



## Deteksi Kematangan dan Kesegaran Tomat Berbasis Web Menggunakan Algoritma YOLOv8

Azfa Pratama Pujiyatno, Sahlan

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang, Tangerang Selatan, Indonesia

[azfapratamapujiyatno@gmail.com](mailto:azfapratamapujiyatno@gmail.com), [dosen02403@unpam.ac.id](mailto:dosen02403@unpam.ac.id)

### Abstrak

Penyortiran kematangan dan kesegaran tomat secara manual bergantung pada pengamatan visual sehingga berpotensi menghasilkan penilaian yang tidak konsisten dan memperlambat proses distribusi pangan. Penelitian ini bertujuan mengembangkan sistem berbasis web yang mampu mendeteksi kondisi tomat secara otomatis menggunakan algoritma You Only Look Once versi 8 (YOLOv8). Dataset penelitian terdiri atas 1.004 citra dengan 2.427 objek beranotasi yang dikelompokkan ke dalam tiga kelas, yaitu Tomat Segar, Tomat Matang, dan Tomat Tidak Layak. Anotasi objek dilakukan menggunakan bounding box pada platform Roboflow. Data kemudian melalui tahap augmentasi berupa pembalikan horizontal, rotasi, serta penyesuaian hue, kontras, saturasi, kecerahan, dan eksposur sebelum digunakan untuk melatih model. Kinerja deteksi dievaluasi menggunakan precision, recall, mean Average Precision pada ambang Intersection over Union 0,50 (mAP50), mAP50–95, dan confusion matrix. Model menghasilkan precision sebesar 0,924, recall 0,851, mAP50 0,964, dan mAP50–95 0,714. Pengujian confusion matrix pada 214 sampel menghasilkan akurasi keseluruhan 79,9%, dengan precision tertinggi pada kelas Tomat Segar sebesar 83,3% dan recall tertinggi pada kelas Tomat Matang sebesar 92,9%. Model selanjutnya diintegrasikan dengan antarmuka web berbasis HTML, CSS, dan JavaScript, backend PHP, serta modul inferensi Python untuk menampilkan bounding box, label kelas, dan nilai confidence. Hasil penelitian menunjukkan bahwa sistem dapat membantu penyortiran tomat secara lebih cepat dan objektif, meskipun ketepatan lokalisasi objek serta kemampuan menghadapi variasi kondisi nyata masih perlu ditingkatkan.

Kata kunci: Deteksi Kematangan Tomat, Kesegaran Tomat, YOLOv8, Deteksi Objek, Sistem Berbasis Web

### Abstract

Manual sorting of tomato ripeness and freshness relies on visual observation, potentially resulting in inconsistent assessments and slowing down food distribution processes. This research aims to develop a web-based system capable of automatically detecting tomato conditions using the You Only Look Once version 8 (YOLOv8) algorithm. The research dataset consists of 1,004 images with 2,427 annotated objects categorized into three classes: Fresh Tomato, Ripe Tomato, and Unfit Tomato. Object annotation was performed using bounding boxes on the Roboflow platform. The data then underwent augmentation procedures including horizontal flipping, rotation, as well as adjustments to hue, contrast, saturation, brightness, and exposure before being used to train the model. Detection performance was evaluated using precision, recall, mean Average Precision at an Intersection over Union threshold of 0.50 (mAP50), mAP50–95, and a confusion matrix. The model yielded a precision of 0.924, a recall of 0.851, an mAP50 of 0.964, and an mAP50–95 of 0.714. Confusion matrix testing on 214 samples produced an overall accuracy of 79.9%, with the highest precision found in the Fresh Tomato class at 83.3% and the highest recall in the Ripe Tomato class at 92.9%. The model was subsequently integrated with a web interface using HTML, CSS, and JavaScript, a PHP backend, and a Python inference module to display bounding boxes, class labels, and confidence values. The results indicate that the system can assist in sorting tomatoes faster and more objectively, although object localization precision and the ability to handle varying real-world conditions still require improvement.

Keywords: Tomato Ripeness Detection, Tomato Freshness, YOLOv8, Object Detection, Web-Based System.

### 1. Pendahuluan

Tomat merupakan komoditas hortikultura yang banyak dikonsumsi dan memiliki nilai penting dari aspek gizi maupun ekonomi. Kandungan vitamin, mineral, serat, dan senyawa bioaktif seperti likopen membuat tomat dimanfaatkan sebagai bahan pangan dalam berbagai bentuk produk (Collins et al., 2022). Meskipun demikian, tomat termasuk produk pertanian yang mudah mengalami perubahan kualitas setelah dipanen. Tingkat kematangan yang tidak seragam, kerusakan fisik selama penanganan, perubahan warna, dan pembusukan dapat menurunkan

mutu produk sebelum sampai kepada konsumen. Kerugian pascapanen tidak hanya dipengaruhi oleh proses produksi, tetapi juga oleh ketidaktepatan pemeriksaan, sortasi, penyimpanan, dan distribusi komoditas (Baladraf, 2024). Oleh karena itu, proses identifikasi kondisi tomat sebelum didistribusikan menjadi bagian penting dalam menjaga kualitas produk sekaligus mengurangi risiko tercampurnya tomat segar dengan tomat yang telah matang berlebihan atau tidak layak dikonsumsi.

Proses sortasi tomat pada tingkat produsen, pedagang, dan pelaku distribusi masih banyak mengandalkan pemeriksaan visual. Petugas umumnya memperhatikan warna kulit, bentuk, tekstur permukaan, dan tanda-tanda kerusakan untuk menentukan kondisi tomat. Pemeriksaan tersebut relatif mudah dilakukan dalam jumlah terbatas, tetapi menjadi kurang efisien ketika volume produk meningkat. Penilaian manual juga dipengaruhi oleh pengalaman, ketelitian, kelelahan, kondisi pencahayaan, dan persepsi masing-masing pemeriksa. Akibatnya, tomat dengan kondisi visual yang hampir serupa dapat dimasukkan ke kategori yang berbeda. Ketidakkonsistenan ini dapat menurunkan objektivitas penyortiran dan memengaruhi kualitas tomat selama proses penyimpanan serta distribusi. Sistem deteksi otomatis diperlukan untuk memberikan hasil pemeriksaan yang lebih cepat, konsisten, dan dapat digunakan sebagai pendukung keputusan.

Perkembangan *computer vision* dan *deep learning* memungkinkan karakteristik visual objek dipelajari secara otomatis langsung dari citra. Pendekatan *deep learning* telah menunjukkan kemampuan yang baik dalam berbagai tugas kecerdasan buatan karena dapat membangun representasi fitur secara bertingkat dari data yang digunakan selama pelatihan (Bengio et al., 2021). Dalam konteks pemeriksaan produk pertanian, karakteristik seperti warna, pola permukaan, bentuk, dan indikasi kerusakan dapat dipelajari sebagai pembeda antarkelas. Salah satu keluarga algoritma yang banyak digunakan untuk deteksi objek adalah *You Only Look Once* (YOLO). Algoritma ini melakukan prediksi lokasi dan kelas objek melalui satu proses inferensi sehingga mendukung deteksi yang relatif cepat (Cong et al., 2023). Kemampuan tersebut sesuai dengan kebutuhan penyortiran produk pertanian karena dalam satu citra dapat ditemukan satu atau beberapa objek yang perlu dikenali secara bersamaan.

Penerapan YOLO untuk pemeriksaan kualitas buah telah dilakukan dalam beberapa penelitian. Wibowo et al. (2023) menggunakan YOLOv5 untuk mendeteksi kondisi segar dan busuk pada apel, pisang, dan jeruk. Penelitian tersebut menunjukkan bahwa algoritma YOLO dapat digunakan untuk membedakan kondisi buah berdasarkan karakteristik visualnya. Namun, objek yang digunakan terdiri atas beberapa jenis buah dan pengelompokan utamanya masih terbatas pada kondisi segar dan busuk. Penelitian tersebut belum secara khusus menggabungkan penilaian tingkat kematangan dan kelayakan tomat dalam satu sistem deteksi. Padahal, dalam proses distribusi tomat, produk yang matang tidak selalu berarti rusak, sedangkan tomat segar dan tomat yang tidak layak membutuhkan perlakuan distribusi yang berbeda.

Rismayanti dan Rahmadewi (2025) menerapkan YOLOv8 untuk mendeteksi serta mengklasifikasikan tingkat kematangan mangga harum manis menjadi kategori belum matang, setengah matang, dan matang. Model yang dikembangkan memperoleh nilai *precision* sebesar 81,1%, *recall* sebesar 81,5%, dan *mean Average Precision* sebesar 79,8%. Hasil tersebut memperlihatkan potensi YOLOv8 dalam membedakan tingkat kematangan buah. Meskipun memiliki kedekatan dari sisi metode, penelitian tersebut berfokus pada buah mangga dan hanya membedakan tingkat kematangannya. Kondisi kesegaran atau ketidaklayakan konsumsi belum menjadi sasaran utama, sementara keluaran model juga belum diarahkan pada sistem berbasis web untuk mempermudah pemanfaatannya oleh pengguna.

Erwanto et al. (2024) mengembangkan sistem berbasis Android dengan YOLO untuk mendeteksi penyakit tanaman tomat melalui citra daun. Penelitian tersebut memperlihatkan bahwa model deteksi objek dapat diintegrasikan ke dalam aplikasi agar hasil analisis dapat diakses oleh pengguna. Namun, objek yang dianalisis berupa daun dan sasaran deteksinya adalah penyakit tanaman, bukan kondisi buah tomat setelah panen. Sementara itu, Hirzan et al. (2023) memperlihatkan bahwa varian YOLO dapat diimplementasikan pada sistem dengan sumber daya komputasi terbatas untuk mendeteksi dan menghitung objek. Temuan tersebut mendukung peluang penerapan YOLO dalam aplikasi praktis, tetapi belum membahas klasifikasi mutu produk pertanian ataupun integrasi model dengan proses sortasi pangan.

Berdasarkan penelitian terdahulu, *state of the art* penerapan YOLO pada bidang pertanian telah berkembang dari deteksi keberadaan buah, klasifikasi buah segar dan busuk, identifikasi kematangan, hingga deteksi penyakit tanaman. Namun, masih terdapat beberapa kesenjangan penelitian. Pertama, penelitian sebelumnya umumnya memisahkan identifikasi kematangan dari penilaian kesegaran atau kelayakan konsumsi. Kedua, belum banyak penelitian yang secara khusus mengelompokkan objek tomat ke dalam kategori operasional

untuk kebutuhan sortasi, yaitu Tomat Segar, Tomat Matang, dan Tomat Tidak Layak. Ketiga, hasil penelitian lebih banyak berfokus pada evaluasi model, sedangkan integrasi model ke dalam sistem yang mudah diakses oleh pengguna masih terbatas. Keempat, hubungan antara hasil deteksi visual dan kebutuhan efisiensi penyortiran pada tahap distribusi pangan belum menjadi perhatian utama. Kesenjangan tersebut menunjukkan perlunya penelitian yang tidak hanya menghasilkan model deteksi, tetapi juga menyediakan implementasi yang dapat dimanfaatkan melalui antarmuka yang praktis.

Kebaruan penelitian ini terletak pada pengembangan sistem deteksi berbasis YOLOv8 yang menggabungkan dua dimensi penilaian kualitas tomat, yaitu tingkat kematangan dan kelayakan konsumsi, ke dalam tiga kelas operasional: Tomat Segar, Tomat Matang, dan Tomat Tidak Layak. Model dilatih menggunakan 1.004 citra dengan total 2.427 objek hasil anotasi *bounding box*, yang terdiri atas 953 objek Tomat Segar, 513 objek Tomat Matang, dan 961 objek Tomat Tidak Layak. Berbeda dengan penelitian yang hanya mengevaluasi model secara eksperimental, model dalam penelitian ini diintegrasikan ke dalam aplikasi berbasis web. Pengguna dapat mengunggah citra tomat dan memperoleh keluaran berupa lokasi objek dalam bentuk *bounding box*, label kelas, serta nilai *confidence*. Integrasi tersebut dirancang untuk menjembatani hasil eksperimen *computer vision* dengan kebutuhan penggunaan yang lebih praktis dalam proses pemeriksaan dan penyortiran tomat.

Berdasarkan permasalahan dan kesenjangan tersebut, penelitian ini bertujuan menerapkan YOLOv8 untuk mendeteksi serta mengklasifikasikan tomat ke dalam kelas Tomat Segar, Tomat Matang, dan Tomat Tidak Layak; mengevaluasi kinerja model menggunakan *precision*, *recall*, mAP50, mAP50–95, dan *confusion matrix*; serta mengintegrasikan model ke dalam sistem berbasis web yang dapat menampilkan hasil deteksi kepada pengguna. Penelitian ini diarahkan untuk menjawab bagaimana YOLOv8 dapat mengenali kondisi tomat berdasarkan karakteristik visual, bagaimana kinerja model dalam mendeteksi ketiga kelas tersebut, dan bagaimana model dapat diimplementasikan dalam aplikasi web. Sistem yang dihasilkan diharapkan dapat menjadi perangkat pendukung penyortiran yang lebih cepat dan objektif serta menjadi dasar pengembangan sistem pemeriksaan kualitas tomat pada skala distribusi pangan yang lebih luas.

## 2. Metode Penelitian

Penelitian ini merupakan penelitian terapan dengan pendekatan eksperimental untuk mengembangkan model deteksi kondisi tomat sekaligus mengintegrasikannya ke dalam aplikasi berbasis web. Model deteksi objek dibangun menggunakan YOLOv8n dari Ultralytics, sedangkan pengembangan aplikasi dilakukan secara iteratif menggunakan metode *Agile*. Tahapan penelitian meliputi pengumpulan citra, anotasi objek, pembagian dataset, prapemrosesan dan augmentasi, pelatihan YOLOv8n, evaluasi model, pengembangan aplikasi web, serta pengujian integrasi sistem. YOLOv8 dipilih karena menggunakan pendekatan deteksi satu tahap yang memungkinkan lokasi dan kelas objek diprediksi dalam satu proses inferensi (Cong et al., 2023).

### 2.1 Tahapan penelitian

Penelitian dimulai dengan mengumpulkan citra buah tomat yang memperlihatkan variasi tingkat kematangan dan kondisi kelayakan. Setiap objek tomat dianotasi menggunakan *bounding box* dan diberikan salah satu dari tiga label kelas, yaitu Tidak Layak, Tomat Segar, atau Tomat Matang. Dataset selanjutnya dibagi menjadi data pelatihan, validasi, dan pengujian. Data pelatihan digunakan untuk memperbarui parameter model, data validasi digunakan untuk memantau perubahan kinerja selama pelatihan, sedangkan data pengujian digunakan untuk mengevaluasi kemampuan model terhadap data yang tidak digunakan dalam pembelajaran.

Citra menjalani proses penyeragaman ukuran dan augmentasi sebelum digunakan untuk melatih YOLOv8n. Model terbaik yang dihasilkan dari proses pelatihan disimpan dalam berkas *best.pt*, kemudian diintegrasikan ke dalam aplikasi berbasis web melalui modul inferensi Python. Sistem akhir diuji mulai dari proses pemilihan atau pengambilan citra, pengiriman citra ke server, inferensi model, hingga penyajian hasil deteksi kepada pengguna.



Gambar 1. Tahapan penelitian deteksi kematangan dan kesegaran tomat menggunakan YOLOv8n

## 2.2 Dataset dan anotasi

Dataset awal terdiri atas 1.004 citra yang memuat satu atau beberapa objek tomat dengan kondisi visual yang berbeda. Anotasi dilakukan menggunakan platform Roboflow dengan menggambar *bounding box* yang mengikuti batas setiap objek tomat. Setiap anotasi memuat koordinat lokasi objek dan label kelasnya. Proses tersebut menghasilkan 2.427 objek, yang terdiri atas 961 objek kelas Tidak Layak, 953 objek kelas Tomat Segar, dan 513 objek kelas Tomat Matang. Distribusi objek diperlihatkan pada Tabel 1.

Tabel 1. Distribusi objek pada setiap kelas

| Kelas        | Label | Jumlah objek | Persentase |
|--------------|-------|--------------|------------|
| Tidak Layak  | 0     | 961          | 39,60%     |
| Tomat Segar  | 1     | 953          | 39,27%     |
| Tomat Matang | 2     | 513          | 21,14%     |
| Total        | —     | 2.427        | 100,00%    |

Tabel 1 menunjukkan bahwa kelas Tidak Layak dan Tomat Segar mempunyai jumlah objek yang relatif seimbang, sedangkan kelas Tomat Matang memiliki jumlah objek lebih sedikit. Perbedaan tersebut dipertahankan sesuai distribusi data yang tersedia dan menjadi pertimbangan ketika hasil evaluasi dianalisis per kelas. Pembagian data menjadi subset pelatihan, validasi, dan pengujian dilakukan pada tingkat citra agar anotasi objek dalam satu citra tidak tersebar pada lebih dari satu subset. Pemisahan pada tingkat citra juga diperlukan untuk mencegah citra yang sama digunakan dalam proses pelatihan sekaligus pengujian.

## 2.3 Prapemrosesan dan augmentasi

Citra dan anotasi terlebih dahulu diperiksa untuk memastikan setiap *bounding box* berada pada objek yang tepat serta memiliki label kelas yang sesuai. Citra kemudian diseragamkan menggunakan metode *fit/letterbox* menjadi  $512 \times 512$  piksel. Metode ini mempertahankan rasio aspek citra dan menambahkan area pengisi apabila perbandingan panjang dan lebar citra tidak sesuai dengan ukuran target. Pada saat pelatihan, citra dimuat oleh YOLOv8n menggunakan parameter *imgsz* sebesar 640 piksel. Dengan demikian, ukuran  $512 \times 512$  piksel merupakan ukuran hasil prapemrosesan dataset, sedangkan 640 piksel merupakan ukuran masukan yang digunakan oleh *data loader* selama pelatihan model.

Augmentasi diterapkan pada data pelatihan untuk meningkatkan variasi posisi, warna, dan pencahayaan tanpa mengubah label objek. Transformasi yang digunakan mencakup pembalikan horizontal, rotasi  $90^\circ$  searah dan berlawanan arah jarum jam, perubahan *hue* antara  $-15^\circ$  sampai  $+15^\circ$ , serta penyesuaian kontras menggunakan *adaptive equalization*. Tingkat saturasi diubah dalam rentang  $-25\%$  sampai  $+25\%$ , kecerahan dalam rentang  $-15\%$  sampai  $+15\%$ , dan eksposur dalam rentang  $-10\%$  sampai  $+10\%$ . Transformasi geometris diterapkan secara bersamaan pada citra dan koordinat *bounding box* agar lokasi anotasi tetap sesuai dengan objek. Augmentasi hanya

diterapkan pada data pelatihan, sedangkan data validasi dan pengujian dipertahankan tanpa augmentasi untuk memperoleh evaluasi yang objektif.

## 2.4 Konfigurasi pelatihan YOLOv8n

Pelatihan dilaksanakan menggunakan Google Colaboratory dengan dukungan GPU dan kerangka kerja Ultralytics YOLOv8. Varian yang digunakan adalah YOLOv8n karena memiliki ukuran relatif ringan sehingga sesuai untuk diintegrasikan dengan aplikasi berbasis web. Model dilatih selama 100 *epoch* menggunakan ukuran *batch* 16 dan resolusi masukan 640 piksel. Konfigurasi pelatihan dirangkum pada Tabel 2.

*Tabel 2. Konfigurasi pelatihan model*

| Parameter            | Nilai                   |
|----------------------|-------------------------|
| Model                | YOLOv8n                 |
| Jumlah kelas         | 3                       |
| Epoch                | 100                     |
| Batch size           | 16                      |
| Image size           | 640 piksel              |
| Optimizer            | SGD/default Ultralytics |
| Lingkungan pelatihan | Google Colaboratory     |
| Framework            | Ultralytics YOLOv8      |
| Format anotasi       | YOLO bounding box       |
| Model keluaran       | best.pt                 |

Pada setiap *epoch*, model memproses seluruh data pelatihan dan memperbarui bobot berdasarkan kesalahan lokalisasi serta klasifikasi objek. Kinerja model dipantau menggunakan data validasi melalui perubahan nilai *loss*, *precision*, *recall*, mAP50, dan mAP50–95. Bobot dengan hasil validasi terbaik digunakan sebagai model akhir untuk pengujian dan integrasi dengan aplikasi web.

## 2.5 Evaluasi kinerja model

Evaluasi dilakukan menggunakan *confusion matrix*, *precision*, *recall*, *F1-score*, *Intersection over Union* (IoU), *Average Precision* (AP), mAP50, dan mAP50–95. Suatu hasil deteksi dinilai benar ketika kelas prediksi sesuai dengan kelas sebenarnya dan nilai IoU memenuhi ambang evaluasi. *Precision* dihitung menggunakan Persamaan (1).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

Pada Persamaan (1), *TP* adalah jumlah objek yang terdeteksi dengan benar, sedangkan *FP* adalah jumlah objek yang diprediksi sebagai kelas tertentu tetapi prediksinya salah. Nilai *precision* menunjukkan proporsi prediksi positif yang benar terhadap seluruh prediksi positif yang dihasilkan model.

Kemampuan model menemukan seluruh objek yang seharusnya terdeteksi dihitung menggunakan *recall* pada Persamaan (2).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

Pada Persamaan (2), *FN* adalah jumlah objek sebenarnya yang tidak berhasil dideteksi atau diklasifikasikan dengan benar. Nilai *recall* yang tinggi menunjukkan bahwa sebagian besar objek dalam citra berhasil ditemukan oleh model.

Keseimbangan antara *precision* dan *recall* dihitung menggunakan *F1-score* pada Persamaan (3).

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

Nilai *F1-score* mendekati satu menunjukkan keseimbangan yang semakin baik antara ketepatan prediksi dan kemampuan model menemukan objek. Kesesuaian posisi kotak prediksi dengan kotak anotasi dihitung menggunakan IoU sebagaimana ditunjukkan pada Persamaan (4).

$$IoU = \frac{\text{Luas}(B_p \cap B_g)}{\text{Luas}(B_p \cup B_g)} \quad (4)$$

Pada Persamaan (4),  $B_p$  merupakan area *bounding box* hasil prediksi dan  $B_g$  merupakan area *bounding box* sebenarnya atau *ground truth*. Simbol irisan menunjukkan luas area yang bertumpang tindih, sedangkan simbol gabungan menunjukkan keseluruhan area yang dicakup oleh kedua *bounding box*.

Nilai AP dihitung berdasarkan luas di bawah kurva *precision-recall* untuk setiap kelas. Rata-rata AP dari seluruh kelas kemudian dihitung menggunakan Persamaan (5).

$$mAP = \frac{1}{C} \sum_{c=1}^C AP_c \quad (5)$$

Pada Persamaan (5),  $C$  merupakan jumlah kelas yang dievaluasi dan  $AP_c$  merupakan nilai *Average Precision* untuk kelas ke- $c$ . Penelitian ini menggunakan tiga kelas sehingga nilai mAP diperoleh dari rata-rata AP kelas Tidak Layak, Tomat Segar, dan Tomat Matang. mAP50 dihitung pada ambang IoU 0,50, sedangkan mAP50–95 merupakan rata-rata mAP pada ambang IoU 0,50 sampai 0,95 dengan kenaikan 0,05. mAP50 digunakan untuk menilai keberhasilan deteksi secara umum, sedangkan mAP50–95 memberikan penilaian yang lebih ketat terhadap ketepatan lokalisasi *bounding box*.

## 2.6 Pengembangan dan pengujian sistem web

Pengembangan sistem dilakukan secara iteratif melalui tahapan penyusunan kebutuhan, perencanaan fitur, implementasi, pengujian, serta evaluasi. Kebutuhan utama sistem mencakup kemampuan memilih citra dari perangkat, mengambil citra melalui kamera, mengirimkan citra ke server, menjalankan inferensi YOLOv8n, dan menampilkan hasil deteksi. Antarmuka dikembangkan menggunakan HTML, CSS, dan JavaScript, sedangkan bagian *backend* menggunakan PHP Native. Modul kecerdasan buatan dikembangkan menggunakan Python, OpenCV, dan Ultralytics YOLOv8.

Komunikasi antara antarmuka dan *backend* menggunakan *Fetch API* dengan metode POST menuju *endpoint* detect.php. Pengguna dapat memilih berkas citra atau mengambil gambar secara langsung melalui kamera menggunakan *MediaDevices API*. Berkas gambar dibaca melalui *FileReader API*, divalidasi oleh *backend*, dan disimpan sementara untuk proses inferensi. PHP selanjutnya menjalankan modul Python sebagai *subprocess*. Modul tersebut memuat model best.pt, membaca citra, melakukan prediksi, dan menghasilkan koordinat *bounding box*, label kelas, nilai *confidence*, serta citra yang telah diberi anotasi.

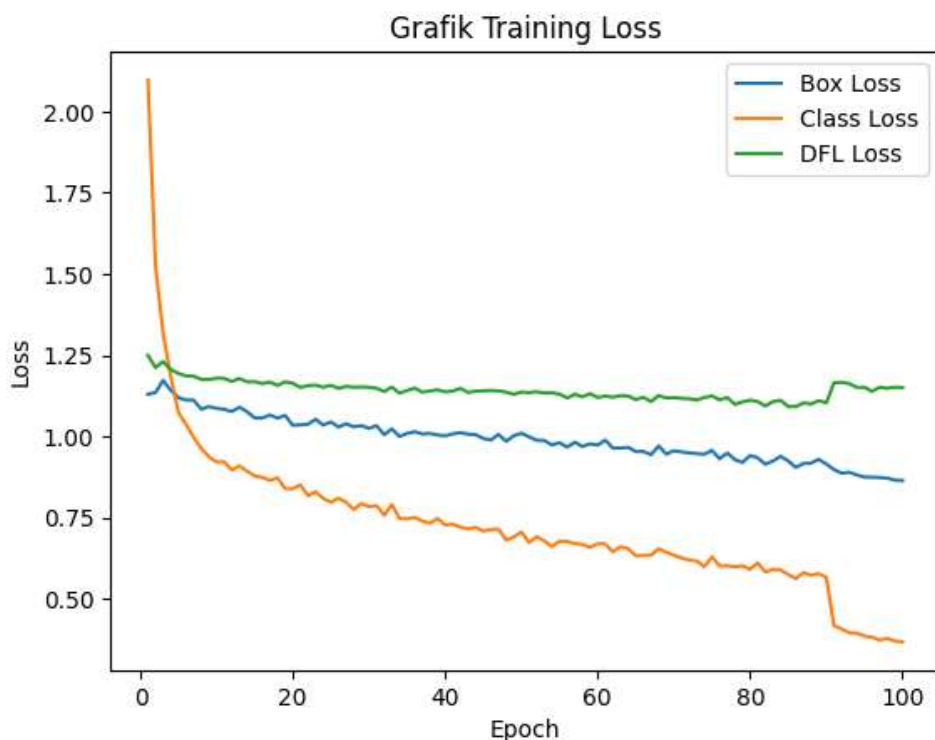
Hasil inferensi dikirim kembali ke halaman web untuk ditampilkan kepada pengguna. Sistem tidak menggunakan basis data permanen karena citra hanya disimpan selama proses deteksi berlangsung. Pengujian fungsional dilakukan terhadap proses pemilihan berkas, akses kamera, validasi dan pengiriman citra, pemanggilan model, serta penyajian hasil deteksi. Fungsi dinyatakan berjalan apabila sistem dapat menerima citra yang valid dan menampilkan lokasi objek, kelas kondisi tomat, serta nilai *confidence* tanpa mengubah data sumber secara permanen.

### 3. Hasil dan Pembahasan

Pelatihan model YOLOv8n dilakukan selama 100 *epoch* menggunakan ukuran *batch* 16 dan resolusi masukan 640 piksel. Selama proses pelatihan, perubahan kinerja diamati melalui *box loss*, *classification loss*, *distribution focal loss*, *precision*, *recall*, mAP50, dan mAP50–95 pada data pelatihan dan validasi. Hasil pelatihan memperlihatkan bahwa ketiga komponen *loss* mengalami penurunan secara bertahap. Penurunan tersebut menunjukkan bahwa pembaruan bobot model mampu mengurangi kesalahan klasifikasi dan lokalisasi objek selama proses pembelajaran.

Nilai *classification loss* menurun relatif cepat pada tahap awal pelatihan. Pola tersebut mengindikasikan bahwa model dapat mempelajari perbedaan karakteristik visual antara Tomat Segar, Tomat Matang, dan Tomat Tidak Layak. Perbedaan antarkelas terutama direpresentasikan melalui warna, tekstur permukaan, bentuk, dan keberadaan tanda kerusakan. Sementara itu, penurunan *box loss* dan *distribution focal loss* berlangsung lebih bertahap. Kondisi tersebut menunjukkan bahwa pembelajaran lokasi objek lebih sulit dibandingkan pembelajaran label kelas karena model harus menyesuaikan batas *bounding box* dengan ukuran, posisi, dan orientasi tomat yang bervariasi.

Nilai *precision* dan *recall* berfluktuasi pada beberapa *epoch* awal, kemudian meningkat dan menjadi lebih stabil mendekati akhir pelatihan. Fluktuasi awal menunjukkan bahwa model masih menyesuaikan representasi fitur dengan variasi citra dan distribusi objek pada setiap kelas. Setelah model memperoleh representasi yang lebih konsisten, perubahan kedua metrik menjadi lebih kecil. Pola peningkatan mAP yang diikuti kestabilan pada *epoch* akhir menunjukkan bahwa proses optimisasi telah mendekati kondisi konvergen. Namun, kestabilan kurva pelatihan tidak secara otomatis membuktikan bahwa model bebas dari *overfitting*. Penilaian tetap harus mempertimbangkan kesesuaian pola antara data pelatihan dan validasi serta kinerja pada data yang tidak digunakan untuk memperbarui bobot.



Gambar 2. Perkembangan loss dan metrik evaluasi selama pelatihan YOLOv8n

#### 3.2 Kinerja keseluruhan model

Model terbaik yang diperoleh selama pelatihan menghasilkan *precision* keseluruhan sebesar 0,924 dan *recall* sebesar 0,851. Nilai mAP50 mencapai 0,964, sedangkan mAP50–95 sebesar 0,714. Ringkasan hasil evaluasi

model disajikan pada Tabel 3. Nilai *precision* 0,924 menunjukkan bahwa sebagian besar objek yang diprediksi sebagai tomat dari kelas tertentu merupakan prediksi yang tepat. Nilai *recall* 0,851 memperlihatkan bahwa model berhasil menemukan sebagian besar objek yang terdapat pada data evaluasi, meskipun masih terdapat objek yang tidak terdeteksi atau tidak diklasifikasikan dengan tepat.

Tabel 3. Ringkasan kinerja model YOLOv8n

| Metrik           | Nilai | Interpretasi                                              |
|------------------|-------|-----------------------------------------------------------|
| <b>Precision</b> | 0,924 | Tingkat ketepatan prediksi positif sebesar 92,4%          |
| <b>Recall</b>    | 0,851 | Proporsi objek yang berhasil ditemukan sebesar 85,1%      |
| <b>mAP50</b>     | 0,964 | Kinerja deteksi pada ambang IoU 0,50 sebesar 96,4%        |
| <b>mAP50-95</b>  | 0,714 | Kinerja rata-rata pada ambang IoU 0,50-0,95 sebesar 71,4% |

Perbedaan sebesar 0,250 antara mAP50 dan mAP50-95 menjadi temuan penting dalam penelitian ini. Nilai mAP50 yang tinggi menunjukkan bahwa model mampu mengenali keberadaan dan kelas objek ketika kesesuaian *bounding box* dinilai pada ambang IoU 0,50. Akan tetapi, penurunan pada mAP50-95 menunjukkan bahwa kinerja berkurang ketika evaluasi menuntut kesesuaian posisi *bounding box* yang lebih presisi. Dengan demikian, kekuatan utama model terletak pada pengenalan objek dan kelasnya, sedangkan ketelitian batas objek masih memiliki ruang perbaikan.

Selisih antara mAP50 dan mAP50-95 dapat dipengaruhi oleh variasi bentuk serta ukuran tomat, posisi objek yang saling berdekatan, dan ketepatan anotasi. Citra yang memuat beberapa tomat dalam satu area dapat menyulitkan model menentukan batas masing-masing objek secara presisi. Perubahan akibat rotasi dan transformasi citra juga dapat meningkatkan keragaman data, tetapi kualitas koordinat *bounding box* harus tetap terjaga. Oleh sebab itu, nilai mAP50 yang tinggi tidak boleh ditafsirkan sebagai bukti bahwa seluruh aspek deteksi telah sempurna. Hasil mAP50-95 memberikan penilaian yang lebih konservatif dan lebih relevan ketika ketepatan lokasi objek menjadi kebutuhan utama.

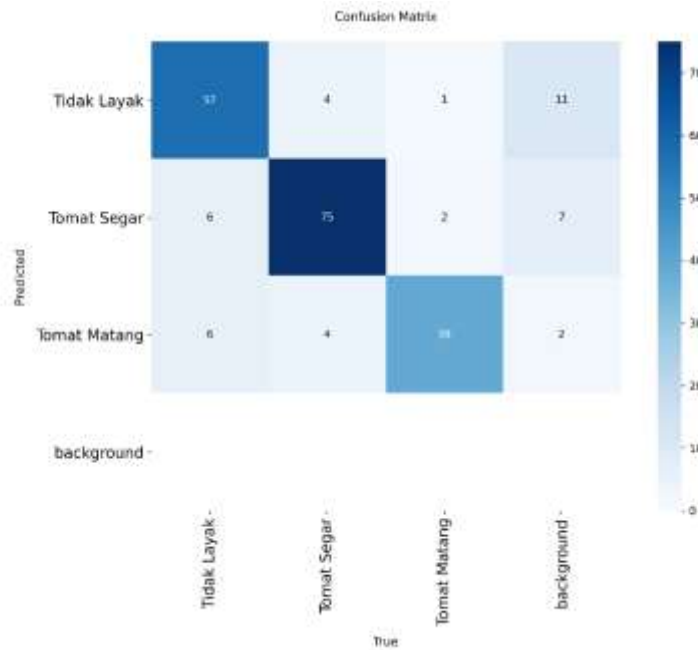
### 3.3 Kinerja setiap kelas dan analisis kesalahan

Analisis *confusion matrix* dilakukan untuk mengetahui kemampuan model dalam membedakan kelas Tidak Layak, Tomat Segar, dan Tomat Matang. Dari 214 entri evaluasi yang tercantum pada matriks, sebanyak 171 prediksi berada pada diagonal utama untuk tiga kelas tomat. Prediksi benar tersebut terdiri atas 57 objek Tidak Layak, 75 objek Tomat Segar, dan 39 objek Tomat Matang. Perbandingan jumlah prediksi diagonal dengan 214 entri menghasilkan nilai deskriptif sebesar 79,9%. Meskipun demikian, nilai tersebut tidak dijadikan indikator utama karena penelitian ini merupakan deteksi objek. Penilaian utama tetap menggunakan *precision*, *recall*, dan mAP yang mempertimbangkan kesesuaian kelas serta posisi *bounding box*.

Hasil evaluasi per kelas disajikan pada Tabel 4. Nilai *F1-score* dihitung dari *precision* dan *recall* setiap kelas. Kelas Tomat Segar menghasilkan *precision* tertinggi sebesar 83,3% dengan *recall* sebesar 90,4%, sehingga menghasilkan *F1-score* sekitar 86,7%. Kelas Tomat Matang memperoleh *recall* tertinggi sebesar 92,9%, tetapi *precision*-nya merupakan yang terendah, yaitu 76,5%. Kelas Tidak Layak menghasilkan *precision* 78,1%, *recall* 82,6%, dan *F1-score* sekitar 80,3%.

Tabel 4. Kinerja deteksi pada setiap kelas tomat

| Kelas               | Prediksi benar | Objek sebenarnya | Precision | Recall | F1-score |
|---------------------|----------------|------------------|-----------|--------|----------|
| <b>Tidak Layak</b>  | 57             | 69               | 78,1%     | 82,6%  | 80,3%    |
| <b>Tomat Segar</b>  | 75             | 83               | 83,3%     | 90,4%  | 86,7%    |
| <b>Tomat Matang</b> | 39             | 42               | 76,5%     | 92,9%  | 83,9%    |



Gambar 3. Confusion matrix hasil deteksi kondisi tomat

Kinerja terbaik berdasarkan *F1-score* diperoleh kelas Tomat Segar. Hasil ini menunjukkan bahwa karakteristik visual tomat segar relatif dapat dikenali secara konsisten. Jumlah objek Tomat Segar yang mencapai 953 juga menyediakan variasi sampel yang lebih besar daripada kelas Tomat Matang. Namun, pengaruh jumlah data tidak dapat dipisahkan sepenuhnya dari tingkat kejelasan karakteristik visual. Permukaan yang relatif baik dan warna yang masih seragam dapat menjadi fitur yang lebih mudah dipelajari daripada perubahan kualitas yang bersifat bertahap.

Kelas Tomat Matang memperlihatkan pola berbeda. *Recall* sebesar 92,9% menunjukkan bahwa hampir seluruh objek Tomat Matang berhasil ditemukan. Akan tetapi, *precision* sebesar 76,5% menunjukkan adanya objek dari kondisi lain yang turut diprediksi sebagai Tomat Matang. Kesalahan tersebut dapat terjadi karena kematangan tomat merupakan kondisi bertahap, bukan kategori yang selalu memiliki batas visual tegas. Warna kemerahan pada tomat segar dapat menyerupai tomat matang, sedangkan perubahan warna pada tomat yang mulai mengalami penurunan kualitas dapat menyerupai ciri kematangan. Kondisi pencahayaan juga dapat mengubah intensitas warna yang diterima model.

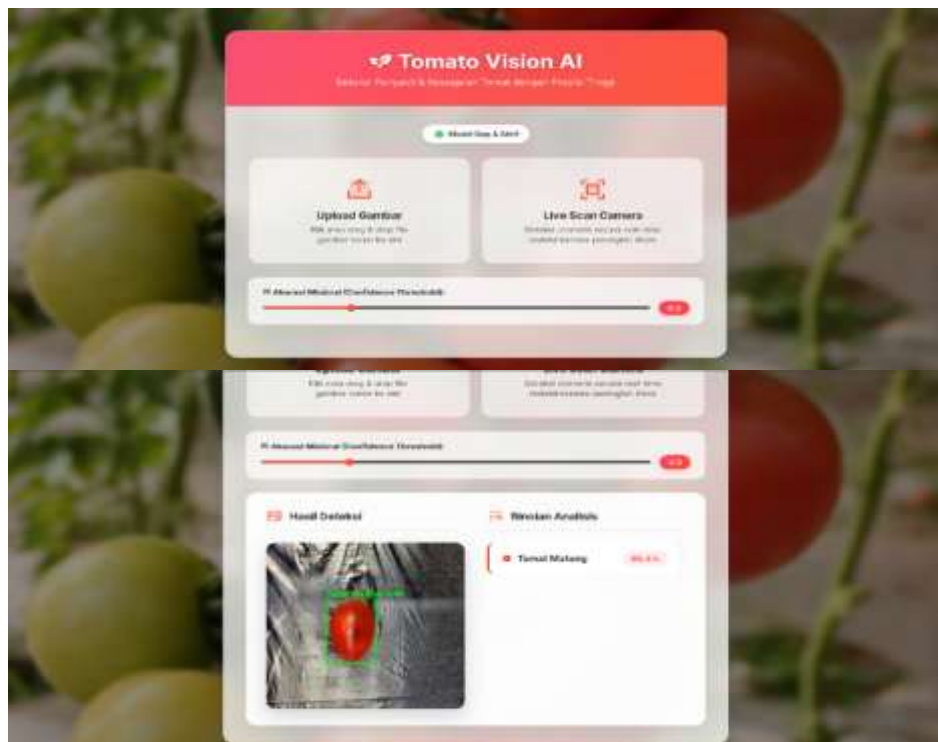
Kelas Tidak Layak memperoleh *recall* paling rendah di antara ketiga kelas, yaitu 82,6%. Artinya, terdapat objek tidak layak yang belum berhasil dikenali dengan benar. Kondisi tidak layak dapat muncul dalam berbagai bentuk, seperti bercak gelap, perubahan tekstur, kerusakan sebagian, dan pembusukan dengan tingkat berbeda. Keragaman visual yang tinggi membuat pola kelas ini lebih kompleks dibandingkan kelas yang terutama dibedakan berdasarkan warna kematangan. Kesalahan mengenali kelas Tidak Layak perlu mendapat perhatian karena dalam penggunaan praktis, objek yang tidak layak tetapi diprediksi sebagai kelas lain dapat lolos dari proses sortasi.

Matriks evaluasi juga menampilkan kategori *background* dengan 20 entri dan nilai *true positive* nol. Dalam evaluasi deteksi objek, *background* umumnya bukan kelas objek yang dilatih, melainkan representasi prediksi yang tidak memiliki pasangan dengan anotasi atau objek anotasi yang tidak memperoleh pasangan prediksi. Oleh karena itu, kategori tersebut tidak tepat ditafsirkan seperti kelas Tomat Segar, Tomat Matang, dan Tidak Layak. Kemunculannya menunjukkan masih terdapat *false positive* atau *false negative*, bukan berarti model gagal mengklasifikasikan 20 citra latar belakang sebagai satu kelas khusus. Interpretasi ini penting agar hasil *confusion matrix* tidak menghasilkan kesimpulan yang berlebihan.

### 3.4 Implementasi model pada aplikasi web

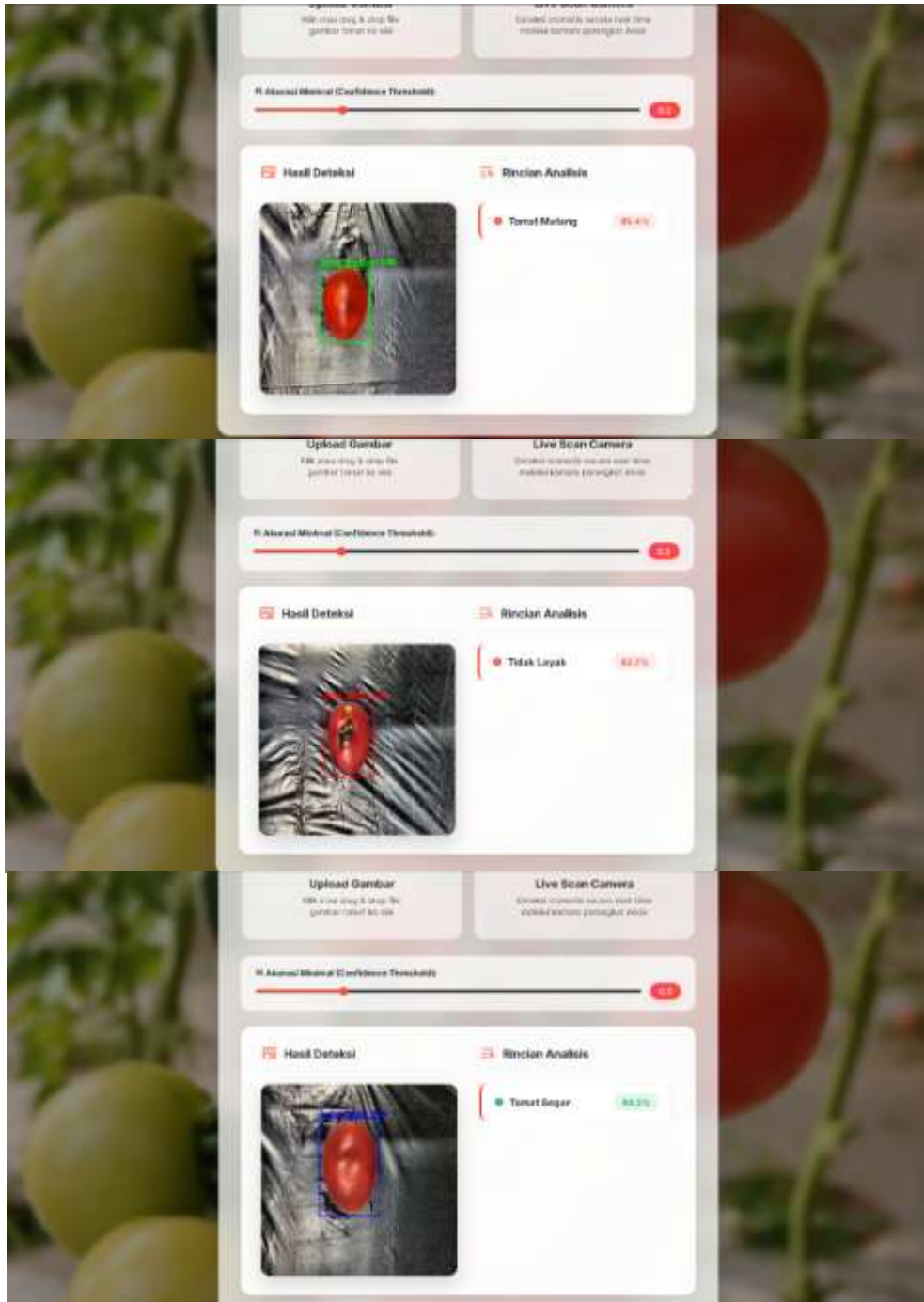
Model *best.pt* berhasil diintegrasikan ke dalam aplikasi berbasis web yang terdiri atas antarmuka HTML, CSS, dan JavaScript, *backend* PHP Native, serta modul inferensi Python. Pengguna dapat memilih citra dari perangkat atau mengambil gambar melalui kamera. Citra dikirim menggunakan *Fetch API* ke *endpoint* PHP, kemudian divalidasi dan disimpan sementara. Modul Python memuat model YOLOv8n, melakukan inferensi, dan mengembalikan koordinat *bounding box*, label kelas, nilai *confidence*, serta citra yang telah diberi anotasi.

Pengujian alur sistem menunjukkan bahwa komponen antarmuka, *backend*, dan modul inferensi dapat berkomunikasi sesuai rancangan. Setelah pengguna mengunggah citra, sistem menampilkan objek yang ditemukan dengan *bounding box* beserta label Tidak Layak, Tomat Segar, atau Tomat Matang. Nilai *confidence* ditampilkan untuk menunjukkan tingkat keyakinan model terhadap setiap prediksi. Data citra hanya digunakan selama pemrosesan dan tidak disimpan ke dalam basis data permanen.



Gambar 4. Antarmuka pengunggahan citra dan penyajian hasil deteksi

Gambar 4 sebaiknya dibuat sebagai satu gambar gabungan agar proses penggunaan sistem dapat dipahami tanpa menampilkan beberapa tangkapan layar yang menyampaikan informasi serupa. Halaman utama yang hanya berisi informasi pengantar tidak perlu ditampilkan apabila tidak memperlihatkan fungsi penelitian. Visualisasi sebaiknya difokuskan pada proses pemilihan citra dan keluaran deteksi karena kedua bagian tersebut membuktikan bahwa model telah terintegrasi dengan aplikasi.



Gambar 5. Contoh hasil deteksi pada tiga kelas kondisi tomat

Contoh keluaran pada Gambar 5 memperlihatkan bahwa sistem dapat menghasilkan prediksi untuk ketiga kelas yang diteliti. Keberadaan *bounding box* menunjukkan bahwa sistem tidak hanya memberikan label untuk keseluruhan citra, tetapi juga menentukan lokasi setiap objek yang dikenali. Fitur ini memungkinkan penggunaan citra yang memuat lebih dari satu tomat. Meskipun demikian, contoh visual tidak dapat digunakan sebagai pengganti evaluasi kuantitatif karena citra yang berhasil dideteksi belum mewakili seluruh variasi kondisi data. Kesimpulan mengenai kinerja tetap didasarkan pada hasil *precision*, *recall*, dan *mAP*.

Implementasi berbasis web menjawab tujuan penelitian mengenai penyediaan akses yang lebih praktis terhadap model. Pengguna tidak perlu menjalankan perintah Python atau berinteraksi langsung dengan lingkungan

pelatihan. Namun, istilah *real-time* perlu digunakan secara hati-hati karena penelitian belum melaporkan waktu inferensi, waktu respons server, jumlah permintaan simultan, atau pengujian pada aliran video. Berdasarkan implementasi yang tersedia, sistem lebih tepat dinyatakan mampu memberikan hasil deteksi secara langsung setelah pengguna mengunggah atau mengambil citra, bukan sebagai sistem *real-time* berkecepatan tertentu.

### 3.5 Pembahasan hasil penelitian

Nilai mAP50 sebesar 0,964 menunjukkan bahwa YOLOv8n efektif untuk mengenali keberadaan dan kelas objek tomat pada ambang IoU 0,50. Hasil tersebut mendukung temuan Wibowo et al. (2023) bahwa keluarga YOLO dapat digunakan untuk membedakan kondisi buah segar dan busuk. Penelitian ini memperluas kategori keluaran menjadi tiga kondisi operasional, yaitu Tomat Segar, Tomat Matang, dan Tomat Tidak Layak. Pengelompokan tersebut memungkinkan sistem membedakan kematangan dari ketidaklayakan, sehingga tomat yang telah matang tidak otomatis dianggap mengalami kerusakan.

Hasil penelitian ini juga sejalan dengan Rismayanti dan Rahmadewi (2025), yang menunjukkan kemampuan YOLOv8 dalam mengidentifikasi tingkat kematangan buah mangga. Penelitian tersebut memperoleh *precision* sebesar 81,1%, *recall* 81,5%, dan mAP sebesar 79,8%, sedangkan model pada penelitian ini menghasilkan *precision* keseluruhan 92,4%, *recall* 85,1%, dan mAP50 96,4%. Perbedaan angka tersebut tidak dapat digunakan untuk menyatakan bahwa satu model secara mutlak lebih unggul karena objek, jumlah data, definisi kelas, skema pembagian data, kualitas anotasi, dan konfigurasi evaluasinya berbeda. Perbandingan tersebut hanya menunjukkan bahwa YOLOv8 mempunyai potensi yang konsisten untuk tugas inspeksi visual pada komoditas pertanian.

Integrasi model ke aplikasi web memperluas hasil penelitian dari eksperimen model menjadi prototipe yang dapat digunakan melalui peramban. Hal ini melengkapi penelitian Erwanto et al. (2024), yang mengintegrasikan YOLO ke aplikasi Android untuk mendeteksi penyakit tanaman tomat melalui citra daun. Perbedaan utamanya terletak pada objek, keluaran, dan media implementasi. Penelitian ini menganalisis buah setelah panen, menghasilkan kelas kondisi yang berhubungan dengan sortasi, dan menggunakan antarmuka web yang tidak memerlukan instalasi aplikasi khusus.

Secara praktis, *recall* tinggi pada kelas Tomat Matang dan Tomat Segar menunjukkan bahwa model dapat membantu menemukan sebagian besar objek dari kedua kondisi tersebut. Akan tetapi, *precision* kelas Tomat Matang sebesar 76,5% dan *recall* kelas Tidak Layak sebesar 82,6% memperlihatkan risiko kesalahan yang masih perlu dipertimbangkan. Sistem belum dapat menggantikan pemeriksaan manusia pada keputusan yang berdampak langsung terhadap penerimaan atau penolakan produk. Pada tahap pengembangan saat ini, sistem lebih tepat ditempatkan sebagai alat bantu penyortiran awal yang memberikan rekomendasi visual kepada pengguna.

Ketidakseimbangan jumlah objek kemungkinan turut memengaruhi perbedaan kinerja antarkelas. Kelas Tomat Matang hanya memiliki 513 objek, hampir setengah dari jumlah objek Tidak Layak dan Tomat Segar. Walaupun *recall* kelas tersebut tinggi, *precision*-nya lebih rendah, yang menunjukkan bahwa penambahan data tidak hanya perlu diarahkan pada jumlah objek Tomat Matang, tetapi juga pada contoh batas antarkelas. Citra transisi antara segar dan matang, matang dan terlalu matang, serta rusak ringan dan tidak layak diperlukan agar model mempelajari perbedaan yang lebih halus.

Perbedaan mAP50 dan mAP50-95 juga mengindikasikan bahwa kualitas lokalisasi masih perlu ditingkatkan. Perbaikan dapat diarahkan pada pemeriksaan ulang konsistensi anotasi, penambahan citra dengan objek bertumpuk, penggunaan citra beresolusi lebih tinggi, serta evaluasi varian YOLOv8 yang memiliki kapasitas lebih besar. Akan tetapi, penggunaan model yang lebih besar akan meningkatkan kebutuhan komputasi dan berpotensi memperlambat respons aplikasi web. Pengembangan berikutnya perlu menyeimbangkan ketepatan lokalisasi dengan ukuran model dan waktu inferensi.

### 4. Kesimpulan

Penelitian ini berhasil mengembangkan model YOLOv8n untuk mendeteksi dan mengklasifikasikan kondisi tomat ke dalam tiga kelas, yaitu Tomat Segar, Tomat Matang, dan Tomat Tidak Layak. Model dilatih menggunakan 1.004 citra yang menghasilkan 2.427 objek beranotasi, terdiri atas 953 objek Tomat Segar, 513 objek Tomat Matang, dan 961 objek Tidak Layak. Hasil evaluasi menunjukkan bahwa model memperoleh

*precision* sebesar 0,924, *recall* sebesar 0,851, mAP50 sebesar 0,964, dan mAP50–95 sebesar 0,714. Nilai mAP50 yang tinggi memperlihatkan bahwa model mampu mengenali keberadaan dan kelas objek tomat dengan baik pada ambang IoU 0,50. Namun, selisih antara mAP50 dan mAP50–95 menunjukkan bahwa ketepatan penempatan *bounding box* masih perlu ditingkatkan ketika model dievaluasi menggunakan ambang IoU yang lebih ketat. Evaluasi per kelas menunjukkan bahwa Tomat Segar memperoleh *F1-score* tertinggi sebesar 86,7%, diikuti Tomat Matang sebesar 83,9% dan Tidak Layak sebesar 80,3%. Kelas Tomat Matang menghasilkan *recall* tertinggi sebesar 92,9%, tetapi memiliki *precision* sebesar 76,5%. Hasil tersebut menunjukkan bahwa sebagian besar tomat matang berhasil ditemukan, meskipun masih terdapat objek dari kelas lain yang diprediksi sebagai Tomat Matang. Sementara itu, *recall* kelas Tidak Layak sebesar 82,6% mengindikasikan bahwa sejumlah objek yang tidak layak masih belum dikenali secara tepat. Perbedaan kinerja antarkelas berkaitan dengan ketidakseimbangan jumlah objek, kemiripan warna antara kelas yang berdekatan, variasi tingkat kerusakan, kondisi pencahayaan, dan ketepatan anotasi. Model terbaik berhasil diintegrasikan ke dalam aplikasi berbasis web melalui arsitektur yang menghubungkan HTML, CSS, JavaScript, PHP Native, dan modul inferensi Python. Sistem dapat menerima citra dari perangkat atau kamera dan menampilkan hasil berupa *bounding box*, label kelas, serta nilai *confidence*. Dengan demikian, penelitian ini telah menjawab tujuan penerapan YOLOv8n, evaluasi kinerja model, dan pembangunan sistem deteksi berbasis web. Sistem berpotensi digunakan sebagai alat bantu penyortiran awal yang lebih objektif, tetapi belum dapat dinyatakan sebagai pengganti penuh pemeriksaan manusia. Pengembangan selanjutnya memerlukan penambahan data pada kondisi batas antarkelas, pengujian terhadap objek non-tomat, evaluasi pada lingkungan distribusi sebenarnya, pengukuran waktu inferensi, dan peningkatan ketepatan lokalisasi objek.

## Reference

1. Aherwadi, N., Mittal, U., Singla, J., Jhanjhi, N. Z., Yassine, A., & Hossain, M. S. (2022). Prediction of fruit maturity, quality, and its life using deep learning algorithms. *Electronics*, 11(24), 4100. <https://doi.org/10.3390/electronics11244100>
2. Baladraf, T. T. (2024). Analisis kerugian pascapanen melalui *Commodity System Assessment Method* pada komoditas tomat di Indonesia. *Prosiding Seminar Nasional Pembangunan dan Pendidikan Vokasi Pertanian*, 5(1), 387–396. <https://doi.org/10.47687/snppvp.v5i1.1121>
3. Bengio, Y., LeCun, Y., & Hinton, G. (2021). Deep learning for AI. *Communications of the ACM*, 64(7), 58–65. <https://doi.org/10.1145/3448250>
4. Collins, E. J., Bowyer, C., Tsouza, A., & Chopra, M. (2022). Tomatoes: An extensive review of the associated health impacts of tomatoes and factors that can affect their cultivation. *Biology*, 11(2), 239. <https://doi.org/10.3390/biology11020239>
5. Cong, X., Li, S., Chen, F., Liu, C., & Meng, Y. (2023). A review of YOLO object detection algorithms based on deep learning. *Frontiers in Computing and Intelligent Systems*, 4(2), 17–20. <https://doi.org/10.54097/fcis.v4i2.9730>
6. Erwanto, B., Pradana, A. I., & Hartanti, D. (2024). Pengembangan sistem deteksi penyakit tanaman tomat melalui citra daun dengan metode You Only Look Once berbasis Android. *G-Tech: Jurnal Teknologi Terapan*, 8(3), 1453–1463. <https://doi.org/10.33379/gtech.v8i3.4327>
7. Khan, A., Hassan, T., Shafay, M., Fahmy, I., Werghi, N., Seneviratne, L., & Hussain, I. (2023). Tomato maturity recognition with convolutional transformers. *Scientific Reports*, 13, 22885. <https://doi.org/10.1038/s41598-023-50129-w>
8. Kim, T., Lee, D. H., Kim, K. C., Choi, T., & Yu, J. M. (2023). Tomato maturity estimation using deep neural network. *Applied Sciences*, 13(1), 412. <https://doi.org/10.3390/app13010412>
9. Li, P., Zheng, J., Li, P., Long, H., Li, M., & Gao, L. (2023). Tomato maturity detection and counting model based on MHSA-YOLOv8. *Sensors*, 23(15), 6701. <https://doi.org/10.3390/s23156701>
10. Nahak, P., Pratihar, D. K., & Deb, A. K. (2025). Tomato maturity stage prediction based on vision transformer and deep convolution neural networks. *International Journal of Hybrid Intelligent Systems*, 21(1), 61–78. <https://doi.org/10.3233/HIS-240021>
11. Rico, A. L. J. (2025). AI-driven computer vision for automated postharvest quality assessment of Diamante Max tomatoes using YOLOv5. *International Journal of Advanced Science and Intelligent Systems*, 47–57. <https://doi.org/10.29284/18tref64>
12. Rismayanti, A., & Rahmadewi, R. (2025). Deteksi dan klasifikasi tingkat kematangan buah mangga harum manis menggunakan You Only Look Once (YOLO) V8. *JATI: Jurnal Mahasiswa Teknik Informatika*, 9(3), 3645–3654. <https://doi.org/10.36040/jati.v9i3.13320>
13. Shobaki, W. A., & Milanova, M. (2025). A comparative study of YOLO, SSD, Faster R-CNN, and more for optimized eye-gaze writing. *Sci*, 7(2), 47. <https://doi.org/10.3390/sci7020047>
14. Stasenko, N., Shukhratov, I., Savinov, M., Shadrin, D., & Somov, A. (2023). Deep learning in precision agriculture: Artificially generated VNIR images segmentation for early postharvest decay prediction in apples. *Entropy*, 25(7), 987. <https://doi.org/10.3390/e25070987>
15. Wang, S., Xiang, J., Chen, D., & Zhang, C. (2024). A method for detecting tomato maturity based on deep learning. *Applied Sciences*, 14(23), 11111. <https://doi.org/10.3390/app142311111>
16. Wibowo, A., Lusiana, L., & Dewi, T. K. (2023). Implementasi algoritma deep learning You Only Look Once (YOLOv5) untuk deteksi buah segar dan busuk. *Paspalum: Jurnal Ilmiah Pertanian*, 11(1), 123–130. <https://doi.org/10.35138/paspalum.v11i1.489>
17. Xiao, B., Nguyen, M., & Yan, W. Q. (2024). Fruit ripeness identification using YOLOv8 model. *Multimedia Tools and Applications*, 83, 28039–28056. <https://doi.org/10.1007/s11042-023-16570-9>
18. Zheng, S., Jia, X., He, M., Zheng, Z., Lin, T., & Weng, W. (2024). Tomato recognition method based on the YOLOv8-Tomato model in complex greenhouse environments. *Agronomy*, 14(8), 1764. <https://doi.org/10.3390/agronomy14081764>

DOI: <https://doi.org/10.69693/ijmst.v4i3.13916>

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)