

# SawitScan: Aplikasi Android untuk Deteksi Kematangan dan Prioritas Panen Kelapa Sawit Berbasis YOLO26 dan Metode SAW

<sup>1</sup>Juriawan Raja Saputra, <sup>2</sup>Irma Salamah, <sup>3</sup>Ahmad Taqwa

<sup>1,2,3</sup>Teknik Elektro, Politeknik Negeri Sriwijaya, Palembang, Indonesia

\*Korespondensi: [irma\\_salamah@polsri.ac.id](mailto:irma_salamah@polsri.ac.id)

Submit : 31 Mei 2026 | Diterima : 28 Jun 2026 | Terbit : 01 Sept 2026

## ABSTRACT

*Determining oil palm fresh fruit bunch (FFB) ripeness in the field is still largely done manually, making the process subjective and prone to harvesting errors, while prioritizing which area to harvest first is rarely done systematically. This study develops and evaluates SawitScan, an Android application combining on-device YOLO26 object detection for ripeness classification (ripe, underripe, unripe) via the smartphone camera, automatic GPS logging per detection, and a Simple Additive Weighting (SAW) system that computes a harvest-priority score per location cluster. The YOLO26n variant was trained on a merged dataset of 1,124 images for its lightweight architecture and end-to-end, NMS-free design suited to mobile deployment, achieving competitive accuracy (89.8% precision, 87.3% recall, 92.1% mAP@0.5). Independent evaluation of the embedded TFLite model on a held-out test split, replicating the application's exact pipeline, yielded mAP@0.5 of 0.748 and F1-score of 0.743. The SAW system ranks harvest areas by combined ripe, underripe, and unripe counts per cluster, weighted 0.6, 0.3, and 0.1 by harvest urgency. Results show on-device detection, GPS-based mapping, and decision support can be practically combined on a mobile device for more objective and efficient oil palm harvest planning. Future work may extend the dataset to additional plantation locations and validate the SAW-based ranking against actual field harvest outcomes.*

**Keywords:** Decision Support System; Object Detection; Oil Palm Ripeness; Simple Additive Weighting; YOLO26

## ABSTRAK

Penentuan kematangan tandan buah segar (TBS) kelapa sawit di lapangan masih banyak dilakukan secara manual, sehingga bersifat subjektif dan rawan menimbulkan kesalahan panen, sementara penentuan area yang diprioritaskan untuk dipanen lebih dahulu juga belum dilakukan secara sistematis. Penelitian ini mengembangkan dan mengevaluasi SawitScan, aplikasi Android yang menggabungkan deteksi objek YOLO26 untuk klasifikasi kematangan (*ripe*, *underripe*, *unripe*) secara *on-device* melalui kamera *smartphone*, pencatatan koordinat GPS otomatis tiap deteksi, dan sistem pendukung keputusan Simple Additive Weighting (SAW) untuk menghitung skor prioritas panen tiap klaster lokasi. Model YOLO26n dilatih pada *dataset* gabungan 1.124 citra karena arsitekturnya yang ringan dan desain *end-to-end* tanpa *Non-Maximum Suppression* yang sesuai untuk implementasi *mobile*, dengan akurasi kompetitif (*precision* 89,8%, *recall* 87,3%, *mAP@0.5* 92,1%). Evaluasi independen model akhir yang ter-embed pada aplikasi (TFLite) pada data uji menghasilkan *mAP@0.5* 0,748 dan *F1-score* 0,743. Sistem SAW memeringkat area panen berdasarkan kombinasi jumlah deteksi *ripe*, *underripe*, dan *unripe* per klaster, dengan bobot 0,6, 0,3, dan 0,1 sesuai urgensi panen. Hasil penelitian menunjukkan deteksi objek *on-device*, pemetaan berbasis GPS, dan sistem pendukung keputusan dapat diterapkan praktis pada perangkat *mobile* untuk mendukung perencanaan panen kelapa sawit yang lebih objektif dan efisien. Penelitian selanjutnya dapat diarahkan pada perluasan *dataset* ke berbagai lokasi perkebunan serta validasi hasil pemeringkatan SAW terhadap data realisasi panen aktual di lapangan.

**Kata Kunci:** Deteksi Objek; Kematangan Kelapa Sawit; Simple Additive Weighting; Sistem Pendukung Keputusan; YOLO26

## PENDAHULUAN

Kelapa sawit merupakan salah satu komoditas perkebunan strategis dengan peran besar dalam perekonomian Indonesia, mulai dari penghasil devisa, penggerak ekonomi kerakyatan, hingga penyerap tenaga kerja terbesar di sektor pertanian (BPDP, 2025; Mohamad Zaki et al., 2025). Skala industri yang besar ini menuntut pengelolaan yang efisien, sehingga penerapan teknologi digital pada perkebunan kelapa sawit menjadi kebutuhan yang semakin mendesak, terutama pada aspek pemetaan lahan dan penentuan kematangan buah (Mohamad Zaki et al., 2025).

Salah satu aspek kritis dalam proses panen adalah penentuan tingkat kematangan tandan buah segar (TBS), karena kematangan buah memengaruhi langsung kandungan minyak dan kualitas produksi. Hingga saat ini, penentuan kematangan di lapangan masih banyak dilakukan secara manual oleh pekerja panen berdasarkan warna tandan dan jumlah brondolan yang jatuh. Cara ini bersifat subjektif, sangat bergantung pada kejelian pengamat, dan rawan menimbulkan kesalahan panen berupa buah mentah yang ikut terpanen atau buah matang yang tertinggal di pohon.

Berbagai pendekatan *computer vision* dan integrasi geospasial telah diusulkan untuk mengatasi keterbatasan metode manual tersebut, sebagaimana diuraikan berikut ini. Namun, sebagian besar penelitian yang ada masih memperlakukan deteksi kematangan buah dan informasi lokasi secara terpisah, atau memerlukan koneksi server/internet yang tidak selalu tersedia di area perkebunan. Sejauh penelusuran penulis, belum ada penelitian yang mengintegrasikan deteksi kematangan TBS berbasis YOLO26 yang berjalan *on-device*, pencatatan koordinat GPS otomatis, dan metode pendukung keputusan multikriteria untuk menentukan prioritas area panen dalam satu aplikasi *mobile* yang dapat bekerja tanpa koneksi internet.

Berdasarkan kesenjangan tersebut, penelitian ini mengusulkan dan mengevaluasi SawitScan, sebuah aplikasi Android yang menggabungkan tiga komponen: (1) deteksi dan klasifikasi kematangan TBS ke dalam tiga kelas (*ripe*, *underripe*, *unripe*) menggunakan model YOLO26 yang berjalan *on-device*; (2) pencatatan dan pemetaan koordinat GPS untuk setiap hasil deteksi; dan (3) metode Simple Additive Weighting (SAW) untuk menghitung skor dan pemeringkatan prioritas area panen pada setiap kluster lokasi. Kontribusi penelitian ini meliputi: penerapan model YOLO26n yang ringan dan efisien sebagai basis deteksi dan klasifikasi kematangan TBS yang sesuai untuk *deployment on-device*; evaluasi akurasi independen terhadap model akhir yang benar-benar ter-embed pada aplikasi, diukur pada data uji yang tidak pernah dilihat selama pelatihan; serta penerapan SAW sebagai mekanisme pendukung keputusan prioritas area panen berbasis hasil deteksi *real-time*. Bagian selanjutnya dari makalah ini menjelaskan metode penelitian, hasil dan pembahasan, serta kesimpulan penelitian.

Berbagai pendekatan *computer vision* telah diusulkan untuk mengatasi keterbatasan metode manual tersebut, sebagaimana dirangkum dalam tinjauan literatur sistematis terhadap metode deteksi kematangan TBS (Ismail et al., 2023). Pendekatan awal berbasis *machine learning* klasik, seperti *K-Means Clustering* dan *K-Nearest Neighbor* dengan fitur warna, bentuk, dan tekstur, menunjukkan hasil yang menjanjikan namun masih bergantung pada fitur yang dirancang secara manual dan sensitif terhadap variasi citra lapangan (Guslianto, 2023; Himmah et al., 2020; Sari et al., 2022; Suriani & Lisnawita, 2024). Pendekatan *deep learning* berbasis *Convolutional Neural Network* (CNN), seperti ResNet50 dan ResNet18, kemudian terbukti mampu mengekstraksi fitur secara otomatis dan meningkatkan akurasi klasifikasi kematangan TBS (Islah et al., 2025; Setiawan et al., 2025; Zulkarnain et al., 2024). Pada ranah deteksi objek, keluarga algoritme YOLO (*You Only Look Once*) banyak diadopsi karena bekerja sebagai *single-stage detector* yang ringan dan cepat untuk diterapkan secara *real-time*, dengan berbagai varian YOLOv4, YOLOv5, YOLOv8, dan YOLO12s yang telah diujicobakan untuk deteksi kematangan TBS dan menunjukkan akurasi kompetitif (Genoveva & Syah, 2024; Josdaan et al., 2024; Kurniawan et al., 2023; Lai et al., 2022; Prawidya Nur'aini & Rahardi, 2025; Rasmila et al., 2025; Suharjito, Asrol, et al., 2023; Suharjito, Naftali, et al., 2025). Versi terbaru, YOLO26, dirilis oleh Ultralytics pada Januari 2026 dengan arsitektur *end-to-end* yang menghilangkan *Non-Maximum Suppression* (NMS), sehingga mengurangi kompleksitas inferensi dan meningkatkan kecepatan deteksi pada perangkat *edge* dan *mobile* (Sapkota et al., 2026).

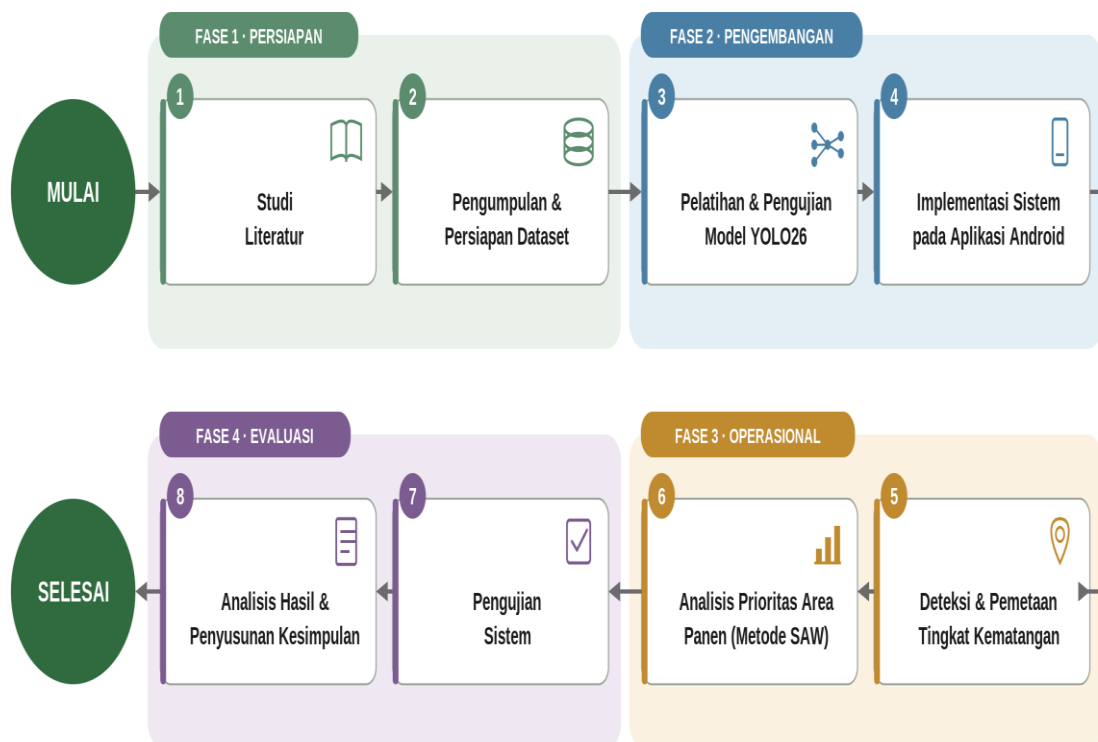
Di sisi lain, teknologi geospasial juga mulai diterapkan pada pengelolaan perkebunan kelapa sawit. Penelitian berbasis Android menunjukkan bahwa pemetaan lahan dapat dilakukan melalui data GPS pada smartphone (Astuti et al., 2022), dan integrasi antara hasil klasifikasi

kematangan dengan analisis geospasial telah diujicobakan pada platform berbasis server untuk verifikasi lokasi perkebunan (Puttinaovarat et al., 2024).

Penelitian-penelitian tersebut menunjukkan bahwa deteksi kematangan berbasis YOLO dan integrasi geospasial telah masing-masing terbukti efektif secara terpisah, namun keduanya umumnya dikembangkan sebagai dua lini penelitian yang berdiri sendiri. Belum ditemukan penelitian yang menggabungkan deteksi kematangan TBS berbasis YOLO generasi terbaru, pencatatan lokasi GPS otomatis, dan metode pendukung keputusan multikriteria untuk memprioritaskan area panen dalam satu sistem *mobile* yang dapat berjalan tanpa koneksi internet, sebagaimana diusulkan pada penelitian ini.

### METODE PENELITIAN

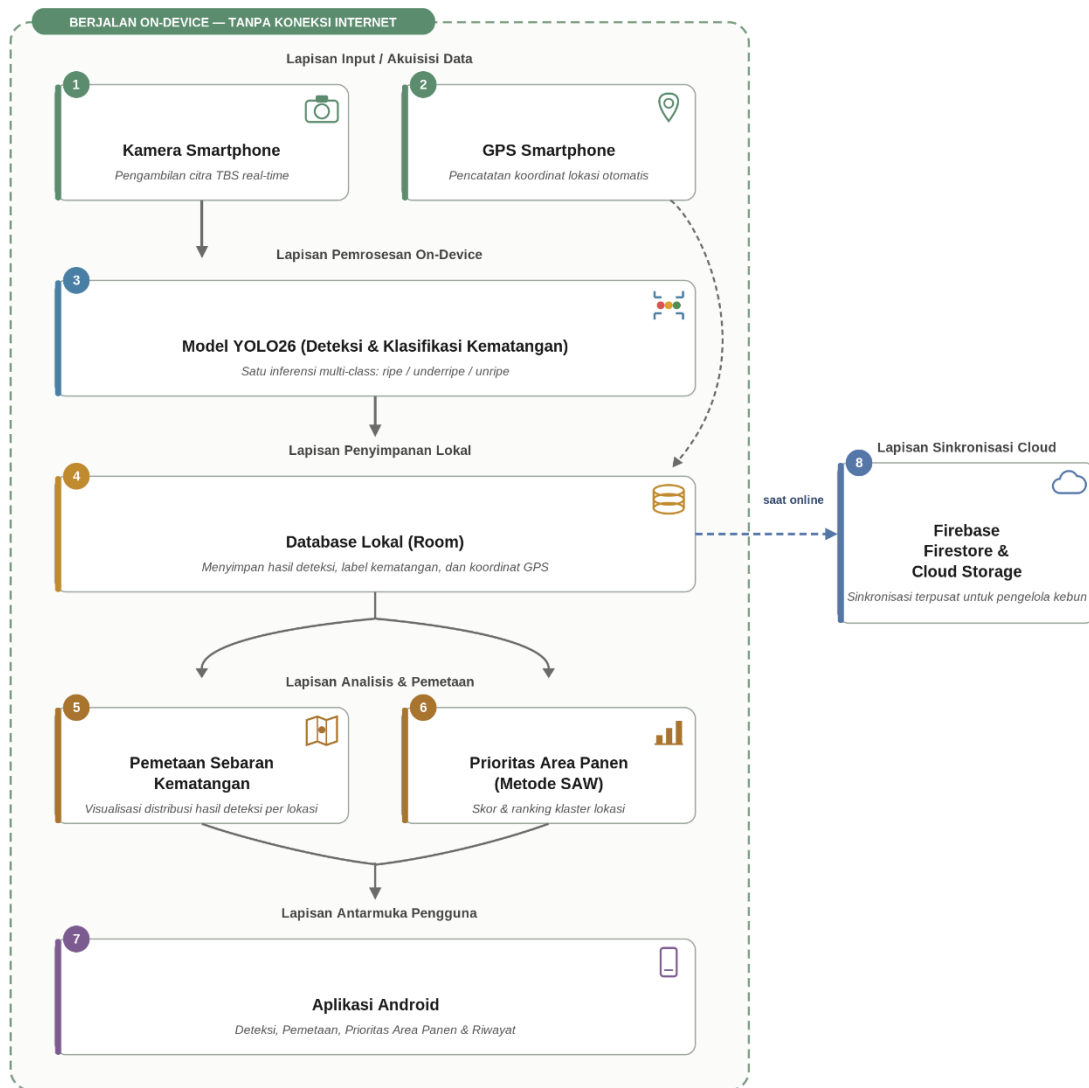
Penelitian ini menggunakan kerangka penelitian yang terdiri atas delapan tahapan: studi literatur, pengumpulan dan persiapan *dataset*, pelatihan dan pengujian model YOLO26, implementasi sistem pada aplikasi Android, deteksi dan pemetaan tingkat kematangan, analisis prioritas area panen menggunakan SAW, pengujian sistem, serta analisis hasil dan penyusunan kesimpulan, seperti ditunjukkan pada Gambar 1.



Gambar 1. Kerangka Penelitian

Gambar 1 menunjukkan kedelapan tahapan tersebut yang dikelompokkan ke dalam empat fase berkode warna, yaitu fase persiapan (studi literatur serta pengumpulan dan persiapan *dataset*), fase pengembangan model (pelatihan dan pengujian YOLO26), fase implementasi (integrasi pada aplikasi Android, deteksi dan pemetaan kematangan, serta analisis prioritas area panen menggunakan SAW), dan fase evaluasi (pengujian sistem serta analisis hasil dan penyusunan kesimpulan). Pengelompokan ini menggambarkan alur penelitian secara berurutan, dari tahap eksploratif hingga evaluatif.

Arsitektur sistem SawitScan dirancang dalam lima lapisan fungsional yang seluruhnya berjalan secara *on-device* tanpa memerlukan koneksi internet, ditambah satu lapisan sinkronisasi cloud opsional, seperti ditunjukkan pada Gambar 2.



Gambar 2. Diagram Blok Sistem SawitScan

Lapisan input/akuisisi data terdiri atas (1) kamera *smartphone* Android untuk mengambil citra TBS dan (2) GPS *smartphone* untuk mencatat koordinat lokasi secara otomatis. Lapisan pemrosesan *on-device*, yaitu (3) model YOLO26, mendeteksi objek dan mengklasifikasikan kematangan (*ripe*, *underripe*, *unripe*) sekaligus dalam satu kali inferensi. Lapisan penyimpanan lokal, yaitu (4) basis data lokal (Room Database), menyimpan seluruh hasil deteksi beserta koordinat GPS-nya agar aplikasi tetap berfungsi tanpa koneksi internet. Lapisan analisis dan pemetaan terdiri atas (5) modul pemetaan yang bersifat deskriptif untuk memvisualisasikan sebaran hasil deteksi pada peta, dan (6) metode SAW yang bersifat preskriptif untuk menghitung skor dan peringkat prioritas area panen per kluster lokasi; kedua modul ditampilkan sebagai mode tampilan terpisah pada halaman peta meski mengonsumsi data kluster yang sama. Lapisan antarmuka pengguna, yaitu (7) aplikasi Android, menyajikan seluruh keluaran tersebut dalam satu antarmuka *mobile* yang terintegrasi. Terakhir, lapisan sinkronisasi cloud opsional, yaitu (8) backend Firebase (Firestore dan Cloud Storage), menyinkronkan data secara otomatis ke server pusat saat koneksi internet tersedia.

### Pengumpulan Dan Persiapan *Dataset*

*Dataset* penelitian terdiri atas citra TBS kelapa sawit yang dikumpulkan dari area perkebunan di Desa Muara Padang, Banyuasin, Sumatera Selatan, sebanyak 254 citra primer pada tahap awal. *Dataset* ini selanjutnya diperluas menggunakan citra dari *dataset* publik TBS SAWIT yang tersedia di Roboflow Universe (*Dataset* TBS Sawit, 2025), sehingga total mencapai

1.124 citra dengan tiga kelas kematangan: *ripe* (matang), *underripe* (kurang matang), dan *unripe* (belum matang).

Dari 1.124 citra tersebut diperoleh 2.384 kotak anotasi (*bounding box*) yang tersebar pada tiga kelas kematangan, dengan rincian distribusi sebagaimana disajikan pada Tabel 1. Kelas *unripe* paling dominan karena tandan yang belum matang umumnya berjumlah lebih banyak dan lebih sering muncul berkelompok dalam satu citra dibandingkan tandan yang sudah matang.

Tabel 1. Distribusi Kelas Kematangan pada *Dataset*

| Kelas                            | Jumlah Kotak Anotasi<br>( <i>Bounding Box</i> ) | Jumlah Citra |
|----------------------------------|-------------------------------------------------|--------------|
| <i>Ripe</i> (matang)             | 277                                             | 261          |
| <i>Underripe</i> (kurang matang) | 486                                             | 406          |
| <i>Unripe</i> (belum matang)     | 1.621                                           | 767          |
| Total                            | 2.384                                           | 1.124        |

Jumlah citra pada Tabel 1 menunjukkan citra yang memuat minimal satu kotak anotasi dari kelas tersebut; satu citra dapat memuat lebih dari satu kelas kematangan sekaligus, sehingga penjumlahan jumlah citra per kelas tidak selalu sama dengan total 1.124 citra pada *dataset*. *Dataset* gabungan ini selanjutnya dibagi menjadi subset latih (*train*) 784 citra, validasi (*valid*) 187 citra, dan uji (*test*) 153 citra, dengan subset *test* tidak pernah dilibatkan dalam proses pelatihan maupun pemilihan model terbaik.

Tabel 2. Pembagian *Dataset* ke dalam Subset Latih, Validasi, dan Uji

| Subset                    | Jumlah Citra | Persentase |
|---------------------------|--------------|------------|
| Latih ( <i>train</i> )    | 784          | 69,8%      |
| Validasi ( <i>valid</i> ) | 187          | 16,6%      |
| Uji ( <i>test</i> )       | 153          | 13,6%      |
| Total                     | 1.124        | 100%       |

### Anotasi dan Praproses Data

Anotasi dilakukan secara manual menggunakan platform Roboflow dengan menggambar kotak pembatas (*bounding box*) pada setiap tandan buah segar yang terlihat dalam citra, kemudian memberi label salah satu dari tiga kelas kematangan (*ripe*, *underripe*, atau *unripe*) sesuai kondisi visual buah pada tandan tersebut. Hasil anotasi diekspor dalam format YOLO, yaitu setiap baris anotasi berisi indeks kelas beserta koordinat titik tengah, lebar, dan tinggi kotak yang telah dinormalisasi terhadap dimensi citra (rentang 0–1). Citra yang bersumber dari *dataset* publik TBS SAWIT (*Dataset* TBS Sawit, 2025) telah memiliki anotasi bawaan, yang kemudian diverifikasi ulang dan disesuaikan ke dalam skema tiga kelas yang sama sebelum digabungkan dengan citra primer, untuk menjaga konsistensi definisi kelas di seluruh *dataset*. Seluruh citra dalam *dataset* akhir (1.124 citra) selanjutnya melalui tahap praproses yang sama pada platform Roboflow, yaitu auto-orientation untuk mengoreksi orientasi piksel berdasarkan metadata EXIF, dan resize ke resolusi 640×640 piksel (fit within) agar sesuai dengan dimensi input model YOLO.

### Pelatihan Model YOLO26n

Setelah tahap persiapan data selesai, proses pelatihan dimulai menggunakan varian YOLO26n. Varian nano (n) dipilih karena memiliki arsitektur yang ringan dan beban komputasi yang rendah dibandingkan varian YOLO26 lainnya (s dan m), sehingga sesuai untuk implementasi pada perangkat *mobile*. Selain itu, YOLO26n memungkinkan klasifikasi dan lokalisasi objek dilakukan secara simultan dalam satu kali proses inferensi, menghasilkan kecepatan dan efisiensi deteksi yang tinggi, didukung pula oleh arsitektur *end-to-end* tanpa *Non-Maximum Suppression* (NMS) yang mengurangi kompleksitas pasca-proses (Sapkota et al., 2026).

*Framework* Ultralytics YOLO digunakan sebagai lingkungan utama untuk melatih model menggunakan *dataset* yang telah disiapkan. Selama proses optimasi berlangsung secara iteratif, jaringan memperbarui bobot internalnya untuk menangkap pola visual khas dari masing-masing kelas kematangan. Untuk memastikan proses pelatihan menuju konvergensi optimal,

beberapa indikator *loss* dipantau, yaitu *box loss*, *classification loss*, dan *distribution focal loss* (DFL). Rincian *hyperparameter* dan konfigurasi yang digunakan pada penelitian ini disajikan pada Tabel 3.

Table 3. Parameter Pelatihan Model YOLO26n

| Parameter                          | Nilai                                               |
|------------------------------------|-----------------------------------------------------|
| Model                              | YOLO26n                                             |
| <i>Epoch</i> (maksimum)            | 100 ( <i>early stopping</i> , <i>patience</i> = 20) |
| Ukuran citra ( <i>image size</i> ) | 640×640                                             |
| <i>Batch size</i>                  | 4                                                   |
| <i>Optimizer</i>                   | auto (Ultralytics)                                  |
| Jumlah worker ( <i>workers</i> )   | 4                                                   |

Tabel 3 menjelaskan parameter pelatihan yang digunakan, dimana *epoch* maksimum ditetapkan sebesar 100 dengan mekanisme *early stopping* (*patience* = 20) untuk memberikan iterasi yang cukup bagi model dalam menangkap fitur visual kematangan TBS, sekaligus menjaga keseimbangan antara konvergensi model dan efisiensi waktu pelatihan sehingga risiko *underfitting* maupun *overfitting* dapat diminimalkan. Citra input distandardisasi pada resolusi 640×640 piksel, ukuran yang umum digunakan pada arsitektur YOLO karena menjaga detail objek penting sekaligus mempertahankan performa komputasi. Ukuran *batch* sebesar 4 diterapkan untuk menstabilkan gradien pelatihan sesuai dengan kapasitas sumber daya komputasi yang tersedia. Proses pelatihan berhenti pada *epoch* ke-87 melalui mekanisme *early stopping*, menandakan bahwa performa model pada subset *valid* telah konvergen sebelum mencapai batas *epoch* maksimum.

### Penerapan Metode SAW untuk Prioritas Area Panen

Metode Simple Additive Weighting (SAW) dipilih karena telah banyak diterapkan secara luas pada berbagai domain pengambilan keputusan multikriteria berbasis pertanian, misalnya klasifikasi kualitas beras (Cahyana et al., 2025) dan penilaian kesesuaian lahan pertanian (Nabiollahi et al., 2024; Cao & Solangi, 2023), sehingga dipandang sesuai untuk diadaptasi pada penentuan prioritas area panen kelapa sawit. Hasil deteksi pada aplikasi dikelompokkan menjadi klaster lokasi (*area*) berdasarkan kedekatan koordinat GPS antar titik deteksi. Setiap klaster lokasi diperlakukan sebagai alternatif dalam proses pengambilan keputusan SAW, dengan tiga kriteria penilaian: C1 = jumlah deteksi *ripe*, C2 = jumlah deteksi *underripe*, dan C3 = jumlah deteksi *unripe* pada klaster tersebut. Ketiga kriteria bertipe benefit, sehingga nilainya dinormalisasi dan diperingkat melalui dua tahap berikut.

#### 1. Normalisasi Matriks Keputusan

Setiap nilai kriteria dinormalisasi terhadap nilai maksimumnya pada seluruh klaster yang dibandingkan, dengan rumus sebagai berikut (1):

$$r_{ij} = \frac{x_{ij}}{\max(x_{ij})}$$

#### Keterangan:

$r_{ij}$  = nilai normalisasi kriteria ke-*j* pada klaster ke-*i*

$x_{ij}$  = nilai kriteria ke-*j* pada klaster ke-*i* (C1, C2, atau C3)

$\max(x_{ij})$  = nilai maksimum kriteria ke-*j* di antara seluruh klaster yang dibandingkan

Pendekatan benefit dipilih karena jumlah deteksi pada kelas apa pun tetap menjadi indikator volume dan tahap kematangan buah yang informatif bagi perencanaan panen; perbedaan urgensi panen antar kelas direpresentasikan melalui bobot kepentingan, bukan arah normalisasi. Bobot yang digunakan adalah  $w_1 = 0,6$  untuk *ripe*,  $w_2 = 0,3$  untuk *underripe*, dan  $w_3 = 0,1$  untuk *unripe*, mencerminkan urgensi panen tertinggi pada buah yang sudah matang dan urgensi yang menurun pada buah yang baru matang pada ronde panen berikutnya.

## 2. Perankingan Alternatif

Nilai preferensi setiap kluster dihitung sebagai penjumlahan dari hasil kali bobot dan nilai normalisasi setiap kriteria, dengan rumus sebagai berikut (2):

$$V_i = \sum_{j=1}^n w_j \times r_{ij}$$

**Keterangan:**

- $V_i$  = nilai preferensi (skor akhir) kluster ke- $i$   
 $w_j$  = bobot kepentingan kriteria ke- $j$  ( $w_1 = 0,6$ ;  $w_2 = 0,3$ ;  $w_3 = 0,1$ )  
 $r_{ij}$  = nilai normalisasi kriteria ke- $j$  pada kluster ke- $i$   
 $n$  = jumlah kriteria yang digunakan ( $n = 3$ , yaitu C1, C2, C3)

Kluster diperingkat dari nilai preferensi ( $V_i$ ) tertinggi ke terendah untuk menentukan prioritas area panen, yang pada aplikasi ditampilkan dalam kategori performa Tertinggi, Sedang, dan Terendah.

## Evaluasi Sistem

Evaluasi akurasi dilakukan secara independen terhadap model akhir yang benar-benar ter-embed pada aplikasi (format TensorFlow Lite, float32), bukan terhadap bobot PyTorch mentah. *Pipeline* pra-proses dan pasca-proses pada skrip evaluasi ditulis ulang untuk meniru persis logika *decoding* pada aplikasi (*letterbox* ke 640×640, normalisasi RGB/255, output tensor [1, 300, 6] dari model *end-to-end* tanpa NMS terpisah, ambang *confidence*  $\geq 0,20$ , dan filter kotak deteksi yang berukuran lebih dari 95% bidang citra). Pencocokan antara prediksi dan *ground truth* menggunakan  $IoU \geq 0,5$  secara *greedy* berdasarkan *confidence* tertinggi.

Performa model diukur menggunakan empat metrik standar deteksi objek, yaitu *Precision*, *Recall*, F1-score, dan *mean Average Precision* (mAP). *Precision* menyatakan proporsi deteksi yang benar (*true positive*) terhadap seluruh deteksi yang dihasilkan model, dirumuskan sebagai berikut (3):

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

**Keterangan:**

- $TP$  = jumlah *true positive* (deteksi yang sesuai dengan *ground truth*)  
 $FP$  = jumlah *false positive* (deteksi yang tidak sesuai dengan *ground truth*)

*Recall* menyatakan proporsi objek *ground truth* yang berhasil terdeteksi oleh model, dirumuskan sebagai berikut (4):

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

**Keterangan:**

- $FN$  = jumlah *false negative* (objek *ground truth* yang gagal terdeteksi model)

F1-score merupakan rata-rata harmonik antara *precision* dan *recall* yang menyeimbangkan kedua metrik tersebut ke dalam satu nilai tunggal, dirumuskan sebagai berikut (5):

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

mAP dihitung sebagai rata-rata *Average Precision* (AP) dari seluruh kelas yang dievaluasi, dirumuskan sebagai berikut (5):

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^N AP_i$$

#### Keterangan:

**N** = jumlah kelas yang dievaluasi ( $N = 3$ , yaitu *ripe*, *underripe*, dan *unripe*)  
**API** = *Average Precision* kelas ke-*i*, diperoleh melalui interpolasi 101 titik bergaya COCO dari kurva *precision-recall* penuh

Penelitian ini melaporkan dua varian mAP, yaitu mAP@0.5 yang dihitung pada ambang IoU tetap 0,50, dan mAP@[0.5:0.95] yang dihitung sebagai rata-rata pada rentang ambang IoU 0,50 hingga 0,95 dengan kelipatan 0,05, sehingga memberikan penilaian yang lebih ketat terhadap akurasi lokalisasi spasial model. Evaluasi dilakukan secara terpisah pada subset *valid* (digunakan selama pemilihan model ketika pelatihan, sehingga kurang murni sebagai estimasi generalisasi) dan subset *test* (tidak pernah disentuh selama pelatihan, sehingga dijadikan angka acuan utama performa sistem).

### HASIL DAN PEMBAHASAN

#### Performa Pelatihan Model YOLO26n

Performa pelatihan model YOLO26n dipantau melalui metrik *precision*, *recall*, *mean Average Precision* pada IoU 0,5 (mAP@0.5), dan mAP pada rentang IoU 0,5 hingga 0,95 (mAP@[0.5:0.95]) yang dihitung Ultralytics pada subset *valid* di setiap akhir *epoch*. Tabel 4 menunjukkan nilai keempat metrik tersebut pada *epoch* ke-87, yaitu titik proses pelatihan berhenti melalui mekanisme *early stopping*.

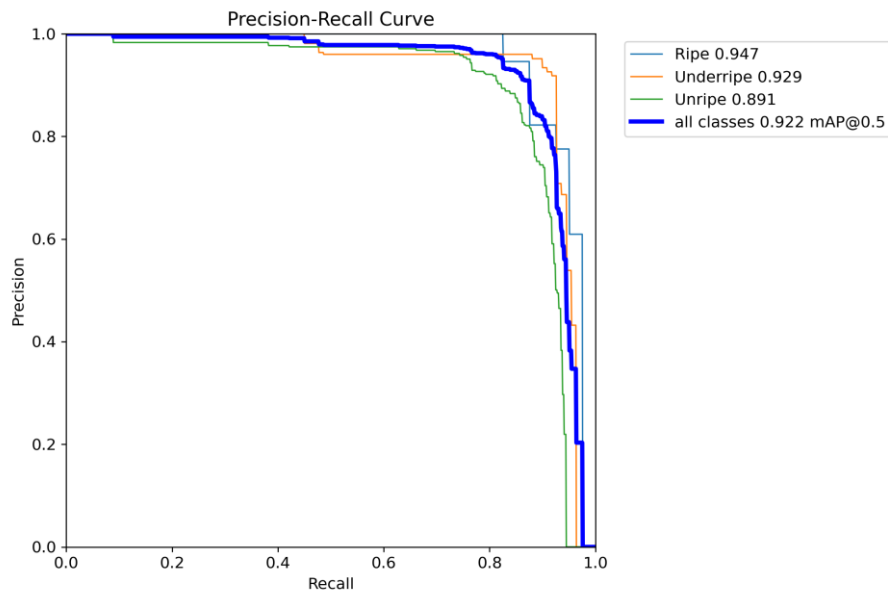
Tabel 4. Metrik Akhir Pelatihan Model YOLO26n pada Subset *Valid* (*Epoch* ke-87)

| Metrik           | Nilai |
|------------------|-------|
| <i>Precision</i> | 0,898 |
| <i>Recall</i>    | 0,873 |
| F1-score         | 0,885 |
| mAP@0.5          | 0,921 |
| mAP@[0.5:0.95]   | 0,626 |

Tabel 4 menunjukkan bahwa model YOLO26n mencapai *precision* sebesar 0,898, yang berarti sekitar 89,8% dari seluruh prediksi yang dihasilkan model merupakan deteksi yang benar (*true positive*), sedangkan sisanya merupakan kesalahan deteksi (*false positive*). *Recall* sebesar 0,873 menunjukkan bahwa model berhasil menemukan 87,3% dari seluruh objek TBS yang sebenarnya ada pada citra valid, dengan F1-score sebesar 0,885 yang mengonfirmasi keseimbangan yang baik antara *precision* dan *recall* tersebut. Nilai mAP@0.5 sebesar 0,921 mengindikasikan akurasi deteksi dan klasifikasi yang tinggi pada ambang IoU 0,5, sedangkan mAP@[0.5:0.95] sebesar 0,626 yang dihitung pada rentang ambang IoU yang lebih ketat menunjukkan model tetap mampu menghasilkan kotak deteksi yang cukup presisi, meskipun nilainya secara alami lebih rendah dibandingkan mAP@0.5 karena mensyaratkan kesesuaian lokasi kotak yang lebih akurat.

#### Analisis Kurva *Precision-Recall*

Selain metrik evaluasi agregat pada Tabel 4, performa model juga perlu ditelaah lebih rinci pada masing-masing kelas kematangan, mengingat *ripe*, *underripe*, dan *unripe* memiliki karakteristik visual yang berbeda dan berpotensi memiliki tingkat kesulitan deteksi yang tidak seragam. Salah satu cara untuk menelaah hal tersebut adalah melalui kurva *precision-recall* per kelas, yang menggambarkan *trade-off* antara *precision* dan *recall* pada berbagai ambang *confidence*. Setelah proses pelatihan selesai, performa model dievaluasi lebih lanjut melalui kurva tersebut, sebagaimana ditunjukkan pada Gambar 3.

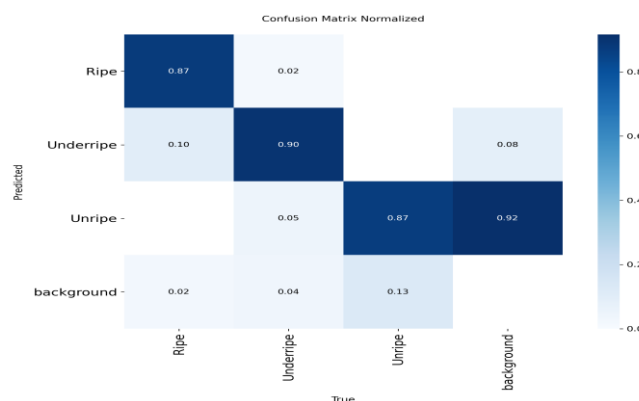


Gambar 3. Kurva *Precision-Recall* Model YOLO26n pada Subset *Valid*

Gambar 3 menyajikan kurva *precision-recall* yang dihasilkan Ultralytics pada akhir proses pelatihan, menunjukkan area *precision-recall* (AP) untuk masing-masing kelas pada ambang IoU 0,5. Kelas *ripe* memperoleh AP tertinggi sebesar 0,947, diikuti *underripe* sebesar 0,929, dan *unripe* sebesar 0,891, dengan mAP@0.5 keseluruhan sebesar 0,922. Ketiga kurva kelas berada dekat sudut kanan atas grafik sepanjang rentang *recall* 0–0,8, menunjukkan model mampu mempertahankan *precision* yang tinggi tanpa banyak mengorbankan *recall* hingga ambang tersebut. Penurunan *precision* yang lebih tajam baru terjadi pada *recall* di atas 0,8, terutama pada kelas *unripe*, yang mengindikasikan sebagian kecil objek *unripe* yang lebih sulit dikenali (misalnya karena kemiripan warna dengan *underripe*) hanya terdeteksi pada ambang *confidence* yang lebih rendah.

### Analisis Matriks Konfusi

Kurva *precision-recall* pada Gambar 3 memberikan gambaran performa deteksi pada berbagai ambang *confidence*, namun belum menunjukkan secara spesifik kelas mana yang paling sering tertukar atau gagal terdeteksi pada satu titik operasi tertentu. Untuk melengkapi gambaran tersebut, performa klasifikasi model pada masing-masing kelas kematangan juga dianalisis melalui matriks konfusi ternormalisasi, yang merepresentasikan proporsi prediksi benar maupun salah antar kelas dalam bentuk persentase, sebagaimana ditunjukkan pada Gambar 4.



Gambar 4. Matriks Konfusi Ternormalisasi Model YOLO26n pada Subset Valid

Gambar 4 menunjukkan matriks konfusi ternormalisasi pada subset *valid*. Nilai diagonal menunjukkan proporsi prediksi yang benar untuk masing-masing kelas, yaitu 0,87 untuk *ripe*, 0,90 untuk *underripe*, dan 0,87 untuk *unripe*. Kesalahan klasifikasi antar kelas relatif kecil, dengan kekeliruan terbesar terjadi antara *underripe* yang sebenarnya diprediksi sebagai *ripe*

(0,10) serta antara *unripe* yang sebenarnya diprediksi sebagai *underripe* (0,05), sejalan dengan kemiripan visual antar tahap kematangan yang berdekatan. Baris *background* pada matriks merepresentasikan objek yang gagal terdeteksi (*missed detection*), dengan proporsi tertinggi pada kelas *unripe* (0,13), kemungkinan disebabkan oleh warna dan kontur *unripe* yang lebih menyatu dengan dedaunan dan latar kebun dibandingkan *ripe* maupun *underripe*.

### Hasil Deteksi pada Citra Validasi

Di samping evaluasi kuantitatif melalui *precision*, *recall*, dan matriks konfusi, performa model juga perlu ditinjau secara kualitatif untuk melihat bagaimana model bekerja pada kondisi citra lapangan yang sesungguhnya, termasuk pada kondisi tandan yang saling berdekatan atau sebagian tertutup oleh daun. Untuk itu, beberapa contoh hasil deteksi pada citra subset *valid* divisualisasikan pada Gambar 5.

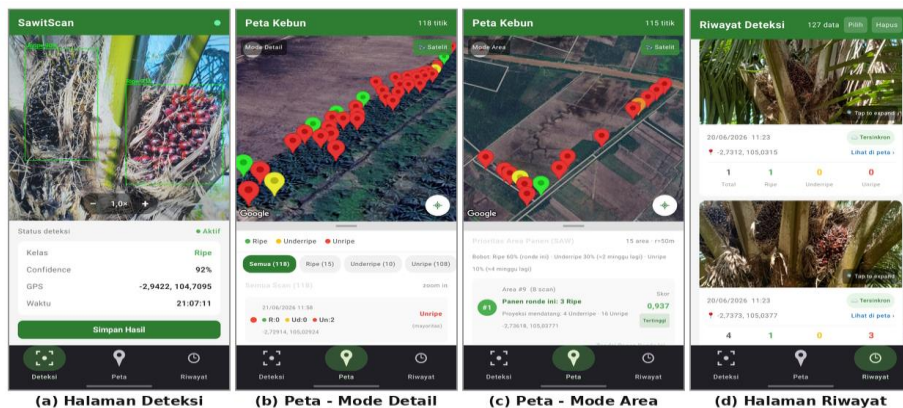


Gambar 5. Contoh Hasil Deteksi Model YOLO26n pada Citra Validasi

Gambar 5 memperlihatkan contoh hasil deteksi model YOLO26n pada beberapa citra *valid*, lengkap dengan kotak deteksi, label kelas, dan skor *confidence* untuk setiap TBS yang terdeteksi. Model berhasil mendeteksi beberapa TBS dengan tingkat kematangan yang berbeda-beda dalam satu citra, termasuk pada kondisi tandan yang saling berdekatan atau sebagian tertutup oleh daun, dengan skor *confidence* yang umumnya berada pada rentang 0,6–0,9. Hasil ini menunjukkan model mampu melakukan deteksi multi-objek secara simultan pada kondisi citra lapangan yang realistis, sebelum model diekspor ke format TensorFlow Lite untuk diintegrasikan ke dalam aplikasi Android, sebagaimana dibahas pada subbagian berikutnya.

### Implementasi Antarmuka Aplikasi Android

Model YOLO26n yang telah diekspor ke format TensorFlow Lite diintegrasikan ke dalam aplikasi Android bernama SawitScan, yang terdiri atas tiga halaman utama yang dapat diakses melalui navigasi bawah, yaitu Deteksi, Peta, dan Riwayat, sebagaimana ditunjukkan pada Gambar 6.



Gambar 6. Antarmuka Aplikasi Android SawitScan: (a) Halaman Deteksi, (b) Peta Mode Detail, (c) Peta Mode Area, (d) Halaman Riwayat

Halaman Deteksi (Gambar 6a) menjalankan inferensi model secara *real-time* pada umpan kamera, menggambar kotak deteksi beserta label kelas dan skor *confidence* langsung di atas pratinjau kamera, serta mencatat koordinat GPS pada saat pengguna menekan tombol simpan hasil.

Setiap hasil deteksi yang tersimpan divisualisasikan pada halaman Peta dalam dua mode tampilan. Mode Detail (Gambar 6b) menampilkan setiap titik deteksi sebagai *marker* individual berwarna sesuai kelas kematangan mayoritasnya (hijau untuk *ripe*, kuning untuk *underripe*, merah untuk *unripe*), dilengkapi *sub-tab* penyaring jumlah titik per kelas.

Mode Area (Gambar 6c) mengelompokkan titik-titik deteksi yang berdekatan menjadi kluster lokasi berdasarkan kedekatan koordinat GPS, kemudian metode Simple Additive Weighting (SAW) menghitung skor preferensi untuk setiap kluster berdasarkan rekapitulasi jumlah deteksi *ripe*, *underripe*, dan *unripe* pada kluster tersebut, sesuai dengan bobot kepentingan yang ditetapkan (0,6/0,3/0,1). Pengujian fungsional menunjukkan bahwa kluster dengan proporsi deteksi *ripe* yang tinggi secara konsisten memperoleh skor preferensi tertinggi dan dikategorikan sebagai performa Tertinggi, sedangkan kluster yang didominasi deteksi *unripe* memperoleh skor terendah dan dikategorikan Terendah. Hasil pemeringkatan ini ditampilkan langsung pada Gambar 6c, lengkap dengan skor preferensi dan proyeksi jumlah TBS pada ronde mendatang, sebagai dasar bagi pengelola kebun dan tim pemanen untuk menentukan area mana yang perlu dikunjungi lebih dahulu, menggantikan pengecekan manual berkeliling kebun yang selama ini dilakukan.

Adapun halaman Riwayat (Gambar 6d) menampilkan seluruh hasil deteksi yang tersimpan secara lokal dalam bentuk daftar kartu, masing-masing memuat foto hasil deteksi, koordinat GPS, rincian jumlah deteksi per kelas, serta status sinkronisasi data ke server.

### Evaluasi Akurasi Model Akhir pada Aplikasi

Akurasi model YOLO26n yang telah diekspor ke format TensorFlow Lite (float32) dan ter-embed pada aplikasi dievaluasi secara independen pada subset test (153 citra, 308 kotak ground truth) menggunakan pipeline pra-proses dan pasca-proses yang identik dengan logika decoding aplikasi, pada titik operasi confidence  $\geq 0,20$ . Tabel 5 menunjukkan nilai *precision*, *recall*, F1-score, mAP@0.5, dan mAP@[0.5:0.95] hasil evaluasi tersebut.

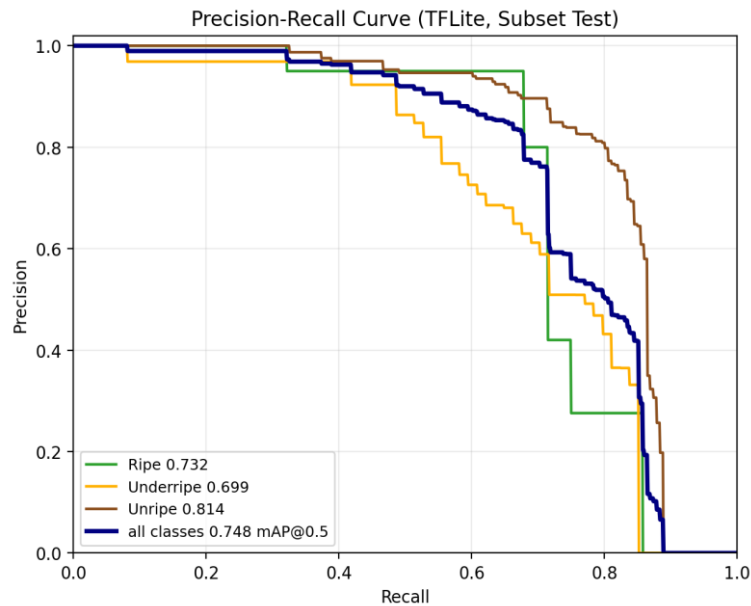
Tabel 5. Metrik Evaluasi Akurasi Model TFLite pada Subset *Test* (Titik Operasi Aplikasi)

| Metrik           | Nilai |
|------------------|-------|
| <i>Precision</i> | 0,745 |
| <i>Recall</i>    | 0,740 |
| F1-score         | 0,743 |
| mAP@0.5          | 0,748 |
| mAP@[0.5:0.95]   | 0,472 |

Tabel 5 menunjukkan bahwa pada subset *test* yang sama sekali tidak dilibatkan selama pelatihan maupun pemilihan model, *precision* sebesar 0,745 berarti sekitar 74,5% dari seluruh prediksi aplikasi merupakan deteksi yang benar, sedangkan *recall* sebesar 0,740 menunjukkan aplikasi berhasil menemukan 74,0% dari seluruh TBS yang sebenarnya ada pada citra *test*, dengan F1-score keseluruhan sebesar 0,743. Nilai mAP@0.5 sebesar 0,748 dan mAP@[0.5:0.95] sebesar 0,472 ini sedikit lebih rendah dibandingkan metrik pelatihan pada Tabel 4 (F1-score = 0,885; mAP@0.5 = 0,921; mAP@[0.5:0.95] = 0,626), karena Tabel 5 diukur dari berkas .tflite hasil ekspor melalui *pipeline decoding* asli aplikasi pada subset *test* yang benar-benar terpisah dari proses pelatihan, sedangkan Tabel 4 diukur dari bobot PyTorch mentah pada subset *valid*.

### Analisis Kurva Precision-Recall pada Aplikasi

Sebagaimana pada evaluasi model PyTorch di subbagian sebelumnya, metrik agregat pada Tabel 5 belum menggambarkan performa model TFLite pada masing-masing kelas kematangan secara terpisah. Untuk melengkapi hasil evaluasi tersebut, performa model TFLite pada subset *test* juga dianalisis melalui kurva *precision-recall* per kelas, sebagaimana ditunjukkan pada Gambar 7.

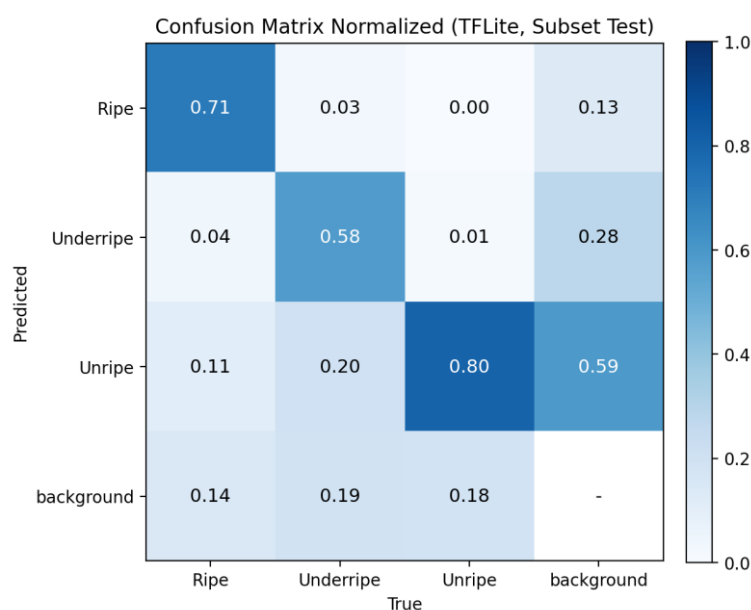


Gambar 7. Kurva *Precision-Recall* Model TFLite pada Subset *Test*

Gambar 7 menyajikan kurva *precision-recall* dari evaluasi independen model .tflite pada subset *test*, dihitung pada ambang IoU 0,5 dengan *confidence* disapu dari mendekati 0 hingga 1. Kelas *unripe* mencapai AP tertinggi sebesar 0,814, diikuti *ripe* sebesar 0,732 dan *underripe* sebesar 0,699, dengan mAP@0.5 keseluruhan sebesar 0,748 sebagaimana dilaporkan pada Tabel 5. Dibandingkan dengan kurva PR pada Gambar 3 yang diukur dari bobot PyTorch mentah pada subset *valid*, kurva pada Gambar 7 menurun lebih cepat pada *recall* menengah, mengindikasikan *pipeline* ekspor TFLite dan subset *test* yang lebih sulit bersama-sama menurunkan presisi pada ambang *recall* yang sama.

### Analisis Matriks Konfusi pada Aplikasi

Selain kurva *precision-recall* pada Gambar 7, pola kesalahan klasifikasi model TFLite pada subset *test* juga perlu ditelaah untuk mengetahui kelas mana yang paling sering keliru diklasifikasikan setelah model dikonversi dan dijalankan melalui *pipeline decoding* aplikasi yang sesungguhnya. Untuk itu, kesalahan klasifikasi tersebut dianalisis melalui matriks konfusi ternormalisasi, sebagaimana ditunjukkan pada Gambar 8.



Gambar 8. Matriks Konfusi Ternormalisasi Model TFLite pada Subset *Test*

Gambar 8 menunjukkan matriks konfusi ternormalisasi dari hasil evaluasi independen model tflite pada subset *test*, dinormalisasi per kolom aktual. Nilai diagonal mengonfirmasi *recall* per kelas yang telah dilaporkan sebelumnya: 0,71 untuk *ripe*, 0,58 untuk *underripe*, dan 0,80 untuk *unripe*. Kolom *background* (kanan) merepresentasikan proporsi prediksi yang sebenarnya merupakan *false positive* (tidak ada objek nyata), sedangkan baris *background* (bawah) merepresentasikan proporsi objek nyata yang gagal terdeteksi (*false negative*) pada masing-masing kelas, dengan proporsi tertinggi pada kelas *ripe* (0,14). Dibandingkan dengan Gambar 4 yang dihasilkan dari bobot PyTorch mentah pada subset *valid*, matriks ini menunjukkan tingkat kesalahan yang lebih tinggi, terutama kesalahan *underripe-ke-background* (0,28) dan *unripe-ke-background* (0,59 dari total kasus *background* pada kolom tersebut), sejalan dengan kesenjangan generalisasi pada subset *test* yang telah dibahas sebelumnya.

## KESIMPULAN

Penelitian ini berhasil merancang dan mengevaluasi SawitScan, sebuah aplikasi Android yang mengintegrasikan deteksi kematangan tandan buah segar kelapa sawit berbasis YOLO26, pencatatan dan pemetaan koordinat GPS, serta metode Simple Additive Weighting (SAW) untuk menentukan prioritas area panen, seluruhnya berjalan secara *on-device* tanpa memerlukan koneksi internet. Model YOLO26n yang dilatih pada *dataset* gabungan 1.124 citra terbukti sesuai untuk implementasi pada perangkat *mobile*, dengan ukuran yang kecil (5,36 MB) dan akurasi yang kompetitif (mAP@0.5 = 0,921 pada data validasi pelatihan), didukung oleh keunggulan arsitektur *end-to-end* tanpa NMS. Evaluasi independen terhadap model akhir yang ter-embed pada aplikasi (format TFLite), dengan *pipeline* pra-proses dan pasca-proses yang identik dengan aplikasi nyata, menunjukkan performa pada data uji yang belum pernah dilihat model sebesar mAP@0.5 = 0,748 dan F1-score keseluruhan = 0,743, dengan kelas *unripe* menunjukkan performa terbaik sedangkan kelas *underripe* menunjukkan kesenjangan generalisasi yang menjadi catatan untuk perbaikan pada penelitian selanjutnya. Metode SAW berhasil diterapkan untuk memeringkat area panen berdasarkan kombinasi jumlah deteksi *ripe*, *underripe*, dan *unripe* per klaster lokasi, sehingga dapat menggantikan pengecekan manual di lapangan dengan rekomendasi yang lebih terstruktur dan berbasis data, sementara penelitian selanjutnya dapat diarahkan pada perluasan *dataset* ke berbagai lokasi dan kondisi perkebunan, validasi hasil pemeringkatan SAW terhadap data realisasi panen aktual di lapangan, serta eksplorasi metode pengambilan keputusan multikriteria lain sebagai pembanding terhadap SAW.

## REFERENSI

- Astuti, I. F., Ahmad, F. N., Cahyadi, D., Rosmasari, Kridalaksana, A. H., & Andrea, R. (2022). Geolocation untuk Lahan Kelapa Sawit Berbasis Android. *Buletin Poltanesa*, 23(1). <https://doi.org/10.51967/tanesa.v23i1.1288>
- BPDP. (2025, September). Peran Strategis Industri Minyak Sawit Nasional. Retrieved June 19, 2026, from <https://www.bpdp.or.id/peran-strategis-industri-minyak-sawit-nasional>
- Cahyana, P. T., Sunarti, T. C., Hardjomidjojo, H., Noor, E., & Laily, N. (2025). Rice Quality Classification Using the Simple Additive Weighting (SAW) Method. *Int. J. Adv. Sci. Eng. Inf. Technol.*, 15(5).
- Cao, J., & Solangi, Y. A. (2023). Analyzing and Prioritizing the Barriers and Solutions of Sustainable Agriculture for Promoting Sustainable Development Goals in China. *Sustainability*, 15(10), 8317.
- Dataset TBS Sawit. (2025, September). TBS SAWIT Dataset [Data set]. *Roboflow Universe*. Retrieved June 19, 2026, from <https://universe.roboflow.com/dataset-tbs-sawit/tbs-sawit-mq8qa>
- Genoveva, Z., & Syah, R. D. (2024). Model Machine Learning untuk Deteksi Tingkat Kematangan Tandan Buah Segar Kelapa Sawit Menggunakan Metode YOLOv8. *Jurnal Pertanian Presisi (Journal of Precision Agriculture)*, 8(2), 121–136. <https://doi.org/10.35760/jpp.2024.v8i2.11848>
- Guslianto, S. I. (2023). Klasifikasi Kematangan Buah Sawit Berdasarkan Fitur Warna, Bentuk dan Tekstur Menggunakan Algoritma K-NN. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 9(3), 407–414.
- Himmah, E. F., Widyaningsih, M., & Maysaroh, M. (2020). Identifikasi Kematangan Buah Kelapa Sawit Berdasarkan Warna RGB dan HSV Menggunakan Metode K-Means Clustering. *Jurnal Sains dan Informatika*, 6(2), 193–202. <https://doi.org/10.34128/jsi.v6i2.242>

- Islah, M., Dhafa, M., & Ikhsan, F. (2025). Klasifikasi Kematangan Tandan Buah Segar Kelapa Sawit Menggunakan ResNet18 Berbasis Citra RGB. *Journal of Informatics and Communication Technology (JICT)*, 7(2), 136–145.
- Ismail, L. I., Wan Hasan, W. Z., Lai, J. W., & Ramli, H. R. (2023, January). Oil Palm Fresh Fruit Bunch Ripeness Detection Methods: A Systematic Review. *Agriculture*, 13(1), 156. <https://doi.org/10.3390/agriculture13010156>
- Josdaan, J., Tamsil, V. C., Harefa, J., & Jingga, K. (2024). Revolutionizing Palm Oil Ripeness Classification: Utilizing YOLOv8 for Ultra-Precise Ripeness Detection. *Procedia Computer Science*, 700–709. <https://doi.org/10.1016/j.procs.2024.10.296>
- Kurniawan, R., Martadinata, A. T., & Cahyo, S. D. (2023). Klasifikasi Tingkat Kematangan Buah Sawit Berbasis Deep Learning dengan Menggunakan Arsitektur YOLOv5. *Journal of Information System Research (JOSH)*, 5(1), 302–309. <https://doi.org/10.47065/josh.v5i1.4408>
- Lai, J., Ramli, H., Ismail, L., & Wan Hasan, W. (2022). Real-Time Detection of Ripe Oil Palm Fresh Fruit Bunch Based on YOLOv4. *IEEE Access*, PP, 1. <https://doi.org/10.1109/ACCESS.2022.3204762>
- Mohamad Zaki, M. A., et al. (2025, March). Impact of Industry 4.0 Technologies on the Oil Palm Industry: A Literature Review. *Smart Agricultural Technology*. <https://doi.org/10.1016/j.atech.2024.100685>
- Nabiollahi, K., et al. (2024). Assessment of Land Suitability Potential Using Ensemble Approaches of Advanced Multi-Criteria Decision Models and Machine Learning for Wheat Cultivation. *Remote Sensing (Basel)*, 16(14), 2566.
- Prawidya Nur'aini, L., & Rahardi, M. (2025). Detection of Ripeness in Oil Palm Fresh Fruit Bunches Using the YOLO12S Algorithm on Digital Images. Retrieved from <http://jurnal.polibatam.ac.id/index.php/JAIC>
- Puttinaovarat, S., Chai-Arayalert, S., & Saetang, W. (2024, May). Oil Palm Bunch Ripeness Classification and Plantation Verification Platform: Leveraging Deep Learning and Geospatial Analysis and Visualization. *ISPRS International Journal of Geo-Information*, 13(5). <https://doi.org/10.3390/ijgi13050158>
- Rasmila, R., Oktavian, S., Dasmen, R., & Amalia, R. (2025). A Comparative Deep Learning Approach for Classifying Oil Palm Fruit Ripeness Levels Using YOLOv8s and Faster R-CNN. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 19(4). <https://doi.org/10.22146/ijccs.111942>
- Sapkota, R., Cheppally, R. H., Sharda, A., & Karkee, M. (2025). YOLO26: key architectural enhancements and performance benchmarking for real-time object detection. *arXiv preprint arXiv:2509.25164*.
- Sari, W. E., Muslimin, M., Franz, A., & Sugiartawan, P. (2022). Deteksi Tingkat Kematangan Tandan Buah Segar Kelapa Sawit dengan Algoritme K-Means. *SINTECH (Science and Information Technology) Journal*, 5(2), 154–164.
- Setiawan, D., Utomo, P. E. P., & Alfalah, M. (2025). Detection of Oil Palm Fruit Ripeness Through Image Feature Optimization Using Convolutional Neural Network Algorithm. *JOIV: International Journal on Informatics Visualization*, 9(2), 674–682.
- Suharjito, Asrol, M., Utama, D. N., Junior, F. A., & Marimin. (2023). Real-Time Oil Palm Fruit Grading System Using Smartphone and Modified YOLOv4. *IEEE Access*, 11, 59758–59773. <https://doi.org/10.1109/ACCESS.2023.3285537>
- Suharjito, Naftali, M. G., Hugo, G., Priyadi, M. R. A., Asrol, M., & Utama, D. N. (2025). Oil Palm Fruits Dataset in Plantations for Harvest Estimation Using Digital Census and Smartphone. *Scientific Data*, 12, 972. <https://doi.org/10.1038/s41597-025-05227-x>
- Suriani, A., & Lisnawita, L. (2024, November). Tinjauan Literatur Sistematis (SLR) Klasifikasi Kematangan Buah Sawit Berdasarkan Citra Warna Menggunakan Algoritma SVM Dan KNN. In *SEMASTER: Seminar Nasional Teknologi Informasi & Ilmu Komputer* (Vol. 3, No. 1, pp. 447–459).
- Zulkarnain, J., Kusriani, & Hidayat, T. (2024, January). Klasifikasi Tingkat Kematangan Tandan Buah Segar Kelapa Sawit Menggunakan Pendekatan Deep Learning. *JST (Jurnal Sains dan Teknologi)*, 12(3). <https://doi.org/10.23887/jstundiksha.v12i3.59140>