkat kesesuaian *mask* hasil prediksi model dengan area sebenarnya (*ground truth*) pada segmentasi objek. IoU menghitung rasio antara luas area irisan (*intersection*) dari hasil prediksi dan *ground truth* terhadap luas area gabungan (*union*) keduanya. Rumus perhitungan IoU ditunjukkan pada Gambar 5.



$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Gambar 5. Perhitungan IoU

Dice Score. *Dice Score* atau biasa dikenal dengan *Dice Similarity Coefficient* (DSC), merupakan metrik evaluasi yang digunakan untuk mengukur tingkat kesesuaian *mask* hasil prediksi model dengan *ground truth*. Tetapi, berbeda dengan IoU yang menggunakan luas gabungan, *Dice Score* menggunakan rata-rata luas area prediksi dan *ground truth* sebagai pembanding sehingga memberikan bobot lebih besar kepada irisan. Hal ini membuat *Dice Score* lebih toleran terhadap kesalahan kecil pada batas objek (Hirling et al., 2023) serta lebih sensitif terhadap perubahan pada area kecil yang mungkin diabaikan oleh IoU. Rumus perhitungan Dice Score ditunjukkan pada Gambar 6.

$$\text{Dice} = \frac{2 \times \text{Area of overlap}}{\text{Total area}} = \frac{2 \times \text{Prediction} \cap \text{Ground truth}}{\text{Prediction} \cup \text{Ground truth}}$$

Gambar 6. Perhitungan *Dice Score*

1.3 Tujuan dan Manfaat

1.3.1 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan dua metode *instance segmentation*, yaitu YOLOv8-seg dan YOLO11-seg, untuk sistem deteksi kematangan TBS kelapa sawit secara langsung dari pohon sawit.
2. Melakukan analisis komparatif terhadap kinerja YOLOv8-seg dan YOLO11-seg dalam menghadapi tantangan nyata di lapangan, seperti objek yang tertutup sebagian dan medan yang sulit dijangkau.

1.3.2 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut:

1. Memberikan analisis komparatif terkait efektivitas dan efisiensi model YOLOv8-seg dan YOLO11-seg dalam mendeteksi kematangan TBS kelapa sawit.
2. Menyediakan alternatif teknologi berbasis *computer vision* dari kejauhan dengan menggunakan kamera teropong yang dapat menggantikan metode konvensional dalam menilai kematangan buah sawit.
3. Mengidentifikasi model segmentasi objek yang lebih tangguh dalam menghadapi masalah *occlusion*, yaitu objek tandan sawit yang tertutupi dan/atau saling menutupi.
4. Mengoptimalkan proses pemanenan kelapa sawit, untuk mendukung *Decision Support System* (DSS) waktu panen TBS bagi petani.

BAB II

METODE PENELITIAN

2.1 Tempat dan Waktu

Penelitian dilakukan selama lima bulan dimulai dari Juni 2025 hingga November 2025. Pengambilan data dilakukan sebanyak dua kali, yaitu di kebun sawit Polewali Mandar pada 11 Mei 2025 dan di Pinrang pada 24 Mei 2025. Adapun untuk penelitian dan analisis dilakukan di Laboratorium *Artificial Intelligence and Multimedia Processing* (AIMP), Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin, seperti pada Gambar 7.



Gambar 7. Lokasi Penelitian (a) Kebun Sawit Polewali Mandar; (b) Kebun Sawit Pinrang; (c) Laboratorium AIMP Universitas Hasanuddin

2.2 Benda Uji dan Alat

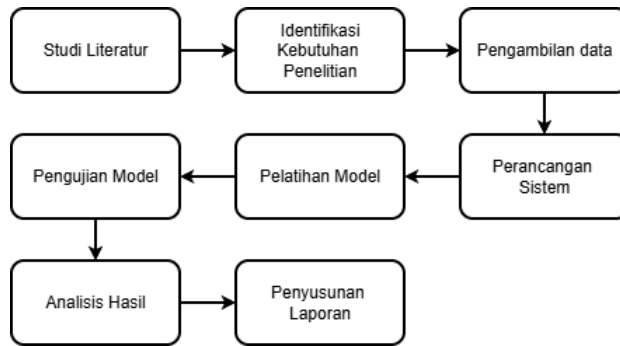
Benda dan alat yang digunakan dalam penelitian ini meliputi perangkat lunak dan perangkat keras berikut:

1. Perangkat lunak
 - a. Operating System: Windows 11 64-bit
 - b. Roboflow
 - c. Kaggle Notebooks
 - d. Visual Studio Code
 - e. Microsoft Excel

- f. Microsoft Word
- g. Microsoft Powerpoint
- h. draw.io
- i. Python
- 2. Perangkat keras
 - a. Laptop dengan processor 13th Gen Intel(R) Core(TM) i7-13650HX 2.60 GHz, RAM 20 GB, SSD 1TB
 - b. Apexel NV008 *Infrared Night Vision Binoculars*
- 3. Bahasa pemrograman
 - a. Python v3.11.13
- 4. Library
 - a. Pandas v1.5.3 oleh *The Pandas Development Team* berguna untuk manipulasi dan analisis data, terutama untuk data berbentuk tabel (*dataframe*) dengan fitur seperti *filtering*, *grouping*, dan transformasi data.
 - b. Pillow (PIL) v9.5.0 oleh *Alex Clark dan tim Pillow* berguna untuk memproses dan memanipulasi gambar seperti membuka, mengubah ukuran, memotong, serta mengonversi format gambar.
 - c. OpenCV-Python (cv2) v4.7.0 oleh *OpenCV Team* berguna untuk pengolahan citra dan video, termasuk deteksi objek, segmentasi, transformasi gambar, serta analisis video.
 - d. Ultralytics v8.0.0 oleh *Ultralytics LLC* berguna untuk pemodelan deteksi objek menggunakan YOLO (*You Only Look Once*) framework, termasuk pelatihan, validasi, dan inferensi model deteksi objek modern.
 - e. IPython v8.12.0 oleh *IPython Development Team* berguna untuk menyediakan lingkungan interaktif berbasis Python, serta mendukung tampilan gambar, tabel, dan grafik di notebook.
 - f. NumPy v1.22.4 oleh *Travis E. Oliphant et al.* Berguna untuk menyediakan dukungan array dan matriks multidimensi serta fungsi matematika untuk operasi numerik pada array.
 - g. Matplotlib v3.7.1 oleh *John D. Hunter dan Michael Droettboom* berguna untuk visualisasi data, memungkinkan pengguna membuat grafik garis, batang, histogram, scatter, dan berbagai tipe plot lainnya.
 - h. Seaborn v0.12.2 oleh *Michael Waskom dan kontributor* berguna untuk menyediakan antarmuka tingkat tinggi untuk visualisasi statistik yang menarik dan informatif, dibangun di atas Matplotlib.

2.3 Tahapan penelitian

Penelitian dilakukan dengan beberapa tahapan seperti yang ditunjukkan pada Gambar 8 berikut:



Gambar 8. Tahapan Penelitian. Diagram ini menunjukkan alur penelitian mulai dari studi literatur, identifikasi kebutuhan penelitian, pengambilan data, perancangan sistem, pelatihan model, pengujian model, analisis hasil, hingga penyusunan laporan akhir.

2.4 Teknik Pengambilan Data

Data yang digunakan dalam penelitian ini merupakan hasil gabungan dari beberapa jenis sumber data yang berbeda. Teknik pengambilan data dibagi menjadi dua kategori utama, yaitu data primer dan data sekunder. Kedua jenis data tersebut kemudian digabungkan, diseleksi, dan diproses lebih lanjut untuk menghasilkan dataset yang siap digunakan dalam proses pelatihan model.

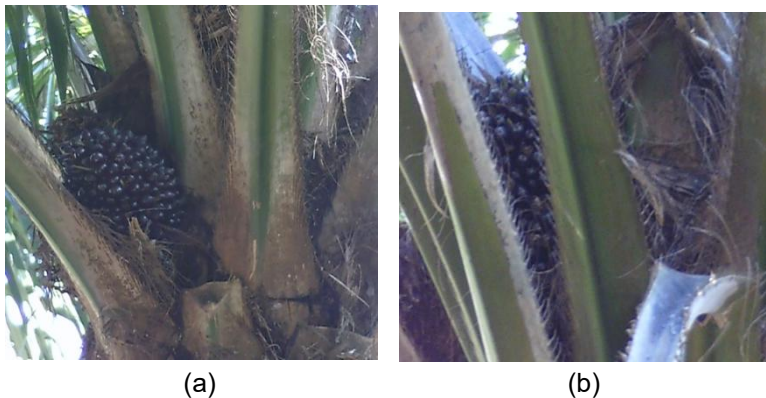
2.4.1 Data primer

Data primer diperoleh langsung oleh peneliti dari sumber yang relevan dengan topik penelitian. Pada studi ini, pengumpulan data primer dilakukan dengan memotret pohon kelapa sawit di lokasi kebun, dengan fokus utama pada tandan buah segar (TBS). Pengambilan citra menggunakan kamera digital dengan kemampuan perbesaran optik tinggi (kamera teropong), yakni *zoom* optik 4× dan *zoom* digital 12×, yang mampu menghasilkan foto beresolusi maksimum 1280×720 piksel. Spesifikasi tersebut memungkinkan pengambilan citra dari jarak jauh tanpa perlu mendekati pohon secara langsung. Ilustrasi proses akuisisi data ditunjukkan pada Gambar 9.



Gambar 9. Ilustrasi Pengambilan Data

Pengambilan citra pohon kelapa sawit dilakukan pada siang hari dengan kondisi cuaca cerah dari jarak minimal 3 meter dari tandan, dengan kamera diarahkan langsung ke target pada pohon. Jarak minimum ini disesuaikan dengan karakteristik pembesaran/ruang kerja kamera yang optimal pada rentang tersebut. Proses akuisisi juga memperhatikan variasi keterhalangan (*occlusion*), misalnya tandan yang sebagian tertutup pelepah, daun, atau tandan lain. Contoh data primer ditunjukkan pada Gambar 10.



Gambar 10. Contoh Dataset Primer (a) Contoh Dataset Primer Tidak Tertutupi; (b) Contoh Dataset Primer Tertutupi

2.4.2 Data sekunder

Data sekunder pada penelitian ini berasal dari hasil penelitian sebelumnya yang dihimpun oleh Muhammad Ljilal Nurhadi. Dataset tersebut berupa kumpulan citra untuk pengembangan sistem deteksi estimasi berat tandan buah segar (TBS) kelapa sawit berbasis YOLO dan *Random Forest Regression*, dengan satu kelas objek, yakni tandan yang masih berada di pohon. Pengunduhan dataset dilakukan atas persetujuan peneliti pembuatnya. Dalam studi ini, hanya citra tandan pada

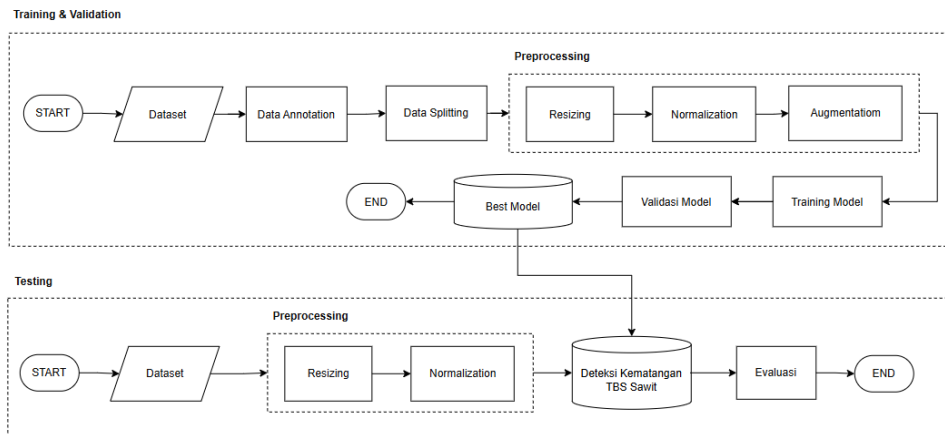
pohon yang digunakan tanpa mengambil data anotasi asli. Sebanyak 201 citra diunduh dan selanjutnya dilakukan proses pelabelan ulang sesuai kebutuhan kelas penelitian ini (Nurhadi, 2024). Contoh dataset karya Muhammad Ijlal Nurhadi ditampilkan pada Gambar 11.



Gambar 11. Contoh dataset Nurhadi, 2024

2.5 Perancangan dan Implementasi Sistem

Perancangan sistem disusun untuk menetapkan langkah-langkah teoritis dan konseptual yang menggambarkan sistem yang akan dikembangkan. Alur proses penelitian ini disajikan pada Gambar 12.



Gambar 12. Alur rancangan sistem

2.5.1 Tahap Training

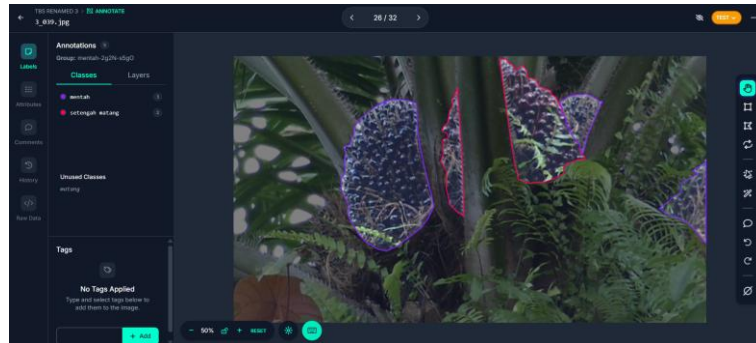
1. Input data

Langkah awal penelitian ini adalah menyiapkan data citra objek yang akan dideteksi untuk keperluan pelatihan dan pengujian sistem. Data yang digunakan mencakup data primer berupa 212 citra yang mengandung tandan serta 125 citra latar (*background*) tanpa tandan, dan data sekunder

sebanyak 201 citra. Seluruh data tersebut kemudian diproses dan dibagi pada tahap berikutnya.

2. Anotasi data (*data annotation*)

Setelah tahap input data, dilakukan pelabelan menggunakan anotasi poligon (*segmentation*) dengan bantuan Roboflow. Anotasi mencakup tiga kelas, yaitu kelas matang, setengah matang, dan mentah. Setiap poligon mengikuti tepi objek sehingga area tertutup merepresentasikan bagian yang relevan. Contoh anotasi disajikan pada Gambar 13.



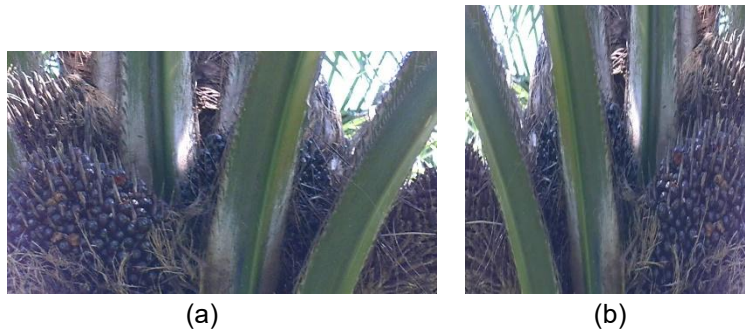
Gambar 13. Anotasi data menggunakan Roboflow

3. Pembagian data (*data splitting*)

Setelah tahap anotasi, data dibagi menjadi tiga kelompok, yaitu data latih, validasi, dan uji. Dari total 768 citra, 538 citra (70%) digunakan sebagai data latih, 115 citra (15%) sebagai data validasi, dan 115 citra (15%) sebagai data uji. Pembagian dilakukan secara manual terutama untuk mencegah kemunculan citra yang sama persis di himpunan latih, sekaligus menjaga keragaman sampel pada proses pelatihan. Seluruh data sekunder serta citra *background* tanpa tandan ditempatkan hanya pada data latih.

4. *Preprocessing*

Setelah pembagian data, dilakukan tahap prapemrosesan yang mencakup *resize*, normalisasi, dan augmentasi. *Resize* adalah proses mengubah dimensi citra ke resolusi tertentu menggunakan metode interpolasi. Pada penelitian ini, seluruh citra diubah menjadi 640×640 piksel agar kinerja model menjadi lebih baik dengan ukuran data input yang konsisten. Proses *resize* diterapkan pada data latih, validasi, dan uji. Contoh hasil *resize* ditampilkan pada Gambar 14.

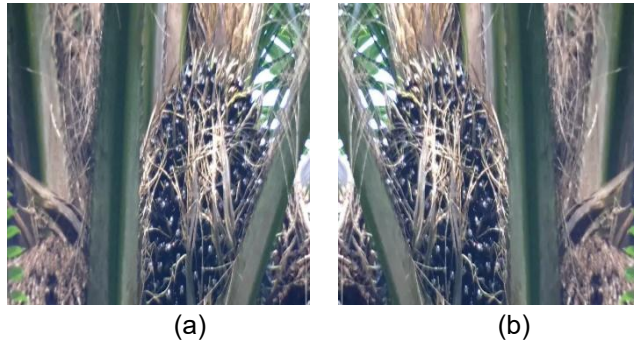


Gambar 14. Contoh hasil *resize* data (a) sebelum *resize*, (b) sesudah *resize*

Normalisasi adalah proses menskalakan intensitas piksel agar berada pada rentang nilai yang seragam. Dalam penelitian ini, nilai piksel citra dinormalisasi dari $[0, 255]$ menjadi $[0, 1]$ dengan membagi setiap piksel dengan 255. Normalisasi membantu stabilitas optimisasi dan percepatan konvergensi model. Proses normalisasi juga diterapkan pada data latih, validasi, dan uji.

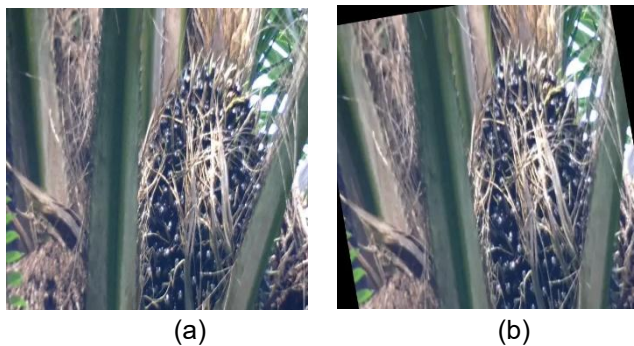
Setelah itu, dilakukan proses augmentasi yang bertujuan untuk meningkatkan keragaman data latih tanpa menambah pengambilan data baru dan mengurangi terjadi resiko *overfitting*. Augmentasi yang digunakan meliputi *horizontal flip* dan *shiftscalerotate*. Augmentasi hanya diterapkan pada data latih, sedangkan data validasi dan uji dibiarkan apa adanya untuk menjaga objektivitas evaluasi. Total data latih setelah diaugmentasi berjumlah 1955 citra.

Horizontal Flip. *Horizontal flip* adalah teknik augmentasi data yang membalik citra secara simetris terhadap sumbu vertikal, sehingga sisi kanan menjadi kiri dan sebaliknya. Pada segmentasi objek, transformasi harus diterapkan serentak pada citra dan *mask* agar batas objek tetap sejajar; untuk anotasi poligon, titik-titik kontrol ikut dicerminkan ke posisi baru. Flip umumnya diterapkan hanya pada data latih dengan probabilitas, guna menambah variasi orientasi tanpa mengubah label kelas. Praktik ini membantu menekan *overfitting* dan meningkatkan generalisasi. Contoh citra hasil *horizontal flip* ditampilkan pada Gambar 15.



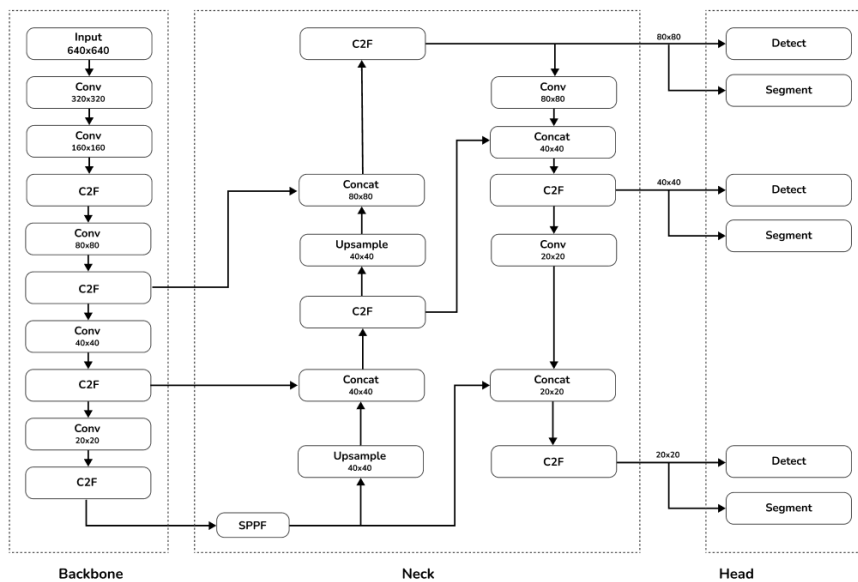
Gambar 15. Contoh hasil *horizontal flip*: (a) sebelum *flip*; (b) sesudah *flip*

ShiftScaleRotate. *ShiftScaleRotate* adalah teknik augmentasi yang menggeser posisi (*shift/translation*), mengubah skala (*zoom in/out*), dan memutar citra dalam satu langkah. Pada segmentasi objek, transformasi harus diterapkan serentak pada citra dan *mask*; untuk anotasi poligon, seluruh titik kontrol digeser, diskalakan, dan diputar dengan cara yang sama agar kontur tetap tepat. Augmentasi ini umumnya digunakan pada data latih untuk memperkaya variasi posisi, ukuran, dan orientasi sehingga membantu menekan *overfitting* dan meningkatkan generalisasi. Contoh citra hasil *ShiftScaleRotate* ditampilkan pada Gambar 16.



Gambar 16. Contoh hasil *ShiftScaleRotate*: (a) sebelum *ShiftScaleRotate*; (b) sesudah *ShiftScaleRotate*

5. Model segmentasi objek YOLOv8-seg



Gambar 17. Arsitektur YOLOv8-seg

Berdasarkan Gambar 17, YOLOv8-seg terdiri dari tiga komponen utama, yaitu *Backbone*, *Neck*, dan *Head*. Masing-masing komponen memiliki peran penting dalam proses segmentasi dan klasifikasi objek.

a. *Backbone*

Backbone berfungsi sebagai ekstraktor fitur utama dari citra masukan. Proses dimulai dengan citra berukuran 640×640 piksel yang terlebih dahulu melewati serangkaian lapisan konvolusi awal (*Conv layer*) yang disebut *Stem*. Lapisan ini berperan dalam mengurangi resolusi spasial secara bertahap sembari meningkatkan kedalaman channel, sehingga menghasilkan representasi fitur yang lebih kaya. Blok utama dari *Backbone* adalah modul C2F (*Cross Stage Partial* dengan dua konvolusi), yang merupakan pengembangan dari blok CSP (*Cross Stage Partial*). Modul C2F secara efisien membagi dan memproses aliran fitur untuk mempelajari representasi fitur yang kompleks tanpa menambah beban komputasi secara signifikan. Secara bertahap, *Backbone* melakukan proses *downsampling* untuk menghasilkan *feature map* pada berbagai skala. *Feature map* terdangkal memiliki detail spasial yang lebih baik, sedangkan *feature map* terdalam mengandung informasi semantik yang lebih kaya. Kombinasi keduanya memungkinkan model untuk memahami baik konteks maupun detail objek secara menyeluruh.

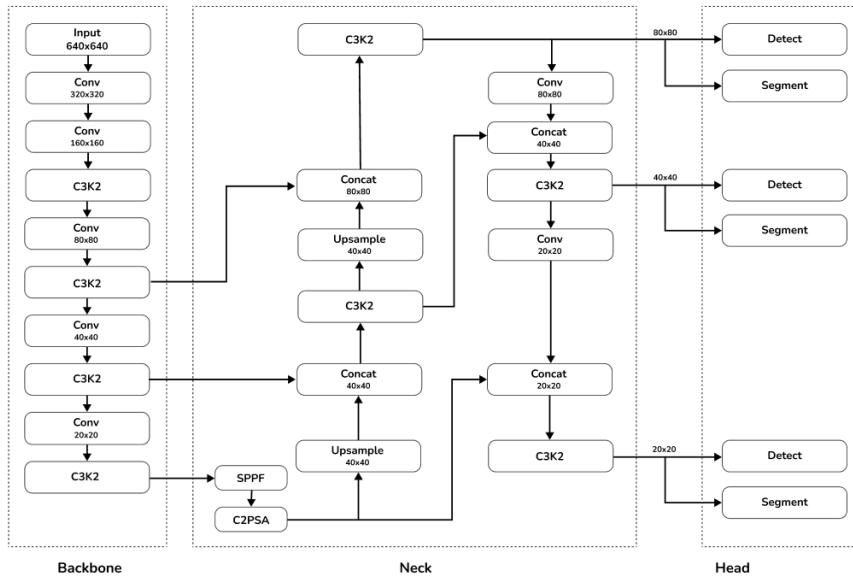
b. *Neck*

Neck berfungsi untuk melakukan fusi fitur (*feature fusion*), yaitu penggabungan informasi dari berbagai level jaringan *Backbone*. Arsitektur yang digunakan pada bagian ini adalah PANet (*Path Aggregation Network*), yang dirancang untuk meningkatkan aliran informasi antara lapisan dengan resolusi berbeda. Proses pada *Neck* diawali dengan pengambilan *feature map* terdalam (berukuran 20×20) dari *Backbone*, yang memiliki informasi semantik paling tinggi. *Feature map* ini kemudian diproses melalui modul SPPF (*Spatial Pyramid Pooling Fast*), yang bertujuan untuk menangkap konteks multi-skala dengan melakukan *max pooling* menggunakan beberapa ukuran kernel secara paralel, kemudian menggabungkan hasilnya. Pendekatan ini membuat model lebih tangguh terhadap variasi ukuran objek. Setelah tahap SPPF, PANet memulai jalur *bottom-up*, di mana *feature map* di-*upsample* dan digabungkan (*concatenate*) dengan *feature map* dari tingkat *Backbone* yang sesuai. Proses ini menggabungkan kembali informasi spasial yang detail (dari lapisan dangkal) ke dalam fitur yang kaya secara semantik (dari lapisan dalam). Setelah mencapai level tertinggi, proses dilanjutkan dengan jalur *top-down*, yang membawa informasi kembali ke lapisan bawah. Dengan demikian, hasil fusi dua arah ini memastikan bahwa setiap level fitur memiliki representasi yang lebih komprehensif dan saling memperkuat antara detail spasial serta konteks semantik.

c. *Head*

Head berfungsi untuk menghasilkan prediksi akhir berdasarkan *feature map* yang telah diperkaya oleh *Neck*. YOLOv8-seg menggunakan arsitektur *Decoupled Head*, di mana setiap keluaran dari tiga skala *Neck* diproses oleh dua cabang independen yang tidak berbagi bobot (*weight*). Cabang pertama adalah *Detection Head*, yang berfokus pada tugas deteksi objek. Pada cabang ini, lapisan konvolusi digunakan untuk memprediksi lokasi *bounding box* serta *confidence score* untuk setiap kelas objek. Cabang kedua adalah *Segmentation Head*, yang bertanggung jawab untuk melakukan *instance segmentation*. Cabang ini menghasilkan serangkaian *mask coefficients* untuk setiap objek terdeteksi, yang kemudian dikombinasikan dengan *mask prototypes* untuk membentuk *mask* piksel yang presisi. Pemisahan kedua tugas ini memungkinkan setiap *head* untuk berfokus pada fungsinya masing-masing dan dioptimalkan secara independen. Pendekatan *decoupled* tersebut meningkatkan akurasi serta efisiensi proses deteksi dan segmentasi pada berbagai skala objek.

6. Model segmentasi objek YOLO11-seg



Gambar 18. Arsitektur YOLO11-seg

Berdasarkan Gambar 18, YOLO11-seg terdiri dari tiga komponen utama, yaitu *Backbone*, *Neck*, dan *Head*. Masing-masing komponen memiliki peran penting dalam proses segmentasi dan klasifikasi objek.

a. *Backbone*

Backbone berfungsi untuk mengekstraksi fitur dari gambar input. Dimulai dengan gambar input 640x640, gambar pertama kali melewati serangkaian lapisan *Conv* (konvolusi) awal yang berfungsi sebagai "Stem". Lapisan ini secara progresif mengurangi resolusi spasial sambil menambah kedalaman channel. Selanjutnya, blok utama dari *Backbone* adalah modul C3K2. Ini adalah blok bangunan baru (evolusi dari blok CSP) yang menggantikan C2F pada arsitektur YOLOv8-seg. Blok ini dirancang untuk ekstraksi fitur yang efisien, yang memungkinkan model untuk mempelajari representasi fitur yang kaya dan kompleks. *Backbone* secara bertahap melakukan *downsampling* untuk menghasilkan *feature map* pada berbagai skala. *Feature map* dari lapisan dangkal kaya akan detail spasial sementara *feature map* dari lapisan dalam kaya akan informasi semantik (konteks).

b. *Neck*

Neck berfungsi untuk melakukan fusi fitur (*feature fusion*), yaitu menggabungkan informasi dari berbagai level jaringan *Backbone* agar representasi fitur menjadi lebih kaya dan adaptif. Proses pada *Neck* diawali dengan pengambilan *feature map* terdalam (berukuran 20×20) dari *Backbone* yang memiliki tingkat informasi semantik tertinggi. Salah satu perubahan signifikan pada versi YOLOv11 adalah penambahan

blok C2PSA (*Cross Stage Partial with Spatial Attention*). Blok ini berfungsi sebagai mekanisme atensi (*attention mechanism*) yang membantu model memfokuskan proses pembelajaran pada area fitur yang paling relevan serta mengurangi pengaruh *noise* atau informasi yang tidak penting, seperti akibat *occlusion* atau latar belakang kompleks. Selanjutnya, fitur yang dihasilkan dilewatkan melalui modul SPPF (*Spatial Pyramid Pooling Fast*) yang berperan untuk menangkap konteks multi-skala dengan melakukan *max pooling* menggunakan beberapa ukuran kernel secara paralel, kemudian menggabungkan hasilnya. Pendekatan ini meningkatkan kemampuan model dalam mengenali objek dengan berbagai ukuran. Setelah tahap SPPF, arsitektur PANet (*Path Aggregation Network*) melakukan proses fusi fitur dua arah. Jalur *bottom-up* dimulai dengan melakukan *upsampling* pada feature map dan menggabungkannya (*concatenate*) dengan *feature map* dari level *Backbone* yang sesuai. Proses ini menggabungkan kembali informasi spasial yang detail dari lapisan dangkal ke dalam fitur dengan konteks semantik yang lebih dalam. Setelah mencapai level tertinggi, jalur *top-down* kemudian membalikkan proses dengan membawa kembali informasi ke lapisan bawah. Dengan demikian, hasil akhir dari *Neck* memiliki representasi fitur yang menyeluruh, di mana setiap skala keluaran memuat keseimbangan antara detail spasial dan konteks semantik.

c. *Head*

Head bertanggung jawab untuk menghasilkan prediksi akhir berdasarkan *feature map* yang telah diperkaya oleh *Neck*. YOLOv11-seg mengadopsi arsitektur *Decoupled Head*, di mana setiap keluaran dari tiga skala *Neck* diproses melalui dua cabang yang independen dan tidak berbagi bobot (*weight*). Cabang pertama adalah *Detection Head*, yang berfokus pada tugas deteksi objek. Cabang ini memiliki lapisan konvolusi khusus untuk memprediksi lokasi *bounding box* serta *confidence score* untuk setiap kelas objek. Cabang kedua adalah *Segmentation Head*, yang bertanggung jawab terhadap proses *instance segmentation*. Pada cabang ini, model memprediksi serangkaian *mask coefficients* untuk setiap objek yang terdeteksi, yang kemudian digabungkan dengan *mask prototypes* guna menghasilkan mask piksel yang presisi. Pemilahan fungsi melalui *Decoupled Head* ini memungkinkan setiap cabang berfokus pada tugas spesifiknya serta dioptimalkan secara independen. Pendekatan tersebut meningkatkan performa model baik dalam aspek akurasi deteksi maupun ketepatan segmentasi pada berbagai skala objek.

7. Skenario pelatihan model

Dalam penelitian ini, dilakukan pelatihan terhadap tiga varian model YOLOv8-seg dan YOLO11-seg, yaitu *nano*, *small*, dan *medium*, untuk menentukan varian yang paling optimal dalam melakukan segmentasi dan

klasifikasi tingkat kematangan tandan buah segar (TBS) kelapa sawit. Varian *nano* memiliki keunggulan pada efisiensi komputasi dengan kebutuhan sumber daya yang rendah, varian *small* menawarkan keseimbangan antara efisiensi dan akurasi yang cukup tinggi, sementara varian *medium* menawarkan akurasi yang lebih tinggi tetapi memerlukan komputasi yang lebih besar.

Tabel 2. Spesifikasi teknis YOLOv8-seg varian *nano*, *small*, dan *medium*

Model	Size (pixels)	mAP (box)	mAP (mask)	Speed CPU ONNX	Speed A100 TensorRT	params	FLOPs
YOLOv8n-seg	640	36.7	30.5	96.1	1.21	3.44	12.6
YOLOv8s-seg	640	44.6	36.8	155.7	1.47	11.8	42.6
YOLOv8m-seg	640	49.9	40.8	317.0	2.18	27.3	110.2

Tabel 3. Spesifikasi teknis YOLO11-seg varian *nano*, *small*, dan *medium*

Model	Size (pixels)	mAP (box)	mAP (mask)	Speed CPU ONNX	Speed T4 TensorRT	params	FLOPs
YOLO11n-seg	640	38.9	32.0	65.9 ± 1.1	1.8 ± 0.0	2.9	9.7
YOLO11s-seg	640	46.6	37.8	117.6 ± 4.9	2.9 ± 0.0	10.1	33.0
YOLO11m-seg	640	51.5	41.5	281.6 ± 1.2	6.3 ± 0.1	22.4	113.2

Penelitian ini memilih varian *nano*, *small*, dan *medium* dari model YOLOv8-seg serta YOLO11-seg berdasarkan pertimbangan keseimbangan antara efisiensi komputasi, tingkat akurasi, dan keterbatasan sumber daya yang tersedia. Proses pelatihan dilakukan dengan memanfaatkan *dataset* primer dan sekunder yang telah dikumpulkan oleh peneliti, menghasilkan total 1.955 citra hasil augmentasi yang digunakan sebagai data pelatihan.

Pelatihan model dilakukan menggunakan pengaturan *hyperparameter* dengan jumlah *epoch* sebanyak 500, *batch size* 16, *optimizer* SGD, serta *learning rate* 0,01. Setelah proses pelatihan selesai, seluruh model diuji untuk menganalisis dan membandingkan kemampuan segmentasi pada data pengujian. Hasil segmentasi berupa *bounding box*, *segmentation mask*, *predicted class*, dan *confidence score*. Pendekatan ini diterapkan untuk memastikan bahwa performa deteksi dan segmentasi dapat dinilai secara objektif dengan mempertimbangkan variasi kondisi lingkungan, kecepatan pemrosesan, serta kapasitas komputasi yang tersedia.

2.5.2 Tahap Testing

Setelah menyelesaikan tahap pelatihan (*training*), keempat model akan menjalani tahap pengujian (*testing*) untuk menentukan model YOLO yang memberikan performa terbaik dalam penelitian ini. Proses pengujian dilakukan melalui tiga skenario yang dirancang untuk menilai kemampuan model dalam berbagai kondisi citra, yaitu:

1. Skenario 1

Pada skenario pertama, pengujian dilakukan menggunakan data uji berupa citra pohon kelapa sawit dengan tandan buah yang terlihat jelas atau terhalangi sekitar kurang dari 50%. Artinya, lebih dari setengah bagian tandan masih terlihat jelas pada citra. Kondisi ini merepresentasikan situasi yang relatif mudah bagi model dalam mendeteksi dan melakukan segmentasi objek. Contoh data uji untuk skenario 1 dapat dilihat pada Gambar 19 berikut.



Gambar 19. Contoh data uji skenario 1

2. Skenario 2

Pada skenario ketiga, pengujian dilakukan menggunakan data uji yang mengandung setidaknya satu tandan buah dengan tingkat keterhalangan sekitar lebih dari 50% dari ukuran aslinya. Pada skenario ini, hanya sebagian kecil tandan yang terlihat, menjadikannya kondisi paling menantang bagi model untuk mendeteksi dan melakukan segmentasi secara akurat. Contoh data uji untuk skenario 2 ditampilkan pada Gambar 20 berikut.



Gambar 20. Contoh data uji skenario 2