Identifikasi Kematangan Buah Apel Menggunakan Algoritma YOLO

I Gede Liyang Anugrah Oktapian[a1], Gst. Ayu Vida Mastrika Giri[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Udayana, Indonesia
Jln. Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, 08261, Bali, Indonesia
[1]oktapian.2308561042@student.unud.ac.id
[2]vida@unud.ac.id

Abstract

The classification of fruit ripeness plays a vital role in the agricultural product processing industry. Manual sorting based on visual perception is often subjective and inconsistent. This research proposes an automatic detection and classification system for apple ripeness levels, namely unripe, half ripe, and ripe, using the YOLOv8n object detection algorithm. A dataset of 1,800 apple images was collected and annotated using YOLO format, then trained on a lightweight YOLOv8n model for 30 epochs. The evaluation results showed high performance, with mAP@0.5 of 0.975 and mAP@0.5:0.95 of 0.959. Class-wise, the model achieved F1-scores of 0.94 for unripe, 0.93 for half ripe, and 0.88 for ripe apples. The confusion matrix indicated that most misclassifications occurred between the ripe and half ripe classes, suggesting feature similarity. The model demonstrated accurate and efficient detection, making it suitable for real-time fruit sorting applications. Future work may explore data augmentation, deeper YOLO variants, or integration with IoT devices for deployment in agricultural environments.

Keywords: *Apple Ripeness Detection, YOLOv8, Object Detection, Fruit Classification, Deep Learning, Image Processing*

1. Pendahuluan

Dengan berkembangnya ilmu pengetahuan dan teknologi, industri pengolahan hasil pertanian dan perkebunan ikut berkembang pesat. Dalam pengolahan ini, pemilihan produk berdasarkan kualitasnya, seperti tingkat kematangan buah, menjadi tahap krusial. Umumnya, proses ini masih sangat bergantung pada penilaian manusia terhadap tampilan visual, khususnya warna permukaan buah. Salah satu komoditas hortikultura yang banyak digemari masyarakat adalah buah apel (Malus domestica). Apel memiliki cita rasa yang bervariasi serta kandungan nutrisi yang tinggi, seperti lemak baik, karbohidrat, protein, vitamin C, vitamin A, vitamin B1, B2, dan berbagai zat gizi lainnya [1]. Berdasarkan data dari Badan Pusat Statistik (BPS), pada tahun 2023 produksi apel di Indonesia mencapai 392.563 ton, dengan 392.173 ton di antaranya berasal dari Jawa Timur [2]. Data tersebut mengindikasikan bahwa apel sangat populer dan banyak dikonsumsi, baik segar maupun diolah menjadi berbagai produk seperti manisan, keripik, dodol, dan minuman [1].

Melihat pentingnya kualitas dan ketepatan dalam proses pemilihan hasil pertanian seperti buah apel, maka diperlukan sebuah sistem yang mampu melakukan identifikasi tingkat kematangan buah secara otomatis dan akurat. Salah satu metode yang dapat digunakan untuk tujuan tersebut adalah algoritma *You Only Look Once* (YOLO), yang dikenal karena kecepatan pemrosesan dan akurasinya yang tinggi [3]. Algoritma YOLO mendeteksi objek dengan cara membagi citra menjadi beberapa grid, di mana setiap grid bertugas memprediksi keberadaan objek melalui *bounding box*, skor objektif, dan skor kelas [4].
Proses deteksi pada algoritma YOLO dilakukan melalui tiga langkah utama. Pertama, citra dibagi menjadi *grid* berukuran s × s, dan setiap *grid* akan memprediksi *bounding box* beserta nilai *confidence* yang menunjukkan keyakinan terhadap keberadaan objek serta ketepatan prediksinya berdasarkan nilai *Intersection over Union* (IOU). Nilai IOU yang lebih tinggi

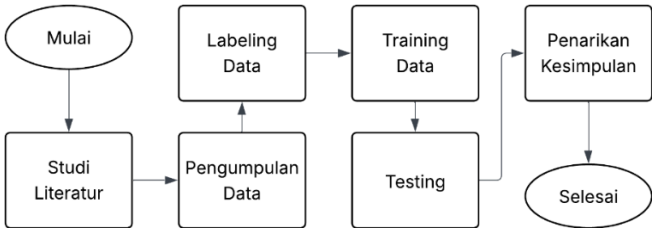
p-ISSN: 2986-3929
e-ISSN: 3032-1948

menunjukkan bahwa prediksi semakin akurat. Kedua, setiap *bounding box* memiliki lima parameter utama, yaitu (x, y) sebagai koordinat pusat, (w, h) sebagai dimensi lebar dan tinggi, serta c sebagai nilai *confidence*. Ketiga, jika suatu *grid* mendeteksi keberadaan objek, maka akan dihitung probabilitas kelas objek tersebut. Nilai probabilitas ini kemudian dikalikan dengan *confidence bounding box* untuk menghasilkan nilai *confidence* spesifik terhadap masing-masing kelas objek [4].

Beberapa penelitian sebelumnya membuktikan efektivitas algoritma YOLO dalam klasifikasi kematangan buah. Agustina dan Sukron [5] mengembangkan sistem berbasis Android untuk mendeteksi kematangan pepaya dengan akurasi 93%, *precision* rata-rata 94%, dan *recall* 93%. Sementara itu, Rahman dkk., [6] menggunakan YOLOv8 untuk mendeteksi tingkat kematangan buah manggis secara *real-time* melalui platform web, dengan *precision* 0,997, *recall* 1, dan mAP50-95 sebesar 0,972. Hasil kedua studi tersebut menunjukkan bahwa YOLO dapat menjadi solusi akurat dan efisien dalam mengotomatisasi proses seleksi buah, serta memiliki potensi besar untuk diterapkan pada identifikasi kematangan buah apel.

2. Metode Penelitian

Penelitian ini dilaksanakan menggunakan metode eksperimental dengan objek penelitian berupa data buah apel yang diperoleh dari situs Kaggle [7]. Rangkaian tahapan dalam metodologi penelitian disajikan dalam bentuk diagram alur yang ditampilkan pada Gambar 1.



Gambar 1. Diagram Alur Penelitian

2.1. Studi Literatur

Pada tahap ini dilakukan penelusuran dan telaah terhadap berbagai literatur yang mendukung penelitian, meliputi sumber dari internet, jurnal ilmiah, penelitian sebelumnya, serta modul-modul yang berkaitan dengan penerapan algoritma YOLO untuk sistem deteksi tingkat kematangan berdasarkan dataset citra buah apel, termasuk kajian dari penelitian sejenis.

2.2. Pengumpulan Data

Tahap ini merupakan bagian awal yang harus dipersiapkan sebelum pelaksanaan penelitian. Kegiatan yang dilakukan meliputi pengumpulan dataset citra buah apel sebagai data sekunder yang diunduh dari situs Kaggle [7]. Citra yang terkumpul selanjutnya diklasifikasikan ke dalam tiga kategori tingkat kematangan, yaitu mentah, setengah matang, dan matang (Gambar 2). Rincian pembagian data dapat dilihat pada Tabel 1.

Gambar 2. Citra Apel Mentah, Setengah Matang, dan Matang

Tabel 1. Rincian Pembagian Data

Kategori	Jumlah Total	Training (66.7%)	Validation (16.7%)	Testing (16.7%)
Mentah	600	400	100	100
Setengah Matang	600	400	100	100
Matang	600	400	100	100
Total	1800	1200	300	300

2.3. *Labeling Data*

Tahap *labeling* dilakukan untuk memberikan anotasi pada setiap citra dalam dataset agar dapat dikenali oleh model saat pelatihan. Penelitian ini menggunakan format YOLO, di mana setiap *file* label (.txt) berisi lima elemen yaitu `class_id`, `x_center`, `y_center`, `width`, dan `height`, yang dinormalisasi terhadap ukuran gambar. Contoh isi file seperti ` 0 0.5 0.5 1.0 1.0` menunjukkan kelas dan koordinat *bounding box*. Kelas terdiri dari mentah (0), setengah matang (1), dan matang (2), yang daftarnya disimpan dalam folder *labels*. Format ini mendukung proses pelatihan YOLO dalam membedakan tingkat kematangan buah apel secara otomatis.

2.4. *Training Data*

Pada tahap ini dilakukan pelatihan model deteksi objek menggunakan algoritma YOLOv8. Model yang digunakan adalah YOLOv8n (nano) yang merupakan versi ringan dan cepat, cocok untuk pelatihan menggunakan perangkat dengan sumber daya terbatas seperti CPU. *Dataset* yang digunakan memiliki resolusi masing-masing 100×100 piksel, serta telah dilabeli dalam format YOLO (.txt). Pelatihan dilakukan menggunakan *library* Ultralytics dengan parameter yang ditampilkan pada tabel 2.

Tabel 2. Parameter *Training Data*

Parameter	Nilai
Model	yolov8n.pt
Epoch	30
Ukuran Gambar	100 × 100 piksel
Batch Size	16
Dataset Format	YOLO .txt
Jumlah Data	1200 (train), 300 (val)
Framework	Ultralytics (YOLOv8)
Perangkat	CPU

2.5. *Testing*

Tahap testing dilakukan setelah proses pelatihan model selesai, dengan tujuan untuk mengukur performa model terhadap data yang belum pernah dilihat sebelumnya. *Dataset* pengujian terdiri dari 300 gambar yang tidak digunakan dalam proses pelatihan maupun validasi, sehingga hasil pada tahap ini merepresentasikan kemampuan model dalam melakukan generalisasi terhadap data baru. Proses testing dilakukan dengan memuat model terbaik hasil pelatihan (`best.pt`) dan menerapkannya pada gambar-gambar dalam folder pengujian menggunakan pustaka Ultralytics YOLOv8. Model akan secara otomatis mendeteksi objek apel pada gambar, menghasilkan *bounding box*, label kelas (mentah, setengah matang, matang), serta nilai *confidence* untuk setiap prediksi.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Untuk mengevaluasi hasil deteksi, dilakukan perhitungan metrik evaluasi menggunakan fungsi `model.val()` yang tersedia dalam pustaka Ultralytics. Evaluasi ini mencakup penghitungan metrik kuantitatif seperti precision, recall, F1-score, mAP\@0.5, mAP\@0.5:0.95, serta confusion matrix, yang memberikan gambaran menyeluruh terhadap akurasi dan konsistensi prediksi model pada data uji.

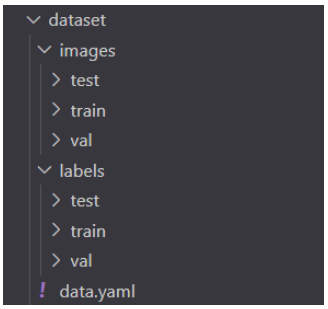
2.6. Penarikan Kesimpulan

Penarikan kesimpulan dalam penelitian ini dilakukan setelah seluruh tahapan utama, mulai dari pengumpulan dan pelabelan data, pelatihan model, hingga pengujian selesai dilakukan. Proses ini bertujuan untuk mengevaluasi sejauh mana model deteksi objek YOLOv8n mampu mengidentifikasi tingkat kematangan buah apel secara akurat dan efisien.

3. Hasil dan Diskusi

3.1. *Preprocessing Data*

Dataset citra buah apel yang digunakan dalam penelitian ini diperoleh dari situs Kaggle [7]. Dataset ini kemudian disusun ulang mengikuti format standar YOLOv8, agar dapat digunakan secara langsung dalam proses *train*, dan *testing*.



Gambar 3. Format Dataset Yolov8

Struktur dataset terdiri dari dua direktori utama, yaitu *images* dan *labels*, di mana direktori images digunakan untuk menyimpan seluruh gambar dataset yang dibagi ke dalam subfolder train, val, dan test, sedangkan direktori labels digunakan untuk menyimpan *file* anotasi berformat .txt yang memuat informasi *bounding box* dan *class ID* dengan struktur subfolder yang sama seperti pada direktori images. Setiap *file* gambar memiliki pasangan *file* label berformat .txt dengan nama yang sama. *File* label ini berisi informasi mengenai *bounding box* dan *class ID* dalam format YOLO, yang dituliskan dalam satu baris dengan lima elemen. Contoh isi file: ` 0 0.5 0.5 1.0 1.0` , adapun penjelasan dari masing-masing elemen adalah sebagai berikut:

- 0 : *Class ID*, dalam hal ini 0 menunjukkan kelas mentah.
- 0.5 : Posisi x tengah *bounding box*, dinormalisasi terhadap lebar gambar
- 0.5 : Posisi y tengah *bounding box*, dinormalisasi terhadap tinggi gambar
- 1.0 : Lebar *bounding box* (100% dari lebar gambar)
- 1.0 : Tinggi *bounding box* (100% dari tinggi gambar)

Selain folder gambar dan label, terdapat pula file konfigurasi bernama data.yaml yang berfungsi untuk mendefinisikan struktur dataset dan daftar kelas objek. File ini digunakan oleh YOLOv8 untuk membaca lokasi data serta nama kelas yang dikenali oleh model.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

```yaml
path: dataset               # Direktori utama tempat folder images dan labels berada

train: images/train         # Lokasi folder gambar untuk data pelatihan
val: images/val             # Lokasi folder gambar untuk data validasi
test: images/test           # Lokasi folder gambar untuk data pengujian

names:                      # Daftar nama kelas yang dikenali oleh model
  0: Mentah                 # class_id 0 = apel mentah
  1: Setengah Matang        # class_id 1 = apel setengah matang
  2: Matang                 # class_id 2 = apel matang
```

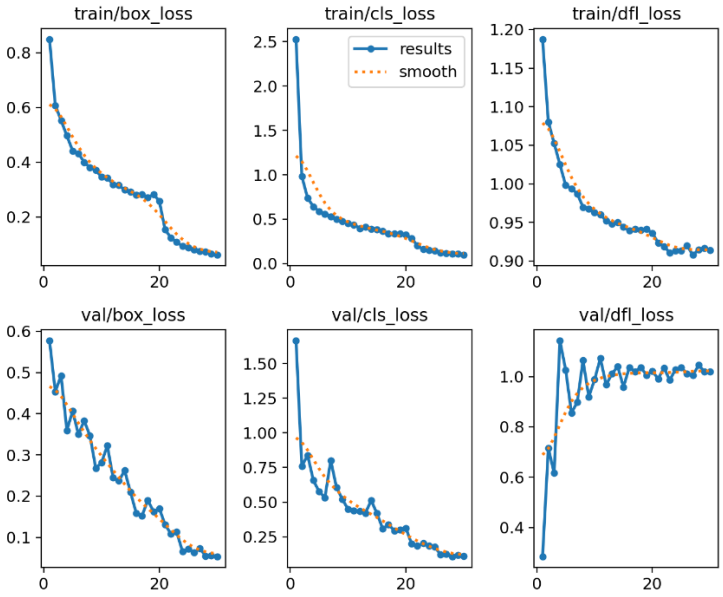
Gambar 4. Format data.yaml

3.2. *Training* Model

Proses *train* model YOLOv8n dilakukan selama 30 epoch menggunakan dataset citra buah apel beresolusi 100×100 piksel. Hasil pelatihan menunjukkan bahwa model mengalami peningkatan performa yang signifikan seiring bertambahnya *epoch*.

a. *Loss Function*

Nilai *box loss* pada data pelatihan (train/box_loss) mengalami penurunan dari sekitar 0,8 menjadi sekitar 0,05, sementara pada data validasi (val/box_loss) turun dari sekitar 0,6 menjadi sekitar 0,1. Hal ini menunjukkan bahwa model berhasil belajar memprediksi posisi objek secara akurat melalui *bounding box*. Selanjutnya, nilai *classification loss* (train/cls_loss dan val/cls_loss) juga menurun secara signifikan, dari 2,5 dan 1,5 menjadi kurang dari 0,25, yang menandakan peningkatan dalam pengenalan kelas kematangan buah. Untuk *distribution focal loss* (dfl_loss), tren penurunan terlihat jelas pada data pelatihan (train/dfl_loss) yang turun dari sekitar 1,2 menjadi 0,91. Sedangkan pada data validasi (val/dfl_loss), terjadi fluktuasi awal dari sekitar 0,4 hingga 1,0, namun nilai tersebut cenderung stabil di angka sekitar 1,0 pada akhir pelatihan. Penurunan seluruh komponen *loss* ini secara konsisten memperlihatkan bahwa model semakin mampu melakukan prediksi yang akurat seiring bertambahnya iterasi pelatihan.

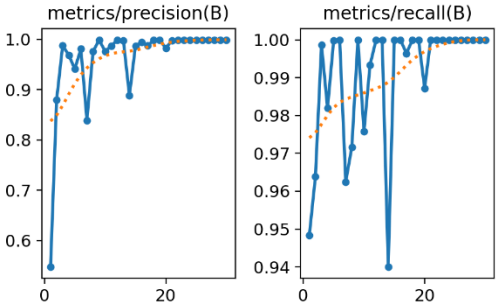


Gambar 5. Grafik *Loss Function*

p-ISSN: 2986-3929
e-ISSN: 3032-1948

b. Evaluasi Performa

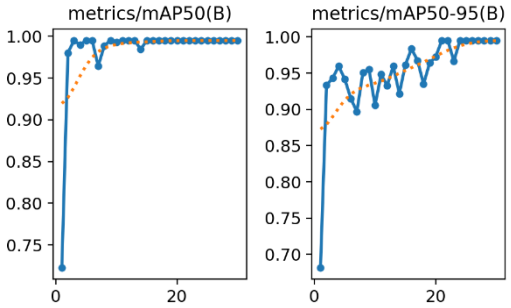
Evaluasi performa model dilakukan menggunakan metrik *precision* dan *recall* terhadap data validasi. Nilai precision meningkat tajam dari sekitar 0,6 menjadi hampir 1,0, yang menunjukkan bahwa sebagian besar prediksi model tepat sasaran. Sementara itu, nilai *recall* juga mengalami peningkatan dari sekitar 0,95 menjadi 1,0, mengindikasikan bahwa hampir seluruh objek yang ada dalam citra berhasil terdeteksi dengan benar oleh model. Fluktuasi ringan pada grafik awal pelatihan adalah hal yang umum, tetapi kestabilan di akhir menunjukkan bahwa model telah mencapai kinerja optimal.



Gambar 6. Grafik Evaluasi Performa

c. *Mean Average Precision*

Nilai *mean Average Precision* (mAP) digunakan untuk mengevaluasi performa keseluruhan model dalam berbagai tingkat ketepatan prediksi berdasarkan nilai *Intersection over Union* (IoU). Hasil menunjukkan bahwa nilai mAP50 berhasil mencapai angka maksimum 1,0, yang menunjukkan bahwa model dapat mendeteksi objek dengan akurasi tinggi pada ambang tumpang tindih minimal 50%. Sementara itu, nilai mAP50-95 meningkat secara bertahap dari sekitar 0,7 menjadi mendekati 0,99. Hal ini mengindikasikan bahwa model tidak hanya unggul dalam deteksi kasar, tetapi juga mampu mempertahankan presisi dalam berbagai variasi IoU, sehingga menunjukkan generalisasi yang sangat baik terhadap data validasi.



Gambar 7. *Mean Average Precision*

3.3. *Testing* Model

Testing model dilakukan untuk mengukur kinerja akhir YOLOv8n dalam mengklasifikasikan tingkat kematangan apel pada data uji, menggunakan metrik *precision*, *recall*, *F1-score*, serta mAP@0.5 dan mAP@0.5:0.95 guna menilai kemampuan generalisasi model terhadap data baru.

a. Evaluasi Per Kelas

Model menunjukkan performa yang sangat baik dalam mengenali kelas mentah dan setengah matang, dengan nilai *F1-score* masing-masing 0.94 dan 0.93. Meskipun kelas

p-ISSN: 2986-3929
e-ISSN: 3032-1948

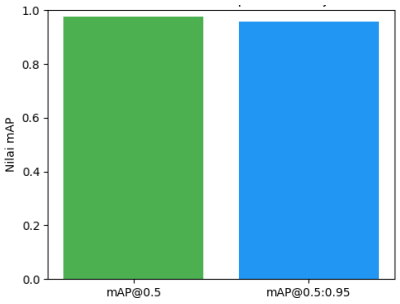
matang memiliki precision sempurna (1.00), *recall*-nya masih relatif rendah (0.79), yang mengindikasikan bahwa sebagian gambar matang tidak berhasil dikenali dengan baik oleh model. Hal ini kemungkinan disebabkan oleh kemiripan visual antar kelas atau kurangnya variasi data pada kelas matang, sehingga mempersulit model dalam membedakan karakteristiknya secara akurat.

Tabel 3. Evaluasi Per Kelas

Kelas	Precision	Recall	F1-Score
Mentah	1.00	0.88	0.94
Setengah Matang	0.88	1.00	0.93
Matang	1.00	0.79	0.88

b. Grafik mAP pada Data Uji

Evaluasi performa deteksi juga dilakukan menggunakan metrik *mean Average Precision* (mAP), yang menjadi indikator utama dalam menilai akurasi prediksi *bounding box* dan klasifikasi objek. Grafik berikut menunjukkan hasil evaluasi model berdasarkan dua skenario IoU:



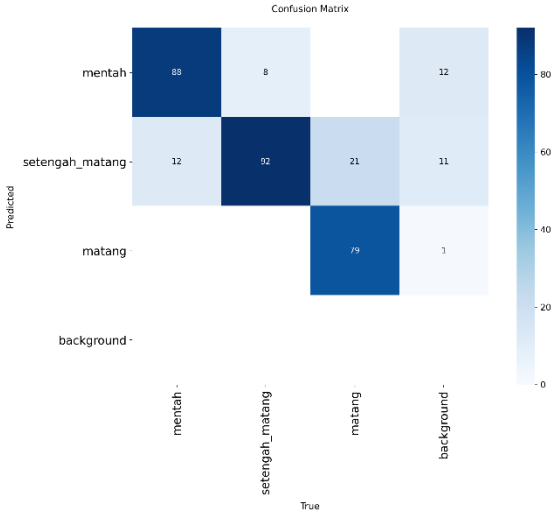
Gambar 8. Grafik mAP Model pada Data Uji

Berdasarkan hasil pengujian, model memperoleh nilai mAP@0.5 sebesar 0.975, dan mAP@0.5:0.95 sebesar 0.959. Nilai mAP yang sangat tinggi ini menunjukkan bahwa model mampu melakukan deteksi dan klasifikasi tingkat kematangan buah apel dengan presisi dan akurasi yang sangat baik, baik pada skenario IoU minimum (0.5) maupun dalam rentang ketelitian yang lebih ketat (0.5–0.95). Hal ini menandakan bahwa model memiliki kemampuan generalisasi yang kuat terhadap data baru.

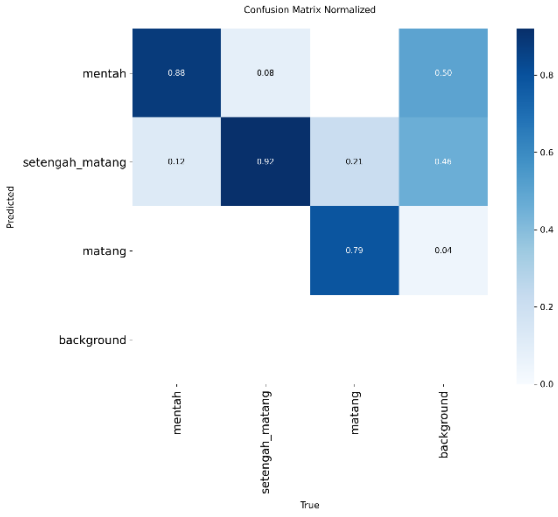
c. *Confusion Matrix*

Confusion matrix disajikan dalam dua bentuk, yaitu belum dan telah ternormalisasi, untuk memberikan gambaran menyeluruh mengenai performa klasifikasi model.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 9. *Confusion Matrix*



Gambar 10. *Confusion Matrix Normalized*

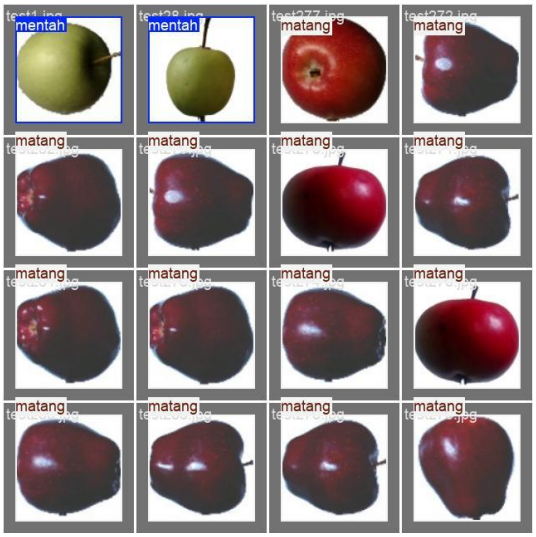
Berdasarkan *confusion matrix* (Gambar 6), model berhasil mengenali sebagian besar gambar secara tepat, dengan jumlah prediksi benar tertinggi pada kelas setengah matang (92 citra), disusul oleh kelas mentah (88 citra), dan matang (79 citra). Kesalahan klasifikasi terlihat pada prediksi silang antar kelas, misalnya 12 citra matang diprediksi sebagai latar belakang dan 21 citra setengah matang diklasifikasikan sebagai matang. Pada *confusion matrix* ternormalisasi (Gambar 7), terlihat bahwa kelas setengah matang memiliki akurasi tertinggi dengan nilai 0,92. Kelas mentah memiliki proporsi benar sebesar 88%, dengan sebagian kecil salah diklasifikasikan ke kelas lain. Sementara itu, kelas matang memiliki recall 0,79, menunjukkan bahwa 21% citra gagal dikenali sebagai matang.

Hal ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan objek matang dari dua kelas lainnya, kemungkinan karena kemiripan fitur visual atau kurangnya variasi data pada kelas tersebut. Secara keseluruhan, confusion matrix memberikan gambaran bahwa meskipun model menunjukkan performa tinggi, masih terdapat ruang untuk perbaikan, khususnya dalam mendeteksi kelas matang. Perbaikan yang dapat dilakukan antara lain melalui penambahan variasi data latih, augmentasi citra, atau *fine-tuning* lebih lanjut pada model.

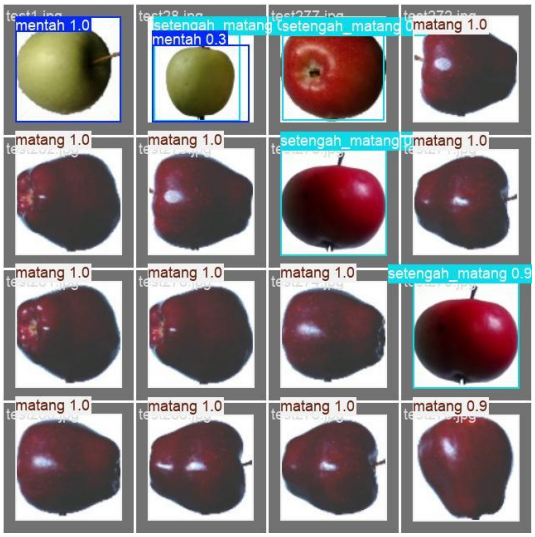
d. Hasil Deteksi

Gambar berikut merupakan contoh perbandingan antara label asli (*ground truth*) dan hasil deteksi model pada citra uji. Model mampu mengenali sebagian besar apel dengan benar, terutama kelas mentah dan setengah matang yang dideteksi dengan confidence tinggi. Namun, terdapat beberapa kesalahan klasifikasi, seperti apel matang yang terdeteksi sebagai setengah matang, sesuai dengan evaluasi sebelumnya bahwa *recall* kelas matang lebih rendah.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 11. Label Asli



Gambar 12. Hasil Prediksi Model

4. Kesimpulan

Penelitian ini berhasil mengimplementasikan algoritma YOLOv8n untuk mendeteksi dan mengklasifikasikan tingkat kematangan buah apel ke dalam tiga kelas yaitu mentah, setengah matang, dan matang. Berdasarkan hasil pelatihan dan evaluasi, model menunjukkan performa yang sangat baik, dengan nilai mAP@0.5 sebesar 0.975 dan mAP@0.5:0.95 sebesar 0.959. Evaluasi per kelas juga menunjukkan nilai F1-score yang tinggi pada kelas mentah (0.94) dan setengah matang (0.93), meskipun *recall* pada kelas matang masih tergolong rendah (0.79), yang mengindikasikan adanya tantangan dalam mengenali objek matang secara konsisten.

Visualisasi *confusion matrix* memperkuat temuan tersebut, di mana sebagian gambar matang diklasifikasikan sebagai kelas lain. Hal ini kemungkinan disebabkan oleh kemiripan visual antar kelas atau kurangnya variasi data latih pada kelas matang. Secara keseluruhan, model mampu melakukan deteksi dan klasifikasi secara cepat dan akurat, sehingga dapat menjadi solusi potensial dalam sistem seleksi buah otomatis berbasis citra. Untuk pengembangan selanjutnya, penelitian ini dapat ditingkatkan dengan menambahkan variasi data citra, menerapkan augmentasi yang lebih kompleks, serta mengeksplorasi arsitektur YOLO yang lebih besar atau metode ensemble untuk meningkatkan akurasi, khususnya pada kelas yang masih kurang optimal. Selain itu, penerapan sistem ini pada perangkat bergerak atau berbasis IoT dapat menjadi arah penelitian lanjutan yang menarik dan aplikatif.

Daftar Pustaka

[1] F. Indra Pratama, A. P. Wijaya, H. Pratiwi, dan A. Budianita, “Klasifikasi Kematangan Buah Apel Berdasarkan Warna Dan Tekstur Menggunakan Algoritma K-Nearest Neighbor,” *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, vol. 5, no. 1, hlm. 11–18, Mei 2023, doi: 10.46772/intech.v5i1.1119.

[2] Badan Pusat Statistik, “Produksi Tanaman Buah-buahan - Tabel Statistik - Badan Pusat Statistik Indonesia.” Diakses: 3 Juli 2025. [Daring]. Tersedia pada: https://www.bps.go.id/id/statistics-table/2/NjIjMg==/produksi-tanaman-buah-buahan.html

[3] J. Zophie dan H. H. Triharminto, “Implemetasi Algoritma You Only Look Once (YOLO) menggunakan Web Camera untuk Mendeteksi Objek Statis dan Dinamis,” *TNI Angkatan Udara Triwulan Pertama*, vol. 1, no. 1, hlm. 98–109, Jan 2023, doi: 10.62828/jpb.v1i1.50.

[4] M. Sarosa dan N. Muna, “Implementasi Algoritma You Only Look Once (YOLO) untuk Deteksi Korban Bencana Alam,” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 8, no. 4, hlm. 787–792, Jul 2021, doi: 10.25126/jtiik.2021844407.

[5] A. Feri dan M. Sukron, “Deteksi Kematangan Buah Pepaya Menggunakan Algoritma YOLO Berbasis Android,” *Jurnal Ilmiah Infokam*, vol. 18, no. 2, hlm. 70–78, Des 2022, doi: 10.53845/infokam.v18i2.320.

[6] R. Akyas Hifdzi Rahman, A. Adi Sunarto, dan Asriyanik, “PENERAPAN YOU ONLY LOOK ONCE (YOLO) V8 UNTUK DETEKSI TINGKAT KEMATANGAN BUAH MANGGIS,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 5, hlm. 10566–10571, Sep 2024, doi: 10.36040/jati.v8i5.10979.

[7] M. Oltean, “Fruits-360 Dataset,” Kaggle. Diakses: 1 Juli 2025. [Daring]. Tersedia pada: https://www.kaggle.com/datasets/moltean/fruits?select=fruits-360_100x100