



Identifikasi Kematangan Buah Tomat (*Solanum lycopersicum*) Menggunakan YOLOv5n

*Ripeness Identification of Tomato (*Solanum lycopersicum*) Using YOLOv5n*

I Made Arya Yogo Satriawan¹, I Putu Gede Budisanjaya^{1*}, I Made Supartha Utama¹

¹ Program Studi Teknik Pertanian dan Biosistem, Fakultas Teknologi Pertanian Universitas Udayana, Badung, Bali, Indonesia.

ABSTRAK

Identifikasi tingkat kematangan buah tomat secara manual di lahan terbuka menghadapi berbagai kendala, seperti ketidakkonsistenan hasil, kebutuhan tenaga kerja yang besar, dan pemanenan yang tidak tepat waktu. Penelitian ini bertujuan untuk mengembangkan sistem deteksi buah tomat dan klasifikasi tingkat kematangannya menggunakan algoritma YOLOv5n yang diimplementasikan pada perangkat Raspberry Pi 2. Dataset yang digunakan berasal dari LaboroTomato yang terdiri dari 440 citra buah tomat, kemudian melalui proses augmentasi menghasilkan total 1.320 citra yang dibagi menjadi data latih (70%), data validasi (15%), dan data uji (15%). Pelatihan model dilakukan menggunakan Google Colab dengan GPU NVIDIA Tesla T4 selama 100 epoch. Hasil penelitian menunjukkan bahwa model YOLOv5n mencapai precision sebesar 96,28%, recall sebesar 94,22%, F1-Score sebesar 95,24%, serta mAP@0.5 sebesar 97,99%. Konversi model ke format ONNX meningkatkan kecepatan inferensi sebesar 20,5% dengan waktu rata-rata 1,47 detik per frame pada Raspberry Pi 2. Validasi lapangan yang melibatkan empat praktisi menunjukkan tingkat persetujuan sebesar 80,56%, dengan persetujuan tertinggi pada kategori tomat mentah (89,29%) dan matang (85,71%), serta terendah pada kategori setengah matang (66,67%). Sistem ini terbukti dapat diimplementasikan pada perangkat embedded dengan performa yang baik dan dapat diterima untuk aplikasi praktis di lapangan pertanian.

Kata Kunci: computer vision, deep learning, deteksi objek, kematangan tomat, Raspberry Pi, YOLOv5

ABSTRACT

Manual identification of tomato ripeness levels in open fields faces various challenges, including inconsistent results, high labor requirements, and improper harvest timing. This research aims to develop a tomato detection and ripeness classification system using the YOLOv5n algorithm implemented on Raspberry Pi 2. The dataset used is from LaboroTomato consisting of 440 tomato images, which after augmentation produced a total of 1,320 images divided into training data (70%), validation data (15%), and test data (15%). Model training was conducted using Google Colab with NVIDIA Tesla T4 GPU for 100 epochs. The results show that the YOLOv5n model achieved a precision of 96.28%, recall of 94.22%, F1-Score of 95.24%, and mAP@0.5 of 97.99%. Model conversion to ONNX format improved inference speed by 20.5% with an average time of 1.47 seconds per frame on Raspberry Pi 2. Field validation involving four practitioners showed an overall agreement rate of 80.56%, with the highest agreement on unripe tomato category (89.29%) and ripe category (85.71%), and the lowest on semi-ripe category (66.67%). This system is proven to be implementable on embedded devices with good performance and acceptable for practical applications in agricultural fields.

Keywords: computer vision, deep learning, object detection, tomato ripeness, Raspberry Pi, YOLOv5

***Corresponding author: I Putu Gede Budisanjaya**

Program Studi Teknik Pertanian, Fakultas Teknologi Pertanian Universitas Udayana, Badung, Bali, Indonesia. Email: budisanjaya@unud.ac.id

Masuk: 02 Januari 2026;

Direvisi: 03 April 2026;

Diterima: 29 September 2026;

Terbit: 30 September 2026

PENDAHULUAN

Tomat (*Solanum lycopersicum*) merupakan komoditas hortikultura bernilai ekonomi tinggi dengan permintaan pasar yang stabil sepanjang tahun. Di Indonesia, produksi tomat menunjukkan tren peningkatan dan telah mencapai lebih dari 1,17 juta ton per tahun (Badan Pusat Statistik, 2024). Namun demikian, produktivitas rata-rata nasional masih belum optimal, terutama akibat praktik pemanenan yang belum presisi (Kementerian Pertanian Republik Indonesia, 2023). Ketidaktepatan waktu panen berkontribusi terhadap penurunan kualitas buah, meningkatnya susut pascapanen, serta berkurangnya nilai jual produk.

Penentuan tingkat kematangan tomat di tingkat petani umumnya masih dilakukan secara manual berdasarkan pengamatan visual terhadap warna kulit buah. Metode ini sangat dipengaruhi oleh kondisi pencahayaan, latar lingkungan, dan subjektivitas tenaga kerja, sehingga menghasilkan tingkat konsistensi yang rendah. Selain itu, proses identifikasi manual membutuhkan tenaga kerja dalam jumlah besar dan sulit diterapkan secara efisien pada skala produksi yang luas (Li et al., 2023).

Perkembangan computer vision berbasis *deep learning* memberikan peluang untuk mengotomatisasi proses identifikasi kematangan buah secara objektif dan konsisten (Goodfellow et al., 2016) (Lecun et al., 2015). Algoritma deteksi objek You Only Look Once (YOLO) merupakan salah satu pendekatan yang banyak digunakan karena kemampuannya melakukan deteksi dan klasifikasi objek secara simultan dalam satu tahap inferensi (Redmon et al., 2016). Karakteristik tersebut menjadikan YOLO unggul dari sisi kecepatan dan sesuai untuk aplikasi lapangan yang membutuhkan pemrosesan hampir waktu nyata (Girshick, 2015) (Ren et al., 2015). Varian YOLOv5n (nano) dirancang dengan jumlah parameter yang ringan sehingga memungkinkan implementasi pada perangkat dengan keterbatasan sumber daya komputasi. Berbagai penelitian sebelumnya telah melaporkan performa tinggi YOLO dalam mendeteksi dan mengklasifikasikan tingkat kematangan buah tomat (Wang et al., 2023) (Huang et al., 2023). Namun, sebagian besar studi masih berfokus pada evaluasi akurasi model di lingkungan komputasi berkinerja tinggi, tanpa mempertimbangkan keterbatasan perangkat keras yang umum digunakan di lapangan. Kajian mengenai implementasi model *deep learning* ringan pada perangkat *embedded* berdaya rendah, khususnya generasi lama, masih relatif terbatas. Versi terbaru seperti YOLOv5 dirancang dengan model yang lebih ringan namun tetap mempertahankan akurasi tinggi, sehingga cocok untuk diterapkan pada perangkat dengan daya komputasi terbatas, seperti Raspberry Pi (Biglari & Tang, 2023).

Raspberry Pi terus berkembang dengan mengeluarkan versi terbarunya, namun semakin baru tentunya diikuti dengan harga yang lebih mahal, maka dari itu Raspberry Pi versi yang lebih awal tetap menjadi pilihan, salah satunya Raspberry Pi 2. Raspberry Pi 2 merupakan komputer papan tunggal berbiaya rendah yang banyak digunakan dalam sistem pertanian presisi dan Internet of Things (IoT) (Upton & Halfacree, 2014). Meskipun memiliki kemampuan komputasi yang terbatas dibandingkan versi terbaru, perangkat ini tetap relevan untuk aplikasi lapangan karena efisiensi konsumsi daya, ketersediaan luas, dan kemudahan integrasi dengan sistem akuisisi citra. Dengan pemilihan arsitektur model yang tepat, Raspberry Pi 2 berpotensi digunakan sebagai solusi praktis untuk mendukung pengambilan keputusan panen di tingkat petani (Arumuga Perumal et al., 2017).

Berdasarkan latar belakang tersebut, penelitian ini mengembangkan sistem deteksi buah tomat dan klasifikasi tingkat kematangannya menggunakan algoritma YOLOv5n yang diimplementasikan pada Raspberry Pi 2. Kontribusi utama penelitian ini meliputi evaluasi performa deteksi pada perangkat *embedded* berdaya komputasi rendah, optimasi inferensi melalui konversi model ke format ONNX, serta validasi lapangan oleh praktisi pertanian. Hasil penelitian diharapkan dapat mendukung penerapan teknologi pertanian presisi yang akurat, efisien, dan terjangkau.

METODE

Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan pada tahun 2025. Proses pelatihan dan evaluasi model deep learning dilakukan menggunakan platform Google Colab dengan dukungan GPU NVIDIA Tesla T4. Implementasi sistem dan pengujian inferensi dilakukan pada perangkat Raspberry Pi 2 Model B. Validasi lapangan dilaksanakan pada tanggal 10 November 2025 dengan petani tomat dan pada tanggal 15 November 2025 dengan pedagang tomat di wilayah Kabupaten Badung, Provinsi Bali.

Alat dan Bahan

Penelitian ini menggunakan seperangkat perangkat keras dan perangkat lunak untuk membangun sistem deteksi tingkat kematangan buah tomat berbasis visi komputer. Perangkat keras utama berupa Raspberry Pi 2 Model B yang dilengkapi prosesor ARM Cortex-A7 quad-core 900 MHz dan RAM 1 GB sebagai unit pemrosesan, serta webcam Buffalo BSW20K04 dengan resolusi 640 × 480 piksel sebagai perangkat akuisisi citra. Media penyimpanan menggunakan microSD card 32 GB Class 10, didukung adaptor daya 5 V 3 A dan modul USB wireless TP-Link TL-WN725N untuk konektivitas jaringan. Proses pelatihan model dilakukan pada platform Google Colab dengan dukungan GPU NVIDIA Tesla T4 dan kapasitas RAM sebesar 25,51 GB guna mempercepat proses komputasi selama training model.

Perangkat lunak yang digunakan meliputi sistem operasi Raspberry Pi OS berbasis Debian Bullseye/Bookworm dan bahasa pemrograman Python versi 3.9. Model deteksi objek dibangun menggunakan arsitektur YOLOv5 berbasis framework PyTorch versi 1.12.1 dengan dukungan CUDA 11.3 pada lingkungan pelatihan, serta diimplementasikan pada perangkat Raspberry Pi menggunakan ONNX Runtime versi 1.10.0 sebagai inference engine. Pemrosesan citra dan visualisasi hasil deteksi dilakukan menggunakan OpenCV versi 4.5.x, sedangkan pengembangan sistem dilakukan melalui Visual Studio Code dengan ekstensi SSH Remote. Bahan penelitian berupa dataset citra buah tomat diperoleh dari LaboroTomato Dataset yang tersedia secara publik melalui Dataset Ninja (Laboro.AI, 2019) dengan lisensi Creative Commons Attribution–NonCommercial–ShareAlike 4.0, yang semula berjumlah 804 citra dan diseleksi menjadi 440 citra tomat dengan keseragaman morfologi jenis beef tomato yang terbagi ke dalam tiga tingkat kematangan, yaitu mentah, setengah matang, dan matang.

Persiapan Dataset

Dataset awal yang terdiri atas 440 citra buah tomat distandardisasi ke dalam resolusi 640 × 640 piksel untuk menyesuaikan kebutuhan input model YOLO dan menjaga konsistensi dimensi citra (Dwyer B et al., 2024). Proses pelabelan dilakukan menggunakan platform Roboflow dengan tiga kelas tingkat kematangan, yaitu mentah (class_id: 0), setengah matang (class_id: 1), dan matang (class_id: 2). Untuk meningkatkan keragaman data dan ketahanan model terhadap variasi kondisi visual, diterapkan augmentasi data berupa horizontal flip, rotasi hingga $\pm 10^\circ$, crop sebesar 0–20%, penyesuaian kecerahan hingga $\pm 15\%$, penyesuaian eksposur hingga $\pm 10\%$, serta blur dengan intensitas maksimum 1,5 piksel, sehingga diperoleh total 1.320 citra (Gonzalez & Woods, 2018). Dataset kemudian dibagi menjadi data latih sebanyak 923 citra (70%), data validasi sebanyak 198 citra (15%), dan data uji sebanyak 199 citra (15%) menggunakan metode stratifikasi untuk menjaga proporsi kelas kematangan yang seimbang pada setiap subset.

Pelatihan Model YOLOv5

Pelatihan model YOLOv5n dilakukan pada platform Google Colab dengan resolusi input sebesar 640 × 640 piksel, batch size 16, dan jumlah epoch sebanyak 100. Proses pelatihan memanfaatkan pretrained weights yolov5n.pt sebagai pendekatan transfer learning untuk meningkatkan konvergensi dan efisiensi pembelajaran model (Jocher et al., 2020). Dataset dimuat ke dalam memori menggunakan fitur cache guna mempercepat proses iterasi selama pelatihan. Pemantauan kinerja model dilakukan secara berkala melalui evaluasi nilai fungsi kehilangan (loss) serta metrik precision, recall, dan mean Average Precision (mAP) pada ambang Intersection over Union (IoU) sebesar 0,5 sebagai indikator utama performa deteksi objek (Zheng et al., 2019).

Ekspor dan Optimasi Model

Model terbaik hasil pelatihan (best.pt) dikonversi ke format ONNX menggunakan utilitas bawaan YOLOv5 dengan fitur simplifikasi graph aktif. Konversi ini menghasilkan file model ONNX berukuran sekitar 7,2 MB yang dioptimasi untuk arsitektur ARM Raspberry Pi 2. Verifikasi model dilakukan untuk memastikan konsistensi output antara format ONNX dan PyTorch original.

Implementasi pada Raspberry Pi

Sistem terintegrasi dengan webcam Buffalo BSW20K04 dan menggunakan *ONNX Runtime* sebagai *inference engine*. Tahapan pemrosesan meliputi: (1) akuisisi citra dengan resolusi 640×480 piksel, (2) preprocessing dengan konversi BGR ke RGB, resize menjadi 640×640 piksel, dan normalisasi, (3) inferensi menggunakan model ONNX, (4) post-processing dengan Non-Maximum Suppression (NMS) untuk eliminasi deteksi overlap, dan (5) visualisasi hasil dengan bounding box dan label pada frame menggunakan OpenCV.

Evaluasi Kinerja

Evaluasi dilakukan menggunakan enam parameter: *accuracy*, *precision*, *recall*, *F1-score*, *mean Average Precision* (mAP), dan waktu inferensi. Rumus yang digunakan adalah:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad [1]$$

$$Precision = \frac{TP}{TP+FP} \quad [2]$$

$$Recall = \frac{TP}{TP+FN} \quad [3]$$

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad [4]$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad [5]$$

$$Waktu Inferensi \times \frac{Total Waktu Proses}{Jumlah Gambar} \times 100 (ms) \quad [6]$$

dengan TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), dan FN (*False Negative*).

Validasi Lapangan

Validasi dilakukan dengan menampilkan 20 citra hasil deteksi sistem kepada empat responden (dua petani dan dua pedagang) yang memiliki pengalaman minimal 3 tahun. Penilaian dilakukan menggunakan skala tiga kategori: Sesuai (S), Kurang Sesuai (KS), dan Tidak Sesuai (TS). Responden dipilih berdasarkan kriteria validator yang kompeten dalam mengidentifikasi tingkat kematangan tomat.

HASIL DAN PEMBAHASAN

Persiapan dan Karakteristik Dataset

Dataset yang digunakan dalam penelitian ini berasal dari LaboroTomato Dataset, yang tersedia secara publik melalui platform Dataset Ninja dengan lisensi Creative Commons Attribution–NonCommercial–ShareAlike 4.0 (CC BY-NC-SA 4.0). Dataset ini berisi citra buah tomat (*Solanum lycopersicum*) yang telah dilabel secara manual pada tingkat objek individu. Dari total 804 citra awal yang tersedia, dilakukan proses seleksi data untuk menjaga keseragaman morfologi dan karakteristik visual objek, sehingga diperoleh sebanyak 440 citra tomat jenis *beef tomato* yang digunakan sebagai dataset resmi penelitian.

Setiap citra selanjutnya dianotasi menggunakan platform Roboflow dengan metode *bounding box annotation* dan diklasifikasikan ke dalam tiga kategori tingkat kematangan, yaitu mentah, setengah matang, dan matang. Hasil anotasi disimpan dalam format YOLOv5 (.txt) yang memuat informasi kelas objek, koordinat titik tengah (*center x*, *center y*), serta dimensi *bounding box* (*width* dan *height*) yang telah dinormalisasi terhadap ukuran citra. Format anotasi ini memungkinkan integrasi data secara langsung ke dalam proses pelatihan model YOLOv5n serta memastikan konsistensi struktur data selama tahap pengolahan dan pelatihan model.

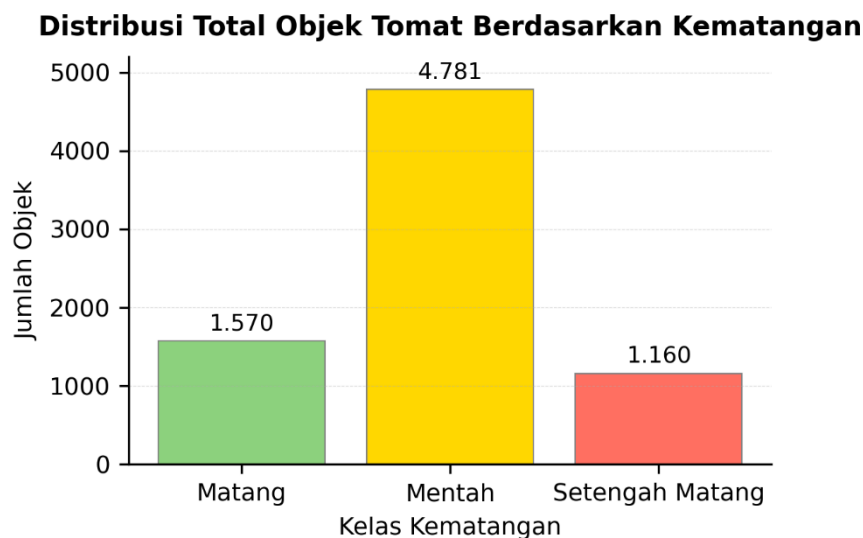


Gambar 1. Proses pelabelan dataset tomat pada platform Roboflow

Seluruh citra diseragamkan ke resolusi 640×640 piksel, disertai normalisasi nilai piksel (0–1) dan augmentasi data otomatis di Roboflow meliputi *horizontal flip*, rotasi ($\pm 10^\circ$), *crop* (0–20%), penyesuaian *brightness* dan *exposure* ($\pm 15\%$), serta *blur* ringan (≤ 1.5 px) untuk meningkatkan variasi citra.

Dataset kemudian dibagi dengan rasio 70:15:15 untuk training, validation, dan testing secara manual di Google Colab. Totalnya terdapat 7.511 objek dari 440 citra, dengan proporsi kelas mentah (63.6%), setengah matang (15.4%), dan matang (20.9%). Distribusi ini mencerminkan kondisi nyata di lahan pertanian, di mana buah mentah lebih dominan. Meskipun terjadi ketidakseimbangan kelas, model YOLOv5n menggunakan *weighted loss function* untuk menjaga keseimbangan pelatihan.

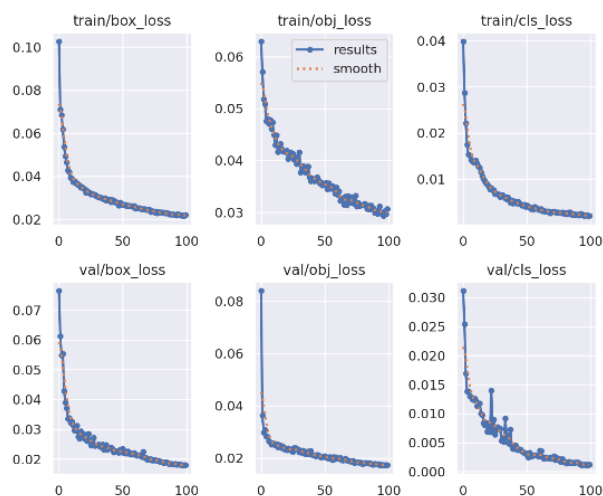
Struktur dataset diorganisasikan sesuai konvensi YOLOv5 dengan direktori *images* dan *labels* untuk masing-masing subset. Validasi akhir memastikan kesesuaian format, rasio 1:1 antara gambar dan label, serta distribusi kelas yang konsisten. Dataset ini dinyatakan siap digunakan pelatihan model YOLOv5n.



Gambar 2. Distribusi dataset setelah proses augmentasi dan pembagian

Pelatihan Model YOLOv5n

Pelatihan model dilakukan menggunakan platform Google Colab dengan akselerasi GPU NVIDIA Tesla T4 (16 GB VRAM) dan framework PyTorch versi 2.0.1. Model YOLOv5n dipilih karena memiliki ukuran parameter ringan (1.9 juta parameter) sehingga sesuai untuk implementasi pada Raspberry Pi 2 dengan daya komputasi terbatas. Arsitektur YOLOv5n terdiri dari backbone (CSPDarknet53) untuk ekstraksi fitur, neck (SPPF + PANet) untuk penggabungan fitur multi-skala, dan head untuk prediksi bounding box dan klasifikasi objek.

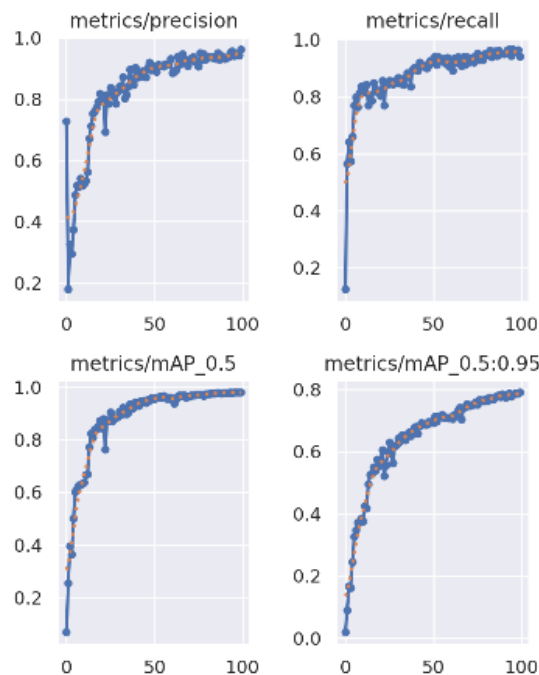


Gambar 3. Grafik perkembangan loss function selama pelatihan

Konfigurasi pelatihan menggunakan 100 epoch dengan *batch size* 16, ukuran input 640×640 piksel, *learning rate* 0.01, optimizer SGD dengan momentum 0.937 dan *weight decay* 0.0005. Pelatihan menggunakan *transfer learning* dari *pretrained weights* yolov5n.pt yang telah dilatih pada dataset COCO untuk mempercepat konvergensi. Fungsi kehilangan terdiri dari tiga komponen: *box loss* (CIoU) untuk akurasi lokalisasi, *objectness loss* (BCE) untuk *confidence* keberadaan objek, dan *classification loss* (BCE) untuk akurasi klasifikasi kelas dengan *weighted sum* untuk menangani ketidakseimbangan kelas dalam dataset.

Berdasarkan Gambar 3, nilai train/box_loss menurun dari 0.10 menjadi 0.02, sementara val/box_loss menurun dari 0.075 menjadi 0.018, mengindikasikan peningkatan akurasi lokalisasi yang konsisten. Penurunan train/obj_loss dan val/obj_loss dari rentang 0.06-0.08 menjadi 0.02-0.03 menunjukkan

kemampuan model dalam mengidentifikasi keberadaan objek semakin baik. Nilai train/cls_loss menurun dari 0.04 menjadi 0.007 dan val/cls_loss mencapai 0.003, menunjukkan akurasi klasifikasi yang tinggi. Pola penurunan loss yang konsisten pada data pelatihan dan validasi mengindikasikan bahwa model tidak mengalami overfitting signifikan, dengan konvergensi optimal tercapai sekitar epoch ke-20 hingga ke-35.



Gambar 4. Grafik evaluasi metrik precision, recall, dan mAP

Berdasarkan Gambar 4, nilai precision meningkat signifikan dari 0.20 pada epoch awal hingga mencapai 0.95-0.97 pada epoch akhir, menunjukkan kemampuan model dalam meminimalkan false positive. Nilai recall meningkat dari <0.30 hingga 0.90-0.95, mengindikasikan kemampuan model mendeteksi objek secara konsisten dengan false negative yang rendah. Metrik mAP@0.5 mencapai nilai tertinggi sekitar 0.97-0.99, menandakan akurasi spasial prediksi bounding box yang sangat baik. Nilai mAP@0.5:0.95 meningkat hingga 0.78-0.82, menunjukkan bahwa model tidak hanya akurat pada threshold IoU longgar (0.5) tetapi juga mempertahankan performa pada threshold yang lebih ketat. Peningkatan konsisten pada semua metrik menunjukkan bahwa model berhasil melakukan pembelajaran fitur secara progresif dan mencapai konvergensi optimal.

Model terbaik (best.pt) dipilih berdasarkan nilai mAP tertinggi pada data validasi dan disimpan pada epoch ke-98. Model ini kemudian diekspor ke format ONNX untuk implementasi pada Raspberry Pi dengan proses optimasi yang akan dijelaskan pada sub-bab berikutnya.

Evaluasi Kinerja Model

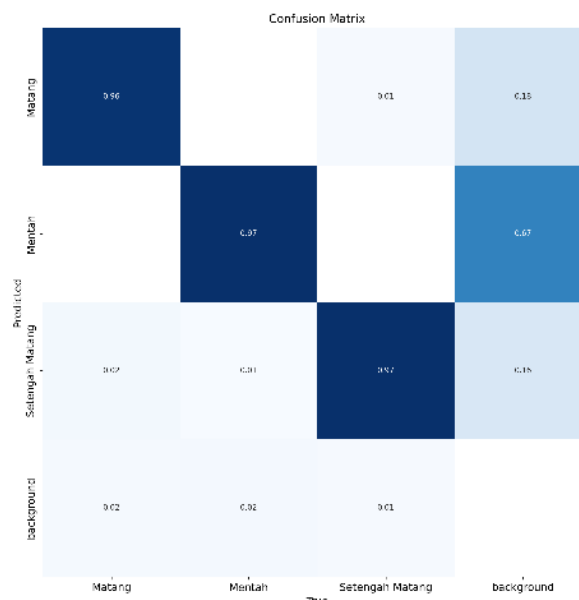
Evaluasi kinerja model dilakukan menggunakan 1167 objek pada 199 citra uji yang tidak pernah digunakan selama proses pelatihan atau validasi. Hasil evaluasi menunjukkan performa yang sangat baik dengan precision 96.28%, recall 94.22%, dan F1-score 95.24%. Nilai mean Average Precision pada threshold IoU 0.5 (mAP@0.5) mencapai 97.99%, sementara mAP@0.5:0.95 sebesar 79.06%. Nilai mAP@0.5 yang mendekati 98% menunjukkan bahwa model memiliki akurasi tinggi dalam mendeteksi lokasi objek dengan overlap minimal 50% terhadap ground truth, sedangkan nilai mAP@0.5:0.95 sebesar 79.06% mengindikasikan bahwa model mempertahankan performa yang baik bahkan pada threshold IoU yang lebih ketat hingga 95%.

Tabel 1. Evaluasi performa model per kelas pada dataset pengujian

Kelas	Precision (%)	Recall (%)	F1-Score (%)	Jumlah Objek
Mentah	97,00	94,00	95,40	737
Setengah Matang	96,00	93,00	94,40	186
Matang	96,00	96,00	96,00	244
Rata-rata	96,28	94,22	95,24	1.167

Berdasarkan Tabel 1, kelas matang menunjukkan performa terbaik dengan precision dan recall masing-masing 96.00%, diikuti oleh kelas mentah dengan precision 97.00% dan recall 94.00%. Kelas setengah matang memiliki precision 96.00% namun recall sedikit lebih rendah (93.00%), yang mengindikasikan bahwa beberapa objek tomat setengah matang tidak terdeteksi. Hal ini dapat disebabkan oleh variasi warna transisi dari hijau menuju merah yang masih mirip dengan kelas mentah atau matang, serta jumlah objek kelas setengah matang yang lebih sedikit (186 objek) dibandingkan kelas mentah (737 objek) sehingga model memiliki exposure yang lebih terbatas terhadap variasi visual kelas ini.

Berdasarkan Gambar 5, model menunjukkan akurasi klasifikasi yang tinggi dengan nilai diagonal 0.96 untuk kelas matang, 0.97 untuk kelas mentah, dan 0.97 untuk kelas setengah matang. Kesalahan klasifikasi terutama terjadi pada kelas setengah matang, dimana beberapa sampel terprediksi sebagai tomat matang (sekitar 2%) atau mentah (sekitar 1%). Fenomena ini dapat dijelaskan karena fase setengah matang memiliki karakteristik visual transisional dengan gradasi warna yang beragam, mulai dari dominan hijau dengan sedikit merah hingga dominan merah dengan sedikit hijau. Kondisi pencahayaan dan sudut pengambilan gambar juga mempengaruhi persepsi warna, sehingga meningkatkan ambiguitas dalam klasifikasi.

**Gambar 5.** Confusion matrix hasil pengujian model

Kesalahan klasifikasi minor juga terjadi pada kelas mentah yang terprediksi sebagai setengah matang (sekitar 2%), umumnya terjadi pada buah mentah yang mulai menunjukkan perubahan warna awal dari hijau tua ke hijau muda atau kekuningan. Meskipun demikian, tingkat kesalahan klasifikasi keseluruhan yang sangat rendah (<4% untuk semua kelas) menunjukkan bahwa model telah berhasil mempelajari fitur visual yang membedakan setiap kelas kematangan dengan baik.

Model juga menunjukkan kemampuan yang baik dalam menangani variasi ukuran objek, posisi objek yang tumpang tindih, dan kondisi pencahayaan yang berbeda dalam dataset pengujian.



Gambar 6. Visualisasi hasil deteksi pada citra uji

Berdasarkan Gambar 6, model mampu mendeteksi dan mengklasifikasikan buah tomat dengan confidence score tinggi (≥ 0.70) pada berbagai kondisi visual yang kompleks. Visualisasi menunjukkan bahwa model dapat menangani citra dengan multiple objects dimana beberapa buah tomat dengan tingkat kematangan berbeda muncul dalam satu frame secara bersamaan. Model menunjukkan kemampuan robust dalam mendeteksi objek yang saling tumpang tindih atau sebagian tertutup oleh daun dan cabang, dengan tetap memberikan bounding box yang akurat dan label klasifikasi yang tepat. Pada citra dengan variasi pencahayaan dan sudut pengambilan gambar yang berbeda, model tetap konsisten dalam memberikan prediksi dengan confidence score yang memadai untuk aplikasi praktis. Kemampuan model dalam menangani kondisi kompleks ini sangat penting untuk implementasi di lapangan, dimana kondisi visual tidak selalu ideal seperti pada dataset pelatihan yang terkontrol.

Performa model yang tinggi pada dataset pengujian dengan mAP@0.5 sebesar 97.99% dan F1-score 95.24% mengindikasikan bahwa model YOLOv5n yang dikembangkan memiliki kemampuan generalisasi yang sangat baik. Model tidak hanya mampu mengingat pola dari data pelatihan tetapi juga dapat menerapkan pengetahuan tersebut pada data baru dengan karakteristik visual yang beragam. Hasil ini menunjukkan bahwa kombinasi antara arsitektur YOLOv5n yang efisien, augmentasi data yang komprehensif, dan konfigurasi pelatihan yang optimal telah berhasil menghasilkan model deteksi kematangan tomat yang akurat dan robust untuk diimplementasikan pada aplikasi lapangan.

Validasi Lapangan oleh Petani dan Pedagang

Validasi lapangan dilakukan untuk menilai kesesuaian hasil klasifikasi tingkat kematangan buah tomat yang dihasilkan oleh sistem dengan penilaian praktisi di lapangan. Validator terdiri atas empat orang, yaitu dua petani tomat dan dua pedagang tomat, sebagaimana dijelaskan pada tahap validasi dalam draft skripsi.

Sebanyak 20 citra buah tomat hasil deteksi sistem digunakan sebagai sampel validasi, yang terdiri atas 7 citra tomat mentah, 6 citra tomat setengah matang, dan 7 citra tomat matang. Setiap citra dinilai oleh seluruh validator sehingga diperoleh total 80 penilaian. Validator diminta menilai tingkat kesesuaian hasil klasifikasi sistem terhadap kondisi nyata buah tomat berdasarkan pengalaman masing-masing.

Tabel 2. Tingkat persetujuan validator per kategori kematangan

Kategori Kematangan	Jumlah Citra	Total Penilaian	Penilaian Sesuai	Tingkat Persetujuan (%)
Mentah	7	28	25	89,29
Matang	7	28	24	85,71
Setengah Matang	6	24	16	66,67
Total	20	80	65	80,56

Hasil validasi menunjukkan bahwa dari total 80 penilaian, sebanyak 65 penilaian dinyatakan sesuai, sehingga diperoleh tingkat persetujuan keseluruhan sebesar 80,56%. Pada kategori tomat mentah, diperoleh 25 penilaian sesuai dari total 28 penilaian dengan tingkat persetujuan sebesar 89,29%. Kategori tomat matang memperoleh 24 penilaian sesuai dari total 28 penilaian dengan tingkat persetujuan sebesar 85,71%. Sementara itu, kategori tomat setengah matang memperoleh 16 penilaian sesuai dari total 24 penilaian dengan tingkat persetujuan sebesar 66,67%.

Tingkat persetujuan yang lebih rendah pada kategori tomat setengah matang menunjukkan bahwa fase kematangan transisional memiliki karakteristik visual yang lebih sulit dibedakan secara jelas dibandingkan fase mentah dan matang. Kondisi ini sejalan dengan hasil evaluasi kinerja model pada tahap pengujian, di mana kesalahan klasifikasi juga lebih banyak terjadi pada kelas setengah matang.

Secara keseluruhan, hasil validasi lapangan menunjukkan bahwa sistem deteksi dan klasifikasi kematangan buah tomat yang dikembangkan telah memiliki tingkat penerimaan yang baik oleh praktisi dan layak digunakan sebagai alat bantu penentuan waktu panen tomat di lapangan.

Berdasarkan hasil evaluasi kinerja model dan validasi lapangan oleh petani serta pedagang tomat, sistem deteksi dan klasifikasi tingkat kematangan buah tomat yang dikembangkan menunjukkan performa teknis yang tinggi serta tingkat penerimaan praktis yang baik. Temuan ini mengindikasikan bahwa pendekatan deteksi berbasis YOLOv5n tidak hanya efektif secara komputasional, tetapi juga relevan untuk diterapkan pada kondisi nyata di lapangan.

KESIMPULAN

Penelitian ini menghasilkan sistem deteksi buah tomat dan klasifikasi tingkat kematangan menggunakan algoritma YOLOv5n secara signifikan dapat diimplementasikan pada perangkat Raspberry Pi 2 untuk aplikasi lapangan. Hasil analisis menunjukkan bahwa model YOLOv5n yang dilatih menggunakan dataset LaboroTomato dengan 1.320 citra hasil augmentasi mencapai performa tinggi dengan precision sebesar 96,28%, recall sebesar 94,22%, F1-Score sebesar 95,24%, dan mAP@0.5 sebesar 97,99%. Evaluasi per kategori menunjukkan bahwa kelas matang dan mentah memiliki precision tertinggi masing-masing 96,00% dan 97,00%, sementara kelas setengah matang mencapai 96,00% dengan recall 93,00% yang mengindikasikan tantangan dalam fase transisi kematangan. Konversi model ke format ONNX terbukti efektif meningkatkan kecepatan inferensi sebesar 20,5% dengan waktu rata-rata 1,47 detik per frame pada Raspberry Pi 2. Sementara itu, validasi lapangan yang melibatkan empat praktisi menunjukkan tingkat persetujuan sebesar 80,56%, dengan persetujuan tertinggi pada kategori mentah (89,29%) dan matang (85,71%), serta terendah pada kategori setengah matang (66,67%). Secara umum, sistem deteksi kematangan tomat

berbasis YOLOv5n yang diimplementasikan pada Raspberry Pi 2 direkomendasikan sebagai teknologi yang layak dan efektif untuk mendukung pengambilan keputusan panen tomat baik secara teknis maupun praktis dalam aplikasi pertanian presisi di tingkat petani.

DAFTAR PUSTAKA

- Arumuga Perumal, V. S., Baskaran, K., & Rai, S. K. (2017). IMPLEMENTATION OF EFFECTIVE AND LOW-COST BUILDING MONITORING SYSTEM(BMS) USING RASPBERRY PI. *Energy Procedia*, 143, 179–185. <https://doi.org/10.1016/j.egypro.2017.12.668>
- Badan Pusat Statistik. (2024). *Statistik Hortikultura 2023: Tomat*. <https://www.bps.go.id>
- Biglari, A., & Tang, W. (2023). A Review of Embedded Machine Learning Based on Hardware, Application, and Sensing Scheme. *Sensors*, 23(4), 2131. <https://doi.org/10.3390/s23042131>
- Dwyer B, Nelson J, & Solawetz J. (2024). *Roboflow: Computer Vision Tools for Developers and Enterprises*. (atau *Roboflow Platform for Computer Vision*). Roboflow Inc. <https://roboflow.com>.
- Girshick, R. (2015). *Fast R-CNN*. 1440–1448. <http://arxiv.org/abs/1504.08083>
- Gonzalez, R. C. ., & Woods, R. E. . (2018). *Digital image processing*. Pearson.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. The MIT Press.
- Huang, C., Liu, C., & Zhao, Y. (2023). *Fruit recognition system based on Raspberry Pi and YOLOv5*. 12918(lccsct), 78. <https://doi.org/10.1117/12.3009436>
- Jocher, G., Chaurasia, A., & Qiu, J. (2020). *YOLOv5 by Ultralytics*. <https://github.com/ultralytics/yolov5>
- Kementerian Pertanian Republik Indonesia. (2023). *Outlook Komoditas Hortikultura: Tomat 2023*. <https://www.pertanian.go.id>
- Laboro.AI. (2019). *LaboroTomato: Instance Segmentation Dataset*. <https://github.com/laboroai/LaboroTomato>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Li, R., Ji, Z., Hu, S., Huang, X., Yang, J., & Li, W. (2023). Tomato Maturity Recognition Model Based on Improved YOLOv5 in Greenhouse. *Agronomy*, 13(2). <https://doi.org/10.3390/agronomy13020603>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015, June 4). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. <http://arxiv.org/abs/1506.01497>
- Upton, E., & Halfacree, G. (2014). *Raspberry Pi User Guide* (2nd ed.). Wiley.
- Wang, C., Wang, C., Wang, L., Wang, J., Liao, J., Li, Y., & Lan, Y. (2023). A Lightweight Cherry Tomato Maturity Real-Time Detection Algorithm Based on Improved YOLOV5n. *Agronomy*, 13(8). <https://doi.org/10.3390/agronomy13082106>
- Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2019). *Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression*. <http://arxiv.org/abs/1911.08287>