

**PEMANFAATAN YOU ONLY LOOK ONCE (YOLO11)
UNTUK DETEKSI TINGKAT KEMATANGAN
BUAH TOMAT**



Disusun Oleh:

N a m a : Lutfi Alfian
NIM : 19523042

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2025**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PEMANFAATAN YOU ONLY LOOK ONCE (YOLO11)

UNTUK DETEKSI TINGKAT KEMATANGAN

BUAH TOMAT

TUGAS AKHIR

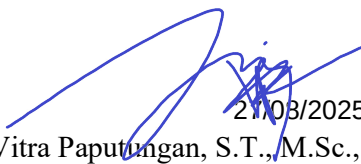


Disusun Oleh:

N a m a : Lutfi Alfian
NIM : 19523042

Yogyakarta, 15 Juli 2025

Pembimbing,


21/03/2025
(Irving Vitra Paputungan, S.T., M.Sc., Ph.D.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PEMANFAATAN YOU ONLY LOOK ONCE (YOLO11) UNTUK DETEKSI TINGKAT KEMATANGAN BUAH TOMAT

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 6 Agustus 2025

Tim Penguji

Irving Vitra Paputungan, S.T., M.Sc., Ph.D.

27/08/2025

Anggota 1

Dr. Raden Teduh Dirgahayu, S.T., M.Sc.

Anggota 2

Erika Ramadhani, S.T., M.Eng.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Lutfi Alfian

NIM : 19523042

Tugas akhir dengan judul:

PEMANFAATAN YOU ONLY LOOK ONCE (YOLO11) UNTUK DETEKSI TINGKAT KEMATANGAN BUAH TOMAT

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 15 Juli 2025



(Lutfi Alfian)

HALAMAN PERSEMBAHAN

Dengan memanjatkan puji syukur kehadiran Allah SWT atas rahmat dan karunia-Nya berupa kesehatan, kemudahan dan kelancaran sehingga penulis dapat menyelesaikan tugas akhir ini. Penulis mempersembahkan dengan segenap rasa penuh cinta dan sayang tak terhingga kepada orang tua serta kakak yang tak pernah berhenti menjadi sumber doa, semangat, motivasi dan inspirasi. Semoga tugas akhir ini dapat bermanfaat bagi penulis maupun pembaca.

HALAMAN MOTO

“Sesungguhnya, bersama kesulitan ada kemudahan.”

- QS. Al-Insyirah (6)

“Allah SWT tidak akan mengubah keadaan suatu kaum sampai mereka mengubah keadaan diri mereka sendiri.”

- QS. Al-Ra'd (11)

“Sampai kamu bahagia dengan dirimu sendiri, kamu tidak akan pernah bahagia dengan apa yang kamu miliki.”

- Zig Ziglar

“Kau tak menulis hidupmu dengan kata-kata. Kau menulisnya dengan tindakan. Apa yang kau pikirkan tidaklah penting. Yang penting hanyalah apa yang kau lakukan.”

- Patrick Ness

“Tidak benar kalau orang berhenti mengejar mimpi karena mereka bertambah tua, mereka bertambah tua karena mereka berhenti mengejar mimpi.”

- Gabriel García Márquez

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur senantiasa penulis panjatkan ke hadirat Allah SWT yang dengan rahmat, taufik, serta inayah-Nya telah memberikan kekuatan dan kemudahan kepada penulis sehingga dapat merampungkan penyusunan tugas akhir ini dengan baik. Tugas akhir yang berjudul “*Pemanfaatan You Only Look Once (YOLO11) Untuk Deteksi Tingkat Kematangan Buah Tomat*” ini diajukan sebagai salah satu syarat untuk menyelesaikan program sarjana Informatika di Universitas Islam Indonesia. Proses penyelesaian tugas akhir yang panjang dan penuh tantangan ini terbukti menjadi sebuah perjalanan akademis yang memberikan berbagai pembelajaran, pengalaman serta mengasah kreativitas dan berpikir ilmiah.

Penyelesaian tugas akhir ini tidak lepas dari dukungan, bimbingan, serta bantuan dari berbagai pihak yang telah memberikan arahan dan motivasi kepada penulis. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menghanturkan apresiasi dan rasa terima kasih yang sebesar-besarnya kepada:

1. **Allah SWT**, atas segala limpahan nikmat dan pertolongan-Nya dalam setiap langkah penulis, yang telah memberikan kekuatan, kesehatan dan kelancaran dalam penyusunan tugas akhir ini.
2. **Kedua Orang Tua dan Kakak Tersayang**, atas doa, cinta, dukungan moral dan material serta motivasi yang pernah henti.
3. **Bapak Dr. Raden Teguh Dirgahayu, S.T., M.Sc.**, selaku Ketua Jurusan Informatika – Fakultas Teknologi Industri Universitas Islam Indonesia.
4. **Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.**, selaku Ketua Program Studi Informatika – Program Sarjana Universitas Islam Indonesia.
5. **Bapak Irving Vitra Paputungan, S.T., M.Sc., Ph.D.**, selaku dosen pembimbing yang telah dengan sabar membimbing, memberikan waktu, ilmu, arahan dan masukan berharga selama proses penelitian hingga penulisan laporan ini.
6. **Ibu Sri Mulyati, S.KOM., M.KOM.**, selaku dosen pembimbing akademik yang telah memberikan arahan, bimbingan dan dukungan selama masa studi penulis.
7. **Seluruh dosen di Program Studi Informatika**, yang telah memberikan bekal ilmu pengetahuan, wawasan dan motivasi selama masa perkuliahan.
8. **Seluruh Staf Administrasi dan Tenaga Pendukung Informatika**, yang selalu siap membantu dalam hal teknis dan administrasi selama masa perkuliahan.

9. **Sahabat-Sahabat terbaik**, yang selalu hadir memberikan dukungan moral, semangat dan kebersamaan dalam perjalanan akademik ini.
10. **Seluruh Pihak yang Tidak Dapat Disebutkan Satu per Satu**, yang telah membantu baik secara langsung maupun tidak langsung pada penyelesaian tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan demi perbaikan dimasa yang akan datang. Semoga tugas akhir ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan, terutama pada bidang informatika serta memberikan manfaat bagi para pembaca.

Yogyakarta, 15 Juli 2025



(Lutfi Alfian)

SARI

Proses penyortiran tomat secara manual yang bersifat subjektif dan memakan biaya menjadi kendala dalam rantai pasok tomat dari petani hingga industri. Permasalahan ini semakin kompleks akibat kesenjangan antara ketersediaan produksi dengan permintaan konsumsi. Penelitian ini memanfaatkan model *You Only Look Once* (YOLO) untuk mendeteksi tingkat kematangan tomat secara *real-time*. Model digunakan untuk mengidentifikasi sekaligus mengklasifikasikan tingkat kematangan tomat ke dalam tiga kategori kematangan yaitu mentah, setengah matang dan matang sehingga dapat mengoptimalkan dan meningkatkan efisiensi seluruh alur kerja pascapanen. Metodologi penelitian meliputi pengumpulan 1.088 citra tomat, anotasi dan augmentasi hingga menghasilkan 2.610 citra untuk pelatihan. Empat varian model YOLO11 (n, s, m, l) dilatih dan dievaluasi performanya menggunakan metrik *precision*, *recall* dan *mAP50*. Hasil evaluasi menunjukkan bahwa varian YOLO11m menjadi model yang paling optimal dengan nilai *precision* 85.9%, *recall* 84.1% dan *mAP50* tertinggi sebesar 89.1%. Model YOLO11m yang dipilih kemudian diuji pada data baru dan video melalui *webcam* dengan hasil kecepatan deteksi menggunakan *webcam* mencapai ± 16 FPS yang membuktikan kelayakannya untuk aplikasi *real-time*. Sistem ini dirancang untuk diimplementasikan menggunakan kamera atau *webcam* yang diarahkan ke jalur penyortiran di area pascapanen. Dengan menyediakan informasi deteksi dan jumlah setiap kelas secara otomatis, sistem ini dapat membantu petani, distributor dan industri pengolahan pangan untuk menjadikan proses penyortiran menjadi lebih cepat, konsisten dan objektif. Temuan penelitian ini diharapkan dapat menjadi solusi teknologi yang praktis serta landasan pengembangan sistem pertanian dan distribusi tomat di masa mendatang.

Kata kunci: Deteksi Objek, YOLO11, Kematangan Tomat.

GLOSARIUM

<i>Augmentasi Data</i>	Proses untuk meningkatkan variasi dan jumlah data latih secara artifisial dengan tujuan memperkuat kemampuan model dalam menghadapi variasi visual yang kompleks.
<i>Bounding Box</i>	Kotak pembatas persegi yang digunakan dalam deteksi objek untuk menentukan lokasi atau posisi presisi dari sebuah objek yang terdeteksi di dalam citra.
<i>Computer Vision</i>	Bidang ilmu yang berfokus pada pengembangan kemampuan mesin untuk memahami, mengidentifikasi dan mengukur objek melalui informasi dari dunia visual.
<i>Deep Learning</i>	Subbidang dari <i>machine learning</i> yang algoritmanya dikembangkan berdasarkan inspirasi dari struktur dan fungsi otak manusia yang dikenal sebagai Jaringan Saraf Tiruan (<i>Artificial Neural Networks</i>).
<i>Epoch</i>	Satu siklus lengkap di mana keseluruhan dataset pelatihan dimasukkan ke dalam model untuk proses pelatihan.
<i>Hyperparameter</i>	Parameter eksternal yang nilainya dapat disesuaikan untuk mengontrol dan memengaruhi struktur serta proses pembelajaran model.
<i>mAP</i>	Metrik yang mengukur performa model dengan menghitung nilai rata-rata dari <i>Average Precision</i> (AP) untuk seluruh kelas objek dan memberikan evaluasi yang komprehensif pada skenario deteksi multi-kelas.
<i>Overfitting</i>	Kondisi di mana model menjadi terlalu spesifik pada data latih sehingga kinerjanya menurun saat dihadapkan pada data baru dan sering kali disebabkan oleh jumlah iterasi pelatihan yang berlebihan.
<i>YOLO</i>	Sebuah arsitektur deteksi objek yang populer karena mampu melakukan prediksi lokasi dan kelas objek dalam satu kali proses (<i>single stage</i>) sehingga cepat dan cocok untuk aplikasi <i>real-time</i> .

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu.....	7
2.2 Buah Tomat	10
2.3 Computer Vision	11
2.4 Deteksi Objek	12
2.5 Deep Learning	13
2.6 Convolutional Neural Network	14
2.7 YOLO (You Only Look Once).....	15
2.8 Hyperparameter Tuning.....	17
2.9 Metrik Evaluasi	18
BAB III METODOLOGI PENELITIAN	22
3.1 Tahapan Penelitian	22

3.2	Pengumpulan Data.....	22
3.3	Data Preparation	24
3.4	Modelling	30
3.5	Evaluation.....	32
3.6	Pengujian Model.....	33
BAB IV HASIL DAN PEMBAHASAN		34
4.1	Hasil Pelatihan Model	34
4.2	Hasil Seluruh Varian Model.....	57
4.3	Hasil Pengujian.....	58
BAB V KESIMPULAN DAN SARAN		63
5.1	Kesimpulan.....	63
5.2	Saran	63
DAFTAR PUSTAKA		64
LAMPIRAN.....		68

DAFTAR TABEL

Tabel 3.1 Konfigurasi <i>Hyperparameter</i> Pelatihan Model	31
Tabel 4.1 Perbandingan Hasil Performa Model.....	58

DAFTAR GAMBAR

Gambar 2.1 Contoh deteksi objek satu dan beberapa objek	12
Gambar 2.2 Struktur <i>Neural Network</i> pada <i>Deep Learning</i>	13
Gambar 2.3 Arsitektur CNN	14
Gambar 2.4 Arsitektur YOLO11	17
Gambar 3.1 Tahapan penelitian	22
Gambar 3.2 Contoh Gambar Tomat pada <i>Dataset</i>	23
Gambar 3.3 Kategori Kematangan Tomat	23
Gambar 3.4 Ragam resolusi data pada gambar.....	24
Gambar 3.5 Sebelum dan Sesudah Pelabelan Objek	25
Gambar 3.6 Dokumentasi Penilaian Petani	26
Gambar 3.7 Tiga Kelas Kematangan Tomat.....	26
Gambar 3.8 Pembagian <i>dataset</i>	27
Gambar 3.9 <i>Auto Orient</i>	28
Gambar 3.10 Mengubah Resolusi pada Citra (<i>Resize</i>)	29
Gambar 3.11 Total <i>dataset</i>	30
Gambar 3.12 Informasi <i>GPU</i> yang digunakan	31
Gambar 4.1 Hasil Validasi YOLO11n.....	34
Gambar 4.2 Hasil Visualisasi Pelatihan YOLO11n.....	35
Gambar 4.3 Tren penurunan <i>loss</i> YOLO11n.....	35
Gambar 4.4 Tren Peningkatan Kinerja YOLO11n	36
Gambar 4.5 Kurva Hubungan <i>Precision</i> dan <i>Recall</i> YOLO11n.....	37
Gambar 4.6 Kurva Hubungan <i>F1</i> dan <i>Confidence</i> YOLO11n	38
Gambar 4.7 <i>Confusion Matrix normalized</i> YOLO11n	39
Gambar 4.8 Hasil Validasi YOLO11s	40
Gambar 4.9 Hasil Visualisasi Pelatihan YOLO11s	40
Gambar 4.10 Tren penurunan <i>loss</i> YOLO11s	41
Gambar 4.11 Tren Peningkatan Kinerja YOLO11s.....	42
Gambar 4.12 Kurva Hubungan <i>Precision</i> dan <i>Recall</i> YOLO11s	43
Gambar 4.13 Kurva Hubungan <i>F1</i> dan <i>Confidence</i> YOLO11s.....	44
Gambar 4.14 <i>Confusion Matrix normalized</i> YOLO11s.....	45
Gambar 4.15 Hasil Validasi YOLO11m.....	46
Gambar 4.16 Hasil Visualisasi Pelatihan YOLO11m.....	46

Gambar 4.17 Tren penurunan <i>loss</i> YOLO11m.....	47
Gambar 4.18 Tren Peningkatan Kinerja YOLO11m	48
Gambar 4.19 Kurva Hubungan <i>Precision</i> dan <i>Recall</i> YOLO11m	49
Gambar 4.20 Kurva Hubungan <i>F1</i> dan <i>Confidence</i> YOLO11m	50
Gambar 4.21 <i>Confusion Matrix normalized</i> YOLO11m	51
Gambar 4.22 Hasil Validasi YOLO11l.....	51
Gambar 4.23 Hasil Visualisasi Pelatihan YOLO11l.....	52
Gambar 4.24 Tren penurunan <i>loss</i> YOLO11l.....	53
Gambar 4.25 Tren Peningkatan Kinerja YOLO11l	54
Gambar 4.26 Kurva Hubungan <i>Precision</i> dan <i>Recall</i> YOLO11l	55
Gambar 4.27 Kurva Hubungan <i>F1</i> dan <i>Confidence</i> YOLO11l	56
Gambar 4.28 <i>Confusion Matrix normalized</i> YOLO11l.....	57
Gambar 4.29 Hasil Pengujian Gambar Satu	59
Gambar 4.30 Hasil Pengujian Gambar Dua.....	60
Gambar 4.31 Hasil Pengujian Gambar Tiga	61
Gambar 4.32 Hasil Pengujian Menggunakan <i>Webcam</i>	62

BAB I

PENDAHULUAN

1.1 Latar Belakang

Tomat (*Solanum lycopersicum* L.) merupakan salah satu tanaman buah yang paling populer dan penting di dunia (Hu et al., 2019). Tidak hanya memberikan rasa yang enak, tomat juga mengandung nutrisi yang penting bagi manusia. Tomat memiliki kandungan air yang tinggi sebanyak 95%, karbohidrat 3%, protein 1,2% dan lipid 1%. Selain itu, tomat mengandung mineral seperti kalsium (Ca), magnesium (Mg), fosfor (P), natrium (Na), kalium (K), mangan (Mn), seng (Zn) dan lain-lain. Tidak hanya mineral, tomat mengandung berbagai vitamin yang dibutuhkan tubuh seperti vitamin A dan vitamin C, asam pantotenat, tiamin, niasin, riboflavin dan piridoksin (Salehi et al., 2019). Data statistik menunjukkan pada tahun 2022, diperkirakan sebanyak 186,1 juta ton tomat diproduksi di dunia. China menempati posisi pertama dengan jumlah produksi sebesar 68,241 juta ton tomat. India berada pada posisi kedua dengan jumlah produksi sebesar 20,694 juta ton tomat, sedangkan Turki berada pada posisi ketiga dengan produksi tomat sebesar 13 juta ton (FAO, 2024). Tingginya angka yang telah diproduksi ini menunjukkan bahwa tidak hanya memiliki nilai ekonomis yang tinggi, tomat menjadi komoditas yang sangat dibutuhkan dalam memenuhi kebutuhan nasional, seperti bahan baku industri pengolahan pangan, campuran bahan olahan dan konsumsi untuk rumah tangga (Muchtar et al., 2021).

Sebagai salah satu negara agraris, Indonesia masih mengandalkan sektor pertanian yang memiliki peran penting dalam memajukan perekonomian negara (Anwar et al., 2022). Sektor pertanian, kehutanan, serta perikanan memiliki kontribusi yang signifikan sebagaimana yang terlihat pada Produk Domestik Bruto (PDB) pada triwulan II-2024 yang menjadikannya sektor kedua terbesar setelah sektor Industri Pengolahan dengan 13,78% (Badan Pusat Statistik, 2024b). Namun, kondisi tersebut tidak sejalan dengan keadaan beberapa tanaman hortikultura di Indonesia, terutama tomat. Jumlah produksi tomat di Indonesia pada tahun 2023 mencapai 1,14 juta ton, turun sebesar 2,14% (24,95 ribu ton) dari tahun 2022 yang menghasilkan 1,16 juta ton. Penurunan ini berbanding terbalik dengan konsumsi tomat pada sektor rumah tangga yang meningkat dari tahun ke tahun. Pada tahun 2023, konsumsi tomat oleh sektor rumah tangga mencapai 697,22 ribu ton, naik sebesar 1,34% (9,24 ribu ton) dari tahun 2022 (Badan Pusat Statistik, 2024a). Berdasarkan data tersebut, terlihat adanya kesenjangan antara produksi

dan konsumsi yang mengindikasikan adanya permasalahan dalam rantai pasok atau *supply chain* tomat di Indonesia. Salah satu faktor penghambat pada produksi adalah proses penyortiran yang tidak efisien. Proses ini penting untuk memastikan bahwa buah mencapai konsumen dalam kondisi terbaik, mengurangi kerugian pascapanen dan memaksimalkan keuntungan bagi petani.

Kegiatan penyortiran merupakan kegiatan dalam proses produksi dan distribusi di sektor pertanian maupun industri. Proses ini bertujuan untuk mengklasifikasikan produk berdasarkan kategori tertentu seperti warna, suhu, bentuk dan bau (Yanti et al., 2024). Pada komoditas buah tomat, penyortiran dilakukan untuk mengelompokkan ke dalam beberapa kategori berdasarkan pada variasi warna yang memperlihatkan tingkat kematangan pada tomat. Pada saat ini, metode tradisional yang banyak digunakan untuk menentukan tingkat kematangan tomat adalah dengan pendekatan manual dan bergantung pada keahlian individu. Kemudian, penentuan secara manual ini terdapat beberapa kendala seperti waktu proses yang panjang dan banyaknya tenaga kerja yang dibutuhkan karena setiap tomat harus diperiksa dan disortir secara individual oleh seorang ahli. Sebagai tambahan, kinerja mereka dapat dipengaruhi oleh beberapa faktor seperti kelelahan dan penilaian pribadi yang bervariasi. Akibatnya, metode ini menggunakan biaya yang mahal dan inefisiensi dalam segi kinerja terlebih ketika harus menangani tomat dalam jumlah yang besar (Mputu et al., 2024). Permasalahan inefisiensi ini berdampak pada seluruh rantai pasok tomat. Bagi petani, penyortiran yang terlambat dapat menyebabkan penumpukan hasil panen dan meningkatkan risiko kerusakan sebelum tomat sampai ke pasar. Untuk distributor, ketiadaan standar penyortiran yang objektif menyulitkan proses kontrol kualitas dan pemenuhan permintaan yang spesifik dari pasar modern atau industri. Sementara itu, industri pengolahan pangan sangat bergantung pada pasokan bahan baku dengan tingkat kematangan yang seragam untuk menjaga konsistensi rasa dan kualitas produk akhir. Oleh karena itu, dibutuhkan sebuah sistem otomatis yang mampu melakukan penyortiran secara cepat, akurat dan konsisten untuk menjawab tantangan di setiap level tersebut.

Teknologi *deep learning* telah berkembang pesat dalam beberapa tahun terakhir, khususnya dalam deteksi objek. Kemajuan ini membuat penerapan teknologi tersebut semakin banyak dimanfaatkan dalam deteksi buah di pertanian digital. Deteksi objek merupakan komponen penting pada pertanian digital karena dapat meningkatkan produktivitas dan hasil produksi pertanian (Badgujar et al., 2024). Deteksi objek yang berbasis *deep learning* memiliki dua jenis utama berdasarkan dari struktur jaringannya yaitu: *two-stage* dan *one-stage*. Setiap algoritma ini memiliki desain dan arsitektur yang berbeda mulai dari kecepatan, akurasi,

efisiensi dan ukuran. Contohnya adalah secara umum, *two-stage* dapat memberikan akurasi yang lebih baik tetapi memiliki kecepatan yang lebih lambat dari *one-stage* yang menyeimbangkan kecepatan dan akurasi sehingga memiliki performa yang lebih cepat (Zhang et al., 2022). Pemilihan metode *one-stage* seperti *You Only Look Once* (YOLO) pada pertanian lebih banyak dipilih dibandingkan dengan metode *two-stage* karena beberapa faktor, terutama efisiensi dan kesesuaiannya pada lingkungan yang kompleks. YOLO menggunakan proses *one-stage* yang memadukan lokalisasi dan klasifikasi dalam satu *forward pass* sehingga memberikan kecepatan yang optimal untuk deteksi secara *real-time*. Hal ini sangat penting dalam pertanian karena kondisi yang dapat berubah dengan cepat sehingga keputusan harus dibuat dengan tepat. Selain itu, ukuran YOLO yang lebih kecil dan jumlah parameter yang lebih sedikit menjadikannya lebih adaptif untuk digunakan pada perangkat dengan daya komputasi yang terbatas, seperti *mobile devices* dan *embedded systems* yang sering digunakan dalam pertanian (Zhao et al., 2024).

Terdapat beberapa penelitian terdahulu yang telah dilakukan dengan menggunakan metode YOLO. Pada penelitian yang dilakukan Wibowo et al. (2023) menggunakan YOLOv5 untuk mendeteksi buah segar dan busuk dengan akurasi sebesar 84%. Penelitian lainnya yang telah dilakukan yaitu penelitian dari Kurniawan et al. (2023) dalam mengklasifikasi kematangan buah sawit dengan YOLOv5 menghasilkan akurasi sebesar 90%. Selanjutnya penelitian dari Muarofah et al. (2023) yang menggunakan arsitektur YOLOv7-tiny untuk deteksi kualitas tomat memperoleh hasil akurasi sebesar 90.6% pada gambar orisinal dengan rata-rata IoU sebesar 0.90 dan waktu deteksi selama 2.7 detik.

Berdasarkan pemaparan di atas, maka penelitian ini akan mengembangkan model deteksi tingkat kematangan buah tomat dengan memanfaatkan *You Only Look Once* (YOLO) versi terbaru, yaitu YOLO11. Dipilihnya YOLO11 didasarkan pada potensinya untuk menawarkan peningkatan lebih lanjut dalam hal akurasi dan kecepatan deteksi dibandingkan dengan pendahulunya. Adanya penelitian ini, diharapkan dapat membantu meningkatkan efisiensi dan menurunkan biaya operasional serta dapat memberikan kontribusi terhadap pengembangan pertanian digital di Indonesia.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, adapun rumusan masalah pada penelitian ini adalah:

- a. Bagaimana merancang dan mengembangkan model deteksi tingkat kematangan buah tomat menggunakan YOLO11?
- b. Bagaimana performa model YOLO11 yang telah dikembangkan dalam mendeteksi tingkat kematangan buah tomat jika diukur menggunakan metrik evaluasi standar?
- c. Bagaimana hasil uji coba model YOLO11 dalam deteksi tingkat kematangan buah tomat secara *real-time*?

1.3 Batasan Masalah

Agar masalah yang diteliti tidak terlalu luas dan menyimpang maka dilakukan pembatasan masalah, batasan-batasan tersebut yaitu:

- a. Penelitian ini hanya berfokus pada buah tomat.
- b. Kategori tingkat kematangan yang dideteksi sebanyak tiga kelas yaitu mentah, setengah matang dan matang.
- c. Penelitian ini difokuskan menggunakan algoritma model YOLO11 dan tidak membandingkan dengan algoritma lainnya.

1.4 Tujuan Penelitian

Tujuan penelitian ini yaitu:

- a. Merancang dan mengembangkan model YOLO11 untuk mendeteksi tingkat kematangan buah tomat.
- b. Mengetahui hasil pengujian dan mendeskripsikan kemampuan model YOLO11 yang telah dikembangkan untuk mendeteksi tingkat kematangan tomat.
- c. Mengevaluasi performa model YOLO11 yang telah dikembangkan berdasarkan metrik seperti *precision*, *recall* dan *mAP50*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat praktis dan akademis. Secara praktis, model deteksi tingkat kematangan tomat yang dikembangkan dapat menjadi solusi teknologi untuk meningkatkan efisiensi dan objektivitas dalam proses penyortiran di berbagai titik rantai pasok. Bagi petani, sistem ini bermanfaat sebagai alat bantu pascapanen untuk memisahkan tomat sesuai standar pasar sehingga berpotensi mengurangi kerugian dan meningkatkan nilai jual. Bagi distributor dan industri pengolahan pangan, sistem ini dapat diimplementasikan untuk melakukan kontrol kualitas yang cepat dan konsisten untuk

memastikan keseragaman produk dan mengoptimalkan proses produksi. Secara akademis, hasil penelitian ini dapat menjadi referensi yang relevan bagi pengembangan lebih lanjut untuk sektor pertanian digital di Indonesia.

1.6 Metodologi Penelitian

Penelitian ini dilakukan secara sistematis melalui beberapa tahapan untuk mencapai tujuan yang telah ditetapkan sebelumnya. Tahapan awal adalah pengumpulan data yang meliputi pengumpulan *dataset* citra buah tomat dari berbagai sumber. Tahap selanjutnya adalah persiapan data yang mencakup proses *preprocessing* seperti pelabelan objek dengan *bounding box*, pembagian *dataset* menjadi data latih, validasi dan uji serta penyesuaian resolusi citra. Proses ini ditambah dengan augmentasi data untuk meningkatkan variasi dan jumlah untuk data latih. Tahap selanjutnya yaitu pemodelan menggunakan YOLO11 yang dilatih menggunakan *dataset* yang telah dipersiapkan untuk membangun model deteksi. Setelah itu, akan dilakukan evaluasi untuk mengukur kinerja model secara kuantitatif menggunakan metrik seperti *precision*, *recall* dan *mAP50* guna memilih varian model dengan performa yang terbaik. Sebagai tahap akhir, akan dilakukan pengujian model yang telah dipilih dengan data citra baru.

1.7 Sistematika Penulisan

Sistematika pada penelitian ini terbagi menjadi lima bab. Penjelasan dari sistematika penulisan ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab pertama ini menjelaskan gambaran umum penelitian yang mencakup latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan menyajikan pembahasan mengenai teori, konsep serta penelitian terdahulu yang relevan dengan topik penelitian. Kemudian, analisis terhadap penelitian-penelitian terdahulu akan dilakukan untuk melihat keterbatasan serta menemukan celah penelitian yang ada. Tujuan dari kajian ini adalah untuk membangun landasan teoritis yang kuat sekaligus memperjelas posisi penelitian ini secara strategis pada konteks literatur yang sudah ada.

BAB III METODOLOGI PENELITIAN

Bab ini memaparkan secara rinci rancangan dan alur kerja penelitian. Pembahasan meliputi metode pengumpulan data dan persiapan data, perancangan model, prosedur pelatihan beserta konfigurasinya serta skenario pengujian untuk mengevaluasi performa model.

BAB IV HASIL DAN PEMBAHASAN

Bab ini akan menyajikan hasil yang diperoleh dari penerapan metodologi penelitian dan dilanjutkan dengan pembahasan mengenai temuan-temuan yang ada. Pemaparan diawali dengan penyajian data hasil pelatihan dan evaluasi setiap varian model, kemudian menganalisis temuan tersebut dengan membandingkan performa masing-masing model untuk menentukan model paling optimal dan interpretasi terhadap hasil pengujian akhir.

BAB V KESIMPULAN DAN SARAN

Bab terakhir ini merangkum temuan-temuan dari penelitian dan memberikan kesimpulan berdasarkan hasil analisis. Kemudian, menyertakan beberapa saran yang dapat dimanfaatkan untuk pengembangan penelitian di masa mendatang.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penerapan berbagai algoritma dengan teknologi untuk deteksi dan klasifikasi tingkat kematangan buah dalam sektor pertanian bukanlah merupakan topik yang baru. Sejumlah besar penelitian yang relevan telah dilakukan dengan menggunakan *dataset*, metode, fungsi optimasi yang berbeda-beda dan perbedaan teknik pengolahan citra baik buah tomat yang menjadi fokus utama penelitian ini maupun komoditas buah lainnya. Oleh karena itu, pemahaman mendalam terhadap penelitian-penelitian terdahulu dapat membantu penelitian ini untuk menambah referensi, menemukan peluang pengembangan atau kebaruan serta menghindari duplikasi dalam penelitian ini. Model You Only Look Once (YOLO) telah banyak digunakan dalam berbagai penelitian untuk deteksi dan klasifikasi buah berdasarkan citra. Berbagai versi YOLO menunjukkan kemampuan yang berbeda-beda tergantung pada objek dan kompleksitas tugas. Seperti model YOLOv5 yang telah menjadi pilihan pada berbagai penelitian.

Penelitian yang dilakukan oleh Aras et al. (2024) menggunakan YOLOv5 dalam menentukan tingkat kematangan buah tomat dalam tiga kategori yaitu muda, setengah matang dan matang. Penelitian ini menggunakan total 1.166 gambar tomat yang terdiri dari 981 gambar untuk data pelatihan, 121 untuk validasi dan 64 untuk pengujian. Proses penelitian melibatkan beberapa tahapan utama yaitu pengumpulan *dataset*, pelabelan objek berdasarkan tingkat kematangan, pra-pemrosesan data hingga pelatihan model menggunakan platform *Google Colab*. Evaluasi kinerja model dilakukan dengan metrik standar seperti akurasi, *precision* dan *recall*. Hasil penelitian menunjukkan nilai akurasi sebesar 73% dan metrik nilai *F1-Score* yang mencapai puncak rata-rata 0.97 pada tingkat *confidence* 0.352 dengan nilai *precision* rata-rata adalah 1.00 pada tingkat *confidence* sebesar 0.499 yang digunakan untuk menganalisis performa sistem. Meskipun demikian, akurasi yang didapat dipengaruhi oleh beberapa faktor seperti kurangnya variasi dalam *dataset* yang digunakan, kualitas gambar, sudut pengambilan gambar dan kemungkinan objek yang bertumpuk atau terpotong. Oleh karena itu, peneliti memiliki saran untuk meningkatkan kualitas dan kuantitas *dataset* serta *confidence* yang lebih tinggi di masa mendatang.

Penelitian dengan model YOLOv5 juga dilakukan Wibowo et al. (2023) untuk deteksi buah segar dan busuk. Penelitian tersebut menggunakan *dataset* gambar keseluruhan 1.530 dengan 1.200 gambar untuk data latih, 330 gambar untuk data validasi dan 6 gambar untuk

data uji. Penggunaan platform *Google Colab* untuk proses pelatihan dengan *GPU* juga merupakan langkah relevan untuk mempercepat komputasi. Penelitian memberikan hasil nilai *precision* rata-rata sebesar 0.990 dan nilai *recall* 1.00 pada *confidence* 0.00. Selain itu, nilai *F1-Score* mendapatkan nilai puncak sebesar 0.99 terhadap nilai *confidence* 0.745. Namun, terdapat perbedaan nilai akurasi pada bagian abstrak yang memberikan hasil akurasi sebesar 90% dan kesimpulan yang memberikan nilai akurasi sebesar 84%. Kemudian, data uji yang digunakan hanya menggunakan 6 gambar. Jumlah data uji yang sangat kecil ini kurang memadai untuk menyimpulkan kemampuan performa model pada data baru yang belum pernah dilihat sebelumnya. Kemudian, Faktor-faktor seperti kualitas *dataset*, variasi sudut pengambilan gambar dan kualitas gambar diakui oleh peneliti dapat mempengaruhi akurasi.

Meskipun YOLOv5 menunjukkan kinerja awal yang menjanjikan untuk deteksi tingkat kematangan buah, keberhasilannya masih dipengaruhi oleh kualitas dan kuantitas *dataset*, variasi pada kondisi pengambilan gambar dan ketepatan dalam metodologi penelitian. Keterbatasan ini membuka peluang untuk penelitian yang mengeksplorasi arsitektur yang lebih baru untuk ketahanan dan efisiensi yang lebih baik pada kondisi data yang kurang optimal. Selain YOLOv5, terdapat beberapa penelitian yang menggunakan model YOLO dengan versi yang berbeda.

Penelitian oleh Prasetya et al. (2024) yang mengimplementasikan algoritma YOLOv8 untuk mendeteksi dan mengklasifikasikan tingkat kematangan buah nanas dengan mengidentifikasi empat kategori kematangan yaitu mentah, setengah matang, matang dan terlalu matang. Pengujian dilakukan menggunakan foto dan video menggunakan platform Roboflow serta *input* kamera langsung menggunakan *webcam*. Hasil pengujian menunjukkan nilai *mAP* sebesar 81%, *precision* 70,5% dan *recall* 75,9%. Hasil visualisasi juga digunakan untuk mengevaluasi kinerja model. Meskipun demikian penelitian ini terdapat beberapa kekurangan. Kurangnya variasi dalam *dataset* yang hanya menggunakan 153 gambar tidak memadai untuk melatih model *deep learning* seperti YOLOv8 yang umumnya memerlukan ribuan gambar untuk generalisasi yang optimal. Meskipun penelitian menggunakan platform Roboflow untuk anotasi, tidak ada informasi mengenai augmentasi yang digunakan untuk meningkatkan variasi *dataset*. Hal ini berpotensi menyebabkan *overfitting* karena hasil validasi dan pengujian menjadi terbatas sehingga model di lingkungan *real-time* mungkin tidak stabil.

Kemudian, Muarofah et al. (2023) mengembangkan sistem deteksi kualitas tomat secara *real-time* menggunakan algoritma YOLOv7-tiny yang merupakan varian ringan dari YOLOv7. Penelitian ini mengklasifikasikan tomat menjadi tiga kategori yaitu mentah, matang dan busuk.

Dataset yang digunakan sebanyak 1.810 gambar. Model yang dilatih menunjukkan performa yang cukup baik pada *set* validasi dengan *recall* 0.97, *precision* 1.0 dan akurasi pada gambar asli mencapai 90.6% dengan rata-rata IoU 0.90 dan waktu deteksi 2.7 detik. Pengujian dengan variasi kondisi pencahayaan juga menunjukkan hasil yang menjanjikan dengan akurasi rata-rata 91.2% untuk gambar dengan tambahan gangguan cahaya dan 92% untuk gambar dengan pengurangan cahaya. Namun, terdapat penurunan akurasi menjadi 78.2% pada gambar dengan gangguan buram, yang mengindikasikan potensi kelemahan model dalam menghadapi kondisi gambar yang kurang ideal. Kemudian, pembagian *dataset* 80% untuk data latih dan validasi sebesar 30% yang secara matematis tidak konsisten jika merujuk pada persentase dari keseluruhan *dataset*.

Selain pendekatan modern berbasis *deep learning* seperti YOLO, penelitian di bidang ini juga masih memanfaatkan metode pengolahan citra digital konvensional. Salah satunya adalah penelitian yang dilakukan oleh Hanny Hikmayanti H et al. (2021) yang menggunakan metode Hue Saturation Value (HSV) untuk mengidentifikasi kematangan tomat. Meskipun memiliki potensi, penelitian ini memiliki beberapa batasan. Pertama, sistem hanya mencapai akurasi rata-rata pada 80% pada 20 kali pengujian. Kedua, penulis menyarankan bahwa pengembangan di masa mendatang dapat menggunakan metode atau algoritma lain untuk meningkatkan akurasi yang lebih baik. Keterbatasan lainnya adalah skala pengujian yang sangat kecil dengan hanya 20 data uji dari 165 citra tomat pada *dataset*. Jumlah tersebut tidak cukup untuk memvalidasi kinerja sistem pada berbagai kondisi. Keterbatasan ini menunjukkan bahwa meskipun metode konvensional dapat digunakan, pendekatan *deep learning* sering kali lebih unggul dalam hal akurasi, skalabilitas dan ketahanan.

Berdasarkan kajian pada penelitian terdahulu, terdapat perbedaan pada pemilihan model YOLO yang digunakan pada penelitian ini. Penelitian ini akan mengimplementasikan YOLO11 yang merupakan iterasi terbaru yang dikembangkan oleh Jocher & Qiu (2024) dari Ultralytics. Berbeda dengan penelitian terdahulu yang menggunakan versi yang lama seperti YOLOv5, YOLOv7, YOLOv8 maupun metode konvensional lainnya. YOLO11 menawarkan sejumlah peningkatan arsitektur dan optimalisasi yang berbeda pada versi sebelumnya termasuk efisiensi komputasi yang lebih baik, akurasi deteksi yang ditingkatkan dan kemampuan generalisasi yang lebih kuat. Keunggulan teknologi terbaru ini diharapkan dapat mengatasi berbagai keterbatasan yang ditemukan pada penelitian-penelitian sebelumnya. Dengan demikian, penelitian ini diharapkan dapat berkontribusi dalam memberikan solusi deteksi tingkat kematangan dengan memanfaatkan perkembangan teknologi terkini.

2.2 Buah Tomat

Buah tomat (*Solanum lycopersicum* L.) merupakan spesies tumbuhan yang berasal dari benua Amerika. Tanaman ini cenderung memiliki siklus hidup yang relatif pendek serta tinggi tanaman mencapai satu hingga tiga meter. Buah tomat termasuk dalam keluarga terung-terungan atau Solanaceae dan termasuk ordo Solanales. Tomat merupakan salah satu sayuran terkemuka di dunia dan juga dapat dikonsumsi mentah sebagai buah. Buah ini disukai karena nutrisinya yang kaya dan rasanya yang enak. Tomat kaya akan serat serta vitamin A dan C yang merupakan antioksidan kuat. Konsumsi tomat dapat membantu mengurangi risiko osteoporosis, masalah kardiovaskular dan beberapa jenis kanker (Wang et al., 2024).

Buah tomat merupakan salah satu buah yang memiliki variasi warna yang kompleks untuk menentukan indikator kematangannya. Secara umum, buah tomat memiliki tiga warna untuk menentukan apakah buah tersebut sudah matang, setengah matang dan mentah yaitu warna merah menandakan kondisi matang, warna kuning menandakan setengah matang dan warna hijau menandakan buah tomat mentah (Hetharua et al., 2021). Perubahan warna ini terjadi selama proses pematangan akibat reaksi fisiologis dan biokimiawi yang kompleks pada buah. Proses ini terjadi juga dipengaruhi oleh kondisi lingkungan eksternal selama masa pertumbuhan tomat. Oleh karena itu, Tingkat kematangan merupakan salah satu faktor penting dalam menilai kualitas pada tomat. Penentuan waktu panen juga sangat berkaitan erat dengan tingkat kematangan tersebut. Jika tomat dipanen terlalu dini, maka tomat mungkin tidak akan mencapai tingkat kematangan yang diinginkan sehingga mengurangi potensi manfaat ekonomisnya. Sebaliknya, jika panen terlambat maka tomat bisa menjadi terlalu matang dan mudah rusak selama proses distribusi. Oleh karena itu, tingkat kematangan yang tepat perlu dipastikan guna menjamin kualitas produk selama pemrosesan, penyimpanan maupun distribusi menggunakan transportasi jarak jauh (Wang et al., 2024).

Penelitian oleh (Sacer et al., 2022) yang mempelajari efektivitas lama transportasi darat yang berbeda terhadap mutu fisik tomat menemukan bahwa persentase kerusakan, susut bobot dan kadar air sangat dipengaruhi oleh lama transportasi. Namun, parameter warna tidak menunjukkan perubahan pada waktu transportasi selama 4 dan 8 jam. Hasil penelitian tersebut mengindikasikan bahwa warna sebagai indikator kematangan yang relatif stabil terhadap gangguan eksternal sehingga layak dijadikan sebagai parameter utama dalam deteksi kematangan secara visual. Selain itu, warna pada tomat memiliki peran penting sebagai daya

tarik utama bagi pembeli, mengingat penampilan visual merupakan dasar penilaian awal konsumen terhadap sebuah produk pangan.

Berdasarkan karakteristik tersebut, kompleksitas warna dan perannya sebagai indikator kematangan yang stabil menjadikan tomat sebagai objek ideal untuk pengembangan sistem deteksi otomatis. Kebutuhan industri akan penilaian yang objektif, konsisten dan cepat mendorong adanya pengembangan teknologi yang mampu meniru kemampuan visual manusia untuk mengenali, membedakan serta menginterpretasikan fitur visual secara akurat.

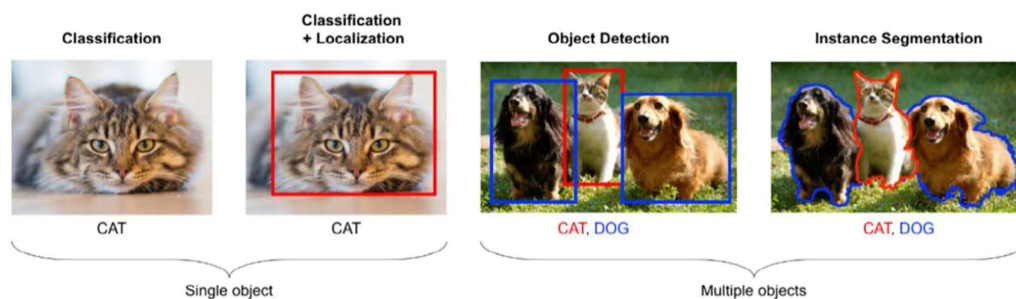
2.3 Computer Vision

Computer Vision (CV) adalah bidang ilmu yang berfokus pada pengembangan kemampuan mesin untuk memahami informasi melalui dunia visual. Teknologi ini memanfaatkan kamera dan komputer untuk mengidentifikasi, melacak dan mengukur objek. Data visual yang telah diproses tersebut selanjutnya dapat disajikan menjadi informasi yang lebih mudah dianalisis oleh manusia atau menjadi *input* bagi sistem lain untuk mendukung proses deteksi dan pengambilan keputusan (Liu et al., 2021). Beberapa tugas yang umum dilakukan dalam *computer vision* meliputi deteksi objek, pengenalan wajah, segmentasi citra, estimasi gerakan dan lain sebagainya. Untuk melaksanakan tugas-tugas tersebut, teknik-teknik dalam *computer vision* memanfaatkan algoritma dan metode matematika untuk menganalisis citra dan mengekstrak fitur-fitur penting dari data visual (Hanum, 2024).

Kualitas fitur pada citra memegang peranan penting karena secara langsung menentukan tingkat performa sistem *computer vision* (Che et al., 2024). Untuk mengekstrak fitur yang relevan, berbagai metode telah dikembangkan mulai dari metode tradisional hingga berbasis *deep learning*. Fitur-fitur inilah yang kemudian menjadi dasar bagi mesin untuk melakukan tugas-tugas spesifik. Salah satu tugas paling mendasar dalam CV yang sangat bergantung pada kualitas fitur adalah deteksi objek. Tugas ini tidak hanya menuntut sistem untuk mengenali kategori objek, tetapi juga menentukan lokasi presisinya pada citra. Dalam penelitian ini, kemampuan deteksi objek menjadi langkah awal sebelum sistem dapat melakukan analisis lebih lanjut terhadap atribut visual buah seperti warna untuk menentukan tingkat kematangannya.

2.4 Deteksi Objek

Object Detection atau deteksi objek merupakan proses untuk mengidentifikasi dan menentukan posisi satu atau beberapa objek dalam sebuah citra. Deteksi objek biasanya menggunakan algoritma *supervised learning* yang dilatih menggunakan *dataset* berlabel yang berskala besar (Diwan et al., 2023). Deteksi objek dapat dianggap sebagai kombinasi dari klasifikasi citra untuk menentukan label kelas pada objek dan lokalisasi objek untuk menentukan posisi objek menggunakan kotak pembatas atau *bounding box* (Kaur & Singh, 2022).



Gambar 2.1 Contoh deteksi objek satu dan beberapa objek

Sumber: (Kaur & Singh, 2022)

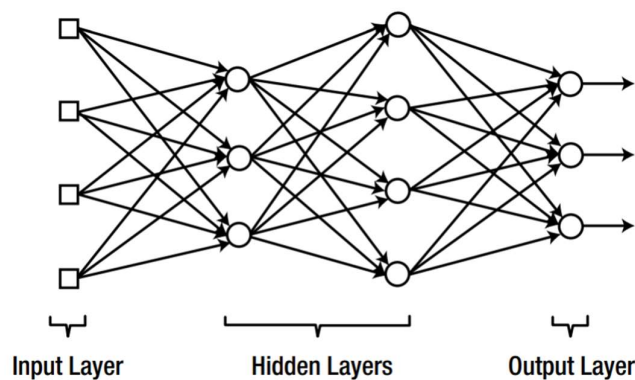
Teknik untuk deteksi objek termasuk ke dalam pendekatan berbasis *machine learning* dan pendekatan berbasis *deep learning*. Berbagai model pendekatan tersebut telah diterapkan untuk meningkatkan kinerja pada proses deteksi objek serta tugas-tugas terkait. Sebelumnya, detektor objek dua tahap (*two stage detector*) merupakan salah satu metode yang populer dan dinilai efektif. Namun, berkat perkembangan terkini pada deteksi objek satu tahap (*one stage detector*) beserta algoritma yang mendasarinya membuat metode tersebut menjadi lebih baik jika dibandingkan dengan sebagian besar detektor objek dua tahap.

Detektor objek dua tahap (*two stage detector*) membagi tugas deteksi menjadi dua tahap. Tahap pertama yaitu menghasilkan sejumlah usulan wilayah (*region proposals*) yang berpotensi memuat objek dan tahap kedua yaitu mengklasifikasikan setiap usulan wilayah tersebut untuk menentukan kelas objek dan menyempurnakan posisi *bounding box*. Algoritma populer yang menggunakan metode ini seperti Fast R-CNN dan Faster R-CNN yang menggunakan *Region Proposal Network* untuk menghasilkan usulan wilayah secara efisien. Namun, prosesnya yang terdiri dari dua tahap membuat kecepatannya menjadi relatif lambat sehingga kurang direkomendasikan jika membutuhkan respons secara *real-time*. Kemudian,

detektor satu tahap (*one stage detector*) dapat memprediksi kelas dan *bounding box* secara langsung dalam satu lintasan melalui jaringan saraf. metode ini memiliki kecepatan komputasi yang lebih tinggi sehingga ideal untuk sistem yang membutuhkan respons cepat. Contoh yang termasuk pada pendeteksi objek satu tahap adalah dengan menggunakan YOLO, SSD, SqueezeDet dan DetectNet (Marpaung et al., 2022).

2.5 Deep Learning

Deep Learning (DL) merupakan subbidang dari *Machine Learning* yang algoritmanya dikembangkan dengan inspirasi dari struktur dan fungsi otak manusia. Struktur ini dikenal sebagai Jaringan Saraf Tiruan atau *Artificial Neural Networks* (ANN). Secara spesifik, DL menggunakan ANN dengan banyak lapisan untuk belajar dan beradaptasi terhadap data dalam jumlah yang besar. Teknologi ini sangat efektif untuk mengolah data mentah yang tidak terstruktur sehingga berguna untuk mengenali pola dan informasi secara otomatis. Kemampuan inilah yang membuat DL dapat menyelesaikan permasalahan kompleks yang sering kali sulit dipecahkan oleh algoritma ML lainnya (Naf'an et al., 2022).



Gambar 2.2 Struktur *Neural Network* pada *Deep Learning*

Sumber: Nima & Shila (2020)

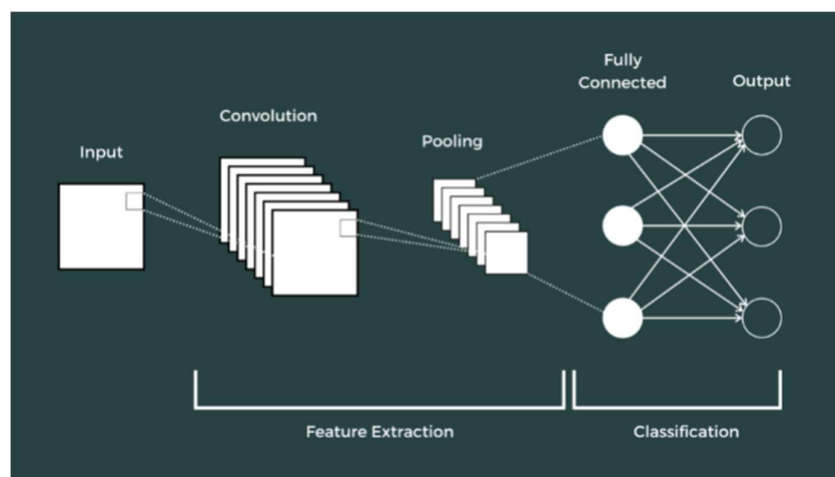
Arsitektur DL pada Gambar 2.2 merupakan arsitektur dasar dari sebuah *Multi Layer Perceptron* (MLP). Struktur tersebut secara umum terdiri dari tiga komponen lapisan utama yaitu *Input Layer*, *Hidden layers* dan *Output layer*. *Input layer* memiliki fungsi untuk menerima nilai-nilai fitur awal dari setiap sampel data seperti pada penelitian ini yang belajar mengenali fitur-fitur seperti gradasi warna dari warna hijau ke warna merah atau bentuk dari buah tomat. Data ini kemudian diproses di dalam *hidden layer* yang menjadi inti dari proses komputasi dan

ekstraksi fitur. Jumlah *hidden layer* dan jumlah neuron di dalamnya menentukan kompleksitas model. Terakhir, *output layer* menghasilkan hasil akhir dari jaringan yang dapat berupa probabilitas kelas atau nilai prediksi seperti menentukan apakah tomat tersebut pada kategori matang, setengah matang atau mentah berdasarkan kombinasi warna dan bentuk dari yang telah dipelajari.

2.6 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu arsitektur dari *Deep Learning* (DL) yang dikembangkan sebagai evolusi dari *Multi Layer Perceptron* (MLP). Keunggulan CNN terletak pada kemampuannya yang terinspirasi dari cara kerja cortex visual manusia dalam memproses informasi visual (Putra et al., 2021). Proses peniruan cortex visual ini diimplementasikan melalui serangkaian lapisan yang masing-masing memiliki fungsi spesifik untuk mengekstraksi fitur dari data.

Arsitektur dari CNN terdiri dari beberapa macam lapisan penyusun seperti pada Gambar 2.3, yaitu *convolutional layer*, *pooling layer* dan *fully connected layer*.



Gambar 2.3 Arsitektur CNN

Sumber: Alhasan & Hamza (2023)

Lapisan pertama adalah *convolutional layer* yang bertanggung jawab untuk melakukan ekstraksi fitur dari citra *input* secara otomatis. Pada lapisan ini, serangkaian filter atau *kernel* digeser secara sistematis ke seluruh bagian citra untuk mendeteksi pola visual mulai dari fitur sederhana seperti tepi dan tekstur hingga bentuk yang lebih kompleks. Hasil dari proses ini

adalah sekumpulan peta fitur (*feature maps*) yang memuat informasi keberadaan fitur-fitur tersebut. Selanjutnya, *pooling layer* dapat diterapkan dengan mekanisme kerja yang serupa dengan lapisan pertama. Namun, perbedaannya adalah lapisan ini memiliki parameter yang lebih sedikit dan prosesnya menyebabkan hilangnya sebagian informasi. Meskipun terjadi reduksi informasi, lapisan ini memainkan peran penting dalam merampingkan arsitektur CNN dan meningkatkan efisiensinya secara keseluruhan (Daniels et al., 2021). Sebagai tahap final, *fully connected layer* mengintegrasikan seluruh informasi fitur yang telah diekstraksi dan disederhanakan untuk menghasilkan *output* akhir. Lapisan ini bekerja seperti jaringan saraf tiruan konvensional yang di mana setiap neuron terhubung ke semua neuron dari lapisan sebelumnya untuk memetakan representasi fitur tersebut ke dalam hasil prediksi atau klasifikasi yang diinginkan (Sihananto et al., 2023).

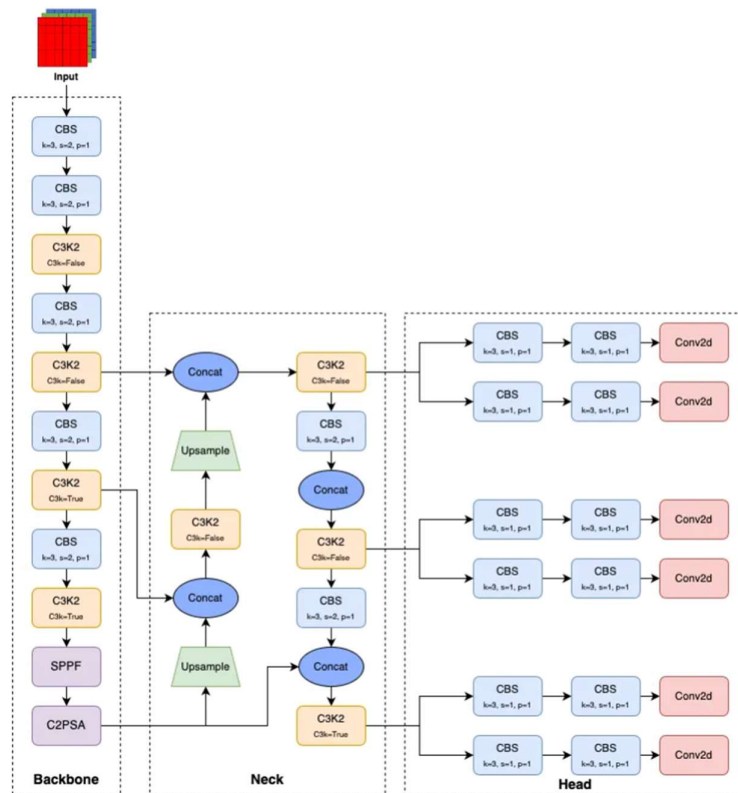
Salah satu penerapan utama dari CNN adalah pada bidang deteksi objek. Arsitektur deteksi objek modern umumnya dapat dibagi menjadi tiga komponen utama. Komponen pertama yang dikenal sebagai *backbone* yang memanfaatkan arsitektur CNN untuk melakukan ekstraksi fitur dari citra dan menghasilkan peta fitur (*feature maps*). Komponen selanjutnya adalah *neck* yang menghubungkan *backbone* dan *head*. Komponen ini bertanggung jawab untuk mengolah peta fitur yang dihasilkan oleh *backbone* agar lebih berguna untuk deteksi objek. Terakhir adalah komponen *head* yang bertanggung jawab untuk melakukan prediksi akhir berdasarkan fitur yang telah dihasilkan oleh *neck*. Bagian ini akan menghasilkan prediksi berupa *bounding box*, *confidence scores* dan klasifikasi kelas pada objek yang terdeteksi (Muzammil & Indraswari, 2024).

2.7 YOLO (You Only Look Once)

YOLO (*You Only Look Once*) adalah sebuah arsitektur deteksi objek yang dikenal karena kecepatannya. Metode ini menggunakan hanya satu jaringan konvolusional (*convolutional network*) yang digunakan untuk memprediksi objek apa saja yang ada pada citra. Hal ini dapat terjadi karena YOLO membagi citra menjadi beberapa sel atau *grid* yang nantinya bertanggung jawab untuk memprediksi jumlah *bounding box*, *confidence score* dan probabilitas kelas (Daniels et al., 2021). Arsitektur ini diperkenalkan oleh Redmon et al. (2016) melalui makalah yang berjudul “*You Only Look Once: Unified, Real-Time Object Detection*” atau yang sering disebut YOLOv1. Keunggulan utama dari YOLO adalah kecepatannya yang tinggi dan efisiensi komputasi menjadikannya cocok untuk berjalan secara *real-time* untuk mendeteksi objek pada video dan citra (Marpaung et al., 2022).

Versi terbaru YOLO yang dikembangkan oleh Ultralytics, yaitu YOLO11 menawarkan peningkatan dalam akurasi dan efisiensi dibandingkan dari versi-versi sebelumnya. Arsitektur YOLO11 menggunakan pendekatan *single stage detector* yang memungkinkan pemrosesan citra dalam satu *forward pass* yang menghasilkan prediksi lebih cepat dibandingkan dengan metode *two stage detector*. Selain itu, YOLO11 memberikan lima model dengan skala yang berbeda-beda mulai dari *nano* hingga *extra large* untuk memenuhi berbagai kebutuhan. Seperti YOLOv8, YOLO11 mencakup berbagai kegunaan seperti deteksi objek, segmentasi, klasifikasi, estimasi pose dan OBB.

Peningkatan utama pada YOLO11 meliputi pengenalan modul C2PSA seperti yang ditunjukkan pada Gambar 2.4. Modul ini menggabungkan keunggulan dari jaringan *cross-stage partial* dengan mekanisme *self attention*. Hal ini memungkinkan model untuk menangkap informasi kontekstual secara lebih efektif di berbagai lapisan sehingga meningkatkan akurasi deteksi objek terutama untuk objek yang berukuran kecil. Kemudian, blok C2f pada YOLO11 telah digantikan oleh C3k2 yang merupakan sebuah implementasi kostum dari *CSP Bottleneck* yang menggunakan dua konvolusi yang berbeda dengan YOLOv8 yang menggunakan satu konvolusi besar. Blok ini menggunakan *kernel* yang lebih kecil sehingga dapat mempertahankan akurasi dan meningkatkan efisiensi dan kecepatan (Jegham et al., 2024).



Gambar 2.4 Arsitektur YOLO11

Sumber: (Tsoi, 2024)

2.8 Hyperparameter Tuning

Hyperparameter pada model *Deep Learning* (DL) merupakan parameter yang dapat disesuaikan untuk memengaruhi struktur model dan algoritma pembelajarannya. Pengaturan nilai *hyperparameter* yang tepat sangat penting karena secara langsung berdampak pada performa dan proses pembelajaran pada model. Beberapa *hyperparameter* yang umum digunakan pada DL dan yang perlu diperhatikan meliputi *learning rate*, *batch size* dan jumlah *epoch*.

Learning rate adalah faktor penskalaan yang digunakan oleh algoritma *backpropagation* untuk memperbarui bobot (*weight*). Nilai *learning rate* yang terlalu rendah dapat memperlambat proses pembelajaran dan jika nilai yang terlalu tinggi berisiko pada model yang dapat melewati nilai minimum optimal. Umumnya, nilai ini diatur pada pangkat 10, seperti 0,1, 0,01 atau 0,001. Kemudian, *batch size* adalah jumlah sampel data yang diproses secara bersamaan dalam satu iterasi. Ukuran *batch* yang kecil dapat mengurangi penggunaan memori dan mempercepat komputasi, tetapi dapat meningkatkan ketidakstabilan akibat adanya *noise*.

Sebaliknya, jika ukuran *batch* yang besar dapat memberikan stabilitas estimasi gradien, tetapi juga meningkatkan kebutuhan memori serta waktu komputasi yang lebih lama. Terakhir, jumlah *epoch* adalah jumlah berapa kali keseluruhan *dataset* dimasukkan ke dalam model untuk dilatih. Satu *epoch* berarti seluruh *dataset* telah diproses sebanyak satu kali. Sebagai contoh, jika terdapat 1000 data dan ukuran *batch* adalah 100 maka dibutuhkan 10 iterasi untuk menyelesaikan satu *epoch*. Penentuan jumlah *epoch* yang tepat menjadi isu penting dalam model DL. Jumlah iterasi yang tidak cukup dapat mengakibatkan *underfitting* karena pelatihan yang tidak memadai, sedangkan jumlah iterasi yang berlebihan dapat menyebabkan *overfitting* (White, 2022).

Dengan demikian, keberhasilan sebuah model *deep learning* sangat bergantung pada penentuan nilai *hyperparameter* yang tepat. Pemilihan *learning rate*, *batch size* dan jumlah *epoch* secara langsung mengendalikan stabilitas dan efisiensi proses pelatihan. Konfigurasi yang optimal tidak hanya membantu mencapai keseimbangan antara akurasi dan kecepatan, tetapi juga memastikan model terhindar dari *underfitting* dan *overfitting* sehingga mampu bekerja dengan baik pada data yang belum pernah dilihat sebelumnya.

2.9 Metrik Evaluasi

Metrik evaluasi merupakan tolak ukur yang dipakai untuk menilai seberapa baik kinerja sebuah model deteksi objek seperti YOLO. Tujuannya adalah untuk mengukur secara angka seberapa akurat dan keandalannya dalam mendeteksi objek. Penilaian ini penting untuk mengetahui kemampuan dan performa model tersebut saat dihadapkan pada data yang benar-benar baru dan belum pernah dipelajari sebelumnya. Setiap metrik memberikan sudut pandang penilaian yang berbeda-beda. Oleh karena itu, pemilihan metrik harus disesuaikan dengan masalah spesifik yang ingin diselesaikan dan juga berdasarkan karakteristik data yang digunakan. Metrik utama yang relevan dalam konteks deteksi objek seperti YOLO meliputi *Intersection over Union* (IoU), *Average Precision* (AP), *mean Average Precision* (mAP), *Precision*, *Recall* dan *F1-Score*.

IoU adalah metrik evaluasi yang digunakan untuk mengukur tumpang tindih antara kotak pembatas hasil prediksi oleh model dengan kotak pembatas kebenaran dasar (*ground truth*). Perhitungan IoU dilakukan dengan cara membandingkan luas area irisan (*intersection*) antara kedua kotak tersebut dengan luas gabungan (*union*). Dengan demikian, nilai IoU yang semakin tinggi maka akan semakin baik model untuk mendeteksi objek dengan akurat. IoU dapat dihitung dengan formula (2.1):

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}} \quad (2.1)$$

- a. *Area of Intersection* adalah luas dari area di mana kotak prediksi dan kotak pembatas kebenaran dasar saling tumpang tindih.
- b. *Area of Union* adalah total luas gabungan yang dicakup oleh kedua kotak tersebut. Luas ini dihitung dengan menjumlahkan luas masing-masing kotak dan kemudian mengurangnya dengan *Area of Intersection* untuk menghindari penghitungan ganda pada area yang tumpang tindih.

Nilai IoU memiliki rentang antara 0 hingga 1. Nilai 1 mengindikasikan bahwa kotak pembatas yang diprediksi dan kotak pembatas kebenaran dasar sepenuhnya tumpang tindih dengan sempurna. Sebaliknya, nilai 0 mengindikasikan tidak ada tumpang tindih sama sekali antara kotak prediksi dan kotak kebenaran dasar.

AP merupakan metrik yang merangkum kinerja sebuah model ke dalam satu nilai tunggal. Metrik ini mengukur kemampuan model dengan menyeimbangkan *precision* dan *recall*. *Precision* mengukur proporsi prediksi positif yang relevan dari total prediksi yang dibuat, sementara *recall* mengukur kemampuan model untuk menemukan kembali seluruh objek relevan yang ada. AP dihitung sebagai luas area di bawah kurva *precision-recall*, yang secara efektif memberikan merepresentasikan kemampuan model dalam menyeimbangkan antara *false positive* (FP) dan *false negative* (FN). Rumus untuk perhitungan AP (2.2) adalah sebagai berikut:

$$AP = \sum_n (Recall_n - Recall_{n-1}) Precision_n \quad (2.2)$$

Nilai AP berada dalam rentang nilai 0 hingga 1. Nilai 1 menunjukkan kinerja model yang sempurna, di mana model mampu mendeteksi objek dengan jumlah *false positives* dan *false negatives* yang minimal. Oleh karena itu, metrik AP ini penting untuk menilai performa model pada tugas deteksi objek, terutama yang mengharuskan keseimbangan antara *precision* dan *recall*.

mAP adalah metrik yang lebih kompleks dibandingkan dengan *precision*. Metrik ini mengukur performa model dengan menghitung nilai rata-rata dari AP untuk seluruh kelas objek

yang ada pada *dataset* sehingga *mAP* memberikan evaluasi yang lebih komprehensif terhadap kinerja model dalam skenario deteksi multi-kelas. Nilai *mAP* menggabungkan kinerja deteksi dari berbagai kategori dan menyajikan gambaran kemampuan model secara keseluruhan dalam mendeteksi berbagai jenis objek pada satu pengujian. Rumus (2.3) untuk *mAP* adalah sebagai berikut:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (2.3)$$

- a. N adalah jumlah total kelas objek yang ada dalam *dataset*
- b. AP_i adalah nilai *Average Precision* untuk kelas i

Metrik ini sangat berguna untuk mengevaluasi deteksi objek multi-kelas, terutama karena setiap kelas objek seringkali memiliki tingkat kesulitan deteksi yang berbeda. Dengan demikian, nilai *mAP* memberikan gambaran yang lebih adil mengenai kinerja model dalam menangani berbagai kelas objek secara bersamaan.

Precision dan *Recall* adalah dua metrik dasar yang digunakan untuk mengukur kinerja model deteksi objek. *Precision* mengukur proporsi prediksi positif yang benar dari total prediksi positif yang dibuat oleh model. Metrik ini berfokus pada akurasi prediksi yang dihasilkan. Di sisi lain, *recall* mengukur proporsi kasus positif aktual yang berhasil diidentifikasi dengan benar oleh model. Metrik ini berfokus pada kelengkapan deteksi. Perhitungan *precision* dihitung dengan rumus (2.4):

$$Precision = \frac{TP}{TP + FP} \quad (2.4)$$

Di mana:

- a. *True Positive* (TP) adalah jumlah sampel dari kelas positif yang diidentifikasi secara benar oleh model sebagai positif.
- b. *False Positive* (FP) adalah jumlah sampel dari kelas negatif yang diidentifikasi secara salah oleh model sebagai positif.

Sementara itu, *recall* dihitung dengan rumus (2.5):

$$Recall = \frac{TP}{TP + FN} \quad (2.5)$$

Di mana:

- a. *False Negative* (FN) adalah jumlah sampel dari kelas positif yang gagal diidentifikasi oleh model sebagai positif.

Kedua metrik ini penting karena memberikan sudut pandang yang saling melengkapi pada kinerja model. *Precision* yang tinggi menandakan bahwa model memiliki tingkat kesalahan deteksi yang rendah. Di sisi lain, *recall* yang tinggi menandakan bahwa model sangat efektif dalam menemukan hampir semua objek yang relevan dan memiliki tingkat objek terlewat yang rendah. Sebuah model yang ideal harus mampu menyeimbangkan keduanya, sehingga analisis gabungan terhadap *precision* dan *recall* menjadi standar dalam penilaian performa yang valid.

F1-Score adalah rata-rata harmonis (*harmonic mean*) antara *precision* dan *recall* yang memberikan gambaran seimbang mengenai kinerja model dalam mendeteksi objek. Metrik ini sangat berguna ketika terdapat ketidakseimbangan antara kelas positif dan negatif pada *dataset*. karena metrik ini memberikan bobot yang setara untuk *precision* dan *recall* dan mampu menghindari fokus berlebihan pada salah satu metrik saja. *F1-Score* memberikan rentang nilai antara 0 dan 1. Nilai 1 menunjukkan kinerja model yang sempurna dengan keseimbangan optimal antara *precision* dan *recall*, sedangkan nilai 0 menunjukkan kinerja terburuk. *F1-Score* dapat dihitung dengan rumus pada (2.6):

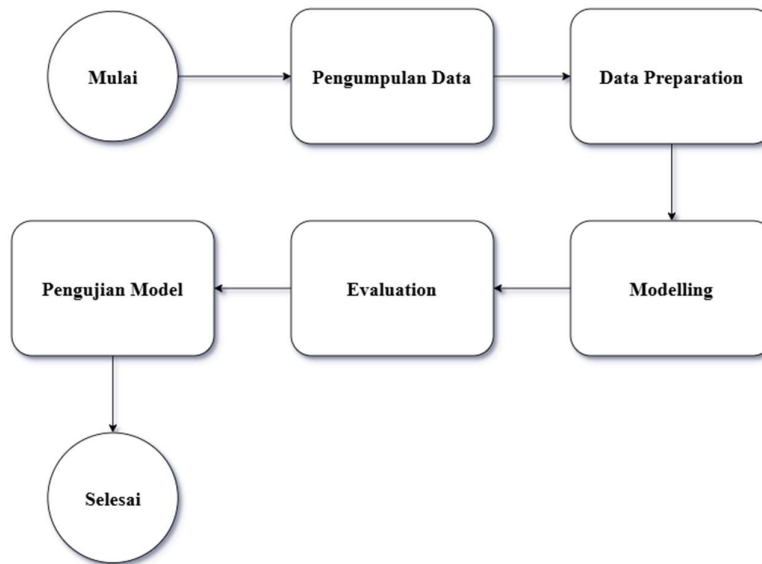
$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.6)$$

BAB III

METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Proses penelitian ini dilakukan melalui beberapa tahapan seperti pada Gambar 3.1 yang terdiri dari pengumpulan data, *data preparation*, *modelling*, *evaluation* dan pengujian model.



Gambar 3.1 Tahapan penelitian

3.2 Pengumpulan Data

Pada tahap ini dilakukan proses pengumpulan *dataset* berupa data citra yang diperoleh dengan cara mencari data gambar dengan kriteria gambar buah tomat yang tersedia di internet dan menggunakan platform Roboflow guna menambah *dataset* agar lebih bervariasi. Total *dataset* mencakup 1088 data citra tomat dengan berbagai tingkat kematangan tomat. Pada Gambar 3.2 merupakan contoh data citra yang akan digunakan untuk deteksi objek pada penelitian ini.



Gambar 3.2 Contoh Gambar Tomat pada *Dataset*



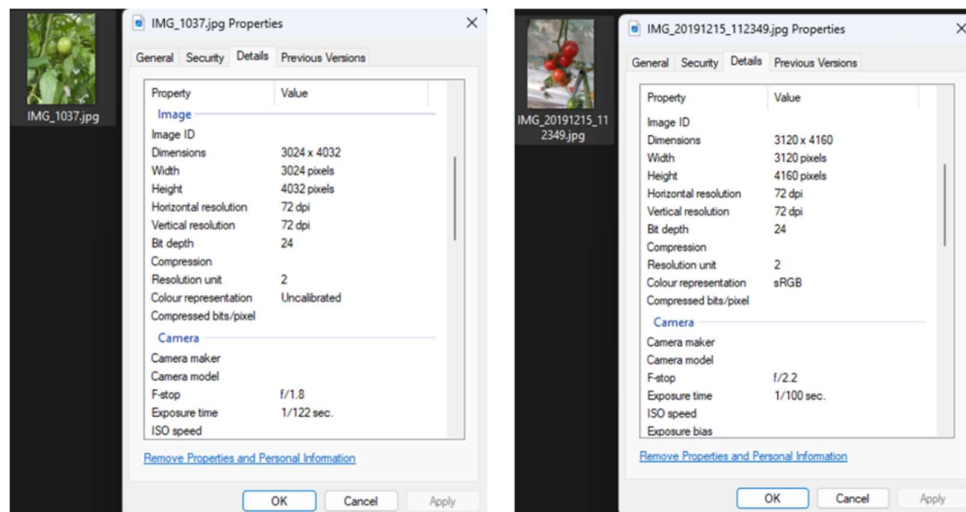
Mentah

Setengah Matang

Matang

Gambar 3.3 Kategori Kematangan Tomat

Gambar 3.4 merupakan contoh keragaman resolusi pada *dataset* citra yang digunakan, dengan contoh resolusi mencakup 3024×4032 piksel dan citra lainnya menunjukkan 3120×4160 piksel. Keragaman resolusi ini merupakan hasil dari proses pengumpulan data citra dengan cara yang memang direncanakan guna menciptakan *dataset* yang kaya akan variasi visual dan latar belakang objek.



Gambar 3.4 Ragam resolusi data pada gambar

3.3 Data Preparation

Tahap persiapan data (*Data Preparation*) merupakan fase yang penting pada penelitian ini karena bertujuan untuk mengubah dan mempersiapkan *dataset* citra buah tomat yang telah dikumpulkan menjadi data yang siap, terstruktur dan optimal untuk digunakan dalam proses pelatihan deteksi objek menggunakan YOLO11. Data yang tidak konsisten atau bervariasi dapat menghambat kemampuan model untuk belajar secara efektif. Untuk mengatasi hal tersebut, tahapan ini dibagi menjadi dua langkah yang akan dilakukan yaitu *preprocessing* untuk menstandarkan *dataset* serta augmentasi data untuk meningkatkan variasi dan jumlah data guna mencegah terjadinya *overfitting*.

3.3.1 Preprocessing

Proses *preprocessing* yang dilakukan pada data citra meliputi beberapa langkah dengan memanfaatkan platform Roboflow, yaitu Pelabelan dengan anotasi berupa *bounding box* pada objek, pembagian *dataset* menjadi beberapa bagian, penggunaan *Auto Orient* dan penyesuaian resolusi citra (*resize*).

Proses pelabelan objek pada keseluruhan *dataset* citra tomat dilakukan secara manual oleh peneliti dengan dua pedoman utama. Pedoman pertama adalah pemberian kotak pembatas (*bounding box*) secara presisi pada setiap tomat yang tampak pada citra seperti pada Gambar 3.5. Untuk setiap tomat yang terdeteksi, dibuat anotasi berupa kotak pembatas yang benar-benar mengikuti bentuk dan ukuran objek sehingga mencakup seluruh area visual tomat secara

akurat. Pemberian kotak pembatas dengan presisi ini penting untuk meminimalkan tertangkapnya area latar belakang yang tidak relevan atau bagian lain dari tanaman tomat seperti daun atau maupun batang.



Gambar 3.5 Sebelum dan Sesudah Pelabelan Objek

Pedoman kedua adalah klasifikasi tingkat kematangan yang dibagi menjadi tiga kategori, yaitu mentah, setengah matang dan matang. Contoh visual untuk masing-masing kategori ini telah disajikan sebelumnya pada Gambar 3.3. Untuk memastikan validitas dan objektivitas, kriteria visual untuk setiap kategori tersebut ditetapkan melalui validasi oleh praktisi pertanian di wilayah kecamatan Turi, Sleman, Yogyakarta, yaitu Ibu Maryuti seperti pada Gambar 3.6 yang memiliki pengalaman sebagai petani tomat selama kurang lebih 10 tahun. Keterlibatan narasumber yang kompeten ini bertujuan untuk memastikan bahwa kriteria yang digunakan sesuai dengan standar praktis di lapangan.



Gambar 3.6 Dokumentasi Penilaian Petani

Dengan kriteria yang telah tervalidasi tersebut, setiap objek tomat pada *dataset* diberikan label sesuai kelas kematangannya di Roboflow seperti pada Gambar 3.7. Penerapan kriteria ini secara konsisten pada seluruh *dataset* untuk memastikan hasil yang akurat untuk tahap pelatihan model.

Classes & Tags

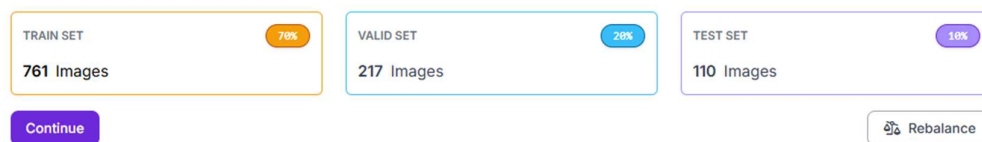
Classes 3

Tags 0

COLOR	CLASS NAME
	Matang
	Mentah
	Setengah Matang

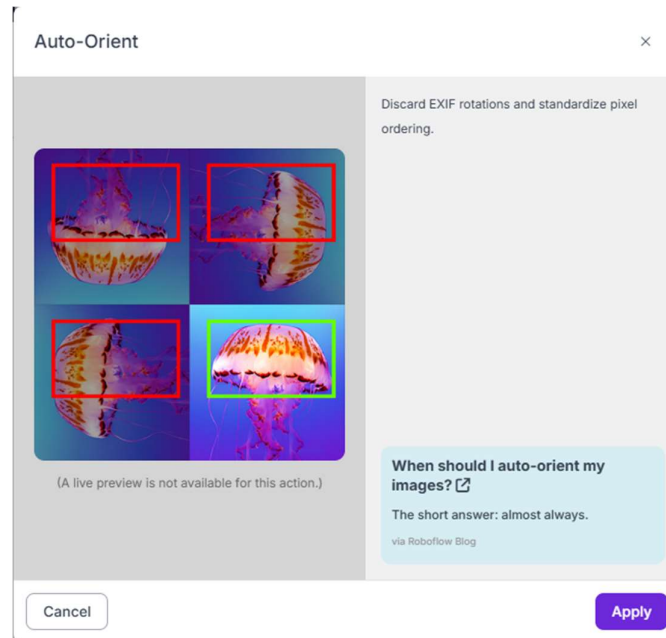
Gambar 3.7 Tiga Kelas Kematangan Tomat

Kemudian, tahapan pembagian *dataset* dikelompokkan menjadi tiga *subset* yang berbeda, yaitu *train*, *valid* dan *test* dengan perbandingan 70% *train* : 20% *valid* : 10% *test* pada masing-masing bagian. Dalam pengembangan model *machine learning*, pembagian data dapat dilakukan dalam beberapa skenario, tetapi pada umumnya jika menggunakan *dataset* yang kecil umumnya menggunakan rasio 70 : 20 : 10 sudah cukup optimal (Marpaung et al., 2022). Dalam penelitian ini, pembagian *dataset* sebesar 70% set *train* ditetapkan untuk memastikan model memiliki cukup informasi guna mempelajari pola dan fitur secara menyeluruh dan esensial untuk memitigasi *underfitting*. 20% set *valid* digunakan selama proses pelatihan untuk menyempurnakan *hyperparameter* model dan mencegah *overfitting*. 10% set *test* merupakan data uji yang sama sekali belum pernah dilihat oleh model selama pelatihan maupun validasi. *Set test* memiliki fungsi sebagai tolak ukur akhir untuk mengevaluasi performa model secara objektif seperti berhadapan dengan data di dunia nyata. Dengan demikian, komposisi 70% : 20% : 10% ini dirancang untuk menyeimbangkan kebutuhan pelatihan yang efektif, validasi untuk optimalisasi dan pencegahan *overfitting* serta evaluasi akhir untuk melihat kemampuan yang sebenarnya. Tahapan pembagian *dataset* dapat dilihat pada Gambar 3.8.



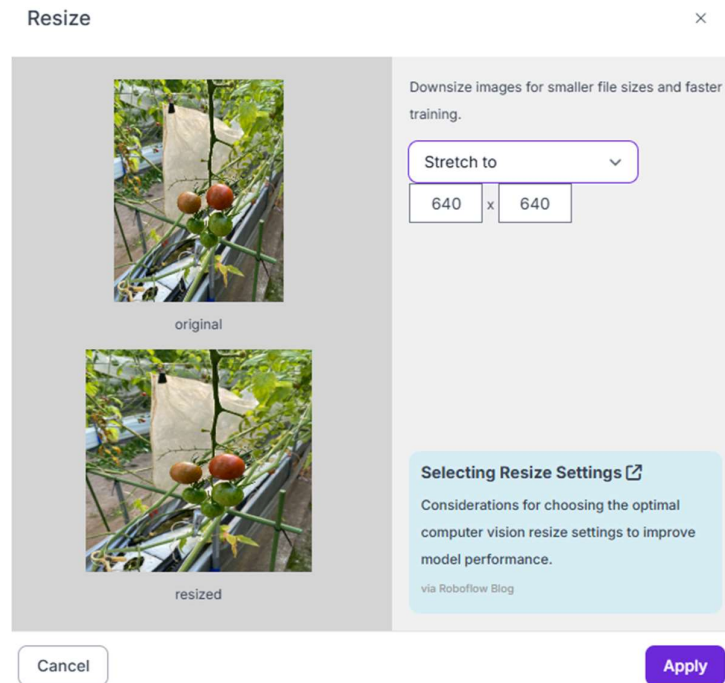
Gambar 3.8 Pembagian *dataset*

Auto Orient digunakan untuk menangani perbedaan orientasi objek pada citra dalam *dataset* penelitian. Fitur ini secara otomatis mengatur orientasi objek dalam *dataset*, sehingga menciptakan konsistensi dan akurasi penempatan posisi objek. Tahapan ini dapat dilihat pada Gambar 3.9.



Gambar 3.9 *Auto Orient*

Tahapan *resize* dilakukan untuk menyesuaikan ukuran resolusi menjadi 640×640 piksel menggunakan platform Roboflow. Perubahan ini bertujuan untuk menyeragamkan resolusi pada seluruh data citra yang telah dikumpulkan, seperti pada Gambar 3.10. Ukuran ini dipilih karena ukuran ini optimal dalam mempertahankan detail penting sekaligus meminimalkan beban komputasi, sehingga dapat meningkatkan efisiensi untuk melatih model. Selain itu, piksel yang seragam memungkinkan proses yang lebih konsisten dan cepat karena tidak perlu menyesuaikan setiap ukuran citra yang berbeda-beda.

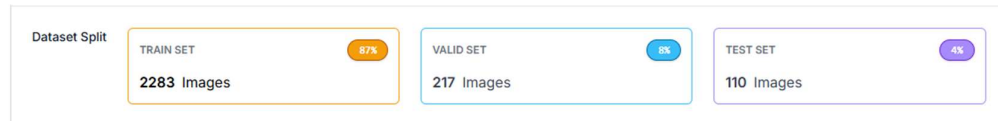


Gambar 3.10 Mengubah Resolusi pada Citra (*Resize*)

3.3.2 Augmentasi Data

Tahapan Augmentasi yang ditambahkan pada data citra setelah melakukan *preprocessing* berupa *Horizontal Flip*, *Rotation* antara -15° dan $+15^\circ$, *Shear* $\pm 15^\circ$ horizontal dan vertikal. Augmentasi ini bertujuan untuk meningkat variabilitas *dataset* dan memperkuat model untuk dijalankan menghadapi variasi visual yang kompleks secara *real-time*.

Setelah dilakukan proses augmentasi pada *dataset* yang telah dikumpulkan, total dari *dataset* yang akan digunakan untuk pelatihan menjadi 2610 data citra dengan penambahan data citra tanpa anotasi yang berguna untuk melatih model dalam mengenali apa yang bukan objek target seperti latar belakang. Perbandingan *dataset* menjadi 87% *train* : 8% *valid* : 4% *test* untuk masing-masing bagian. Perbandingan *dataset* dapat dilihat pada Gambar 3.11. Penambahan data melalui augmentasi hanya difokuskan pada *subset train*. Berdasarkan informasi pada platform Roboflow yang menyatakan bahwa augmentasi pada *subset valid* dan *test* dapat membebani kinerja model.

Gambar 3.11 Total *dataset*

3.4 Modelling

Tahapan modelling memiliki fokus utama yaitu mengembangkan dan melatih model berdasarkan dari *dataset* yang telah disiapkan sebelumnya. Dalam penelitian ini, arsitektur model yang digunakan adalah YOLO11 yang dikembangkan oleh Ultralytics. Pemilihan ini didasarkan pada performa yang ditawarkan untuk deteksi objek secara *real-time*. Performa yang ditawarkan tersebut merupakan hasil penerapan inovasi terkini pada *Deep Learning* sehingga menjadikannya pilihan yang serbaguna untuk berbagai tugas visual.

Sejalan dengan tujuan penelitian untuk mendapatkan kinerja deteksi tingkat kematangan buah tomat yang optimal, pada tahap ini akan dilakukan pelatihan pada varian model YOLO11 seperti varian n (*nano*), s (*small*), m (*medium*) dan l (*large*). Peneliti tidak menggunakan varian x (*extra-large*) dikarenakan kurang cocok untuk mayoritas deteksi objek yang memerlukan performa *real-time* dan berjalan pada *hardware* yang umum dan terbatas. Setiap varian model dilatih menggunakan *dataset* buah tomat yang telah melalui beberapa tahapan sebagaimana yang telah diuraikan pada bagian sebelumnya.

3.4.1 Proses Pelatihan Model

Proses pelatihan model untuk memperoleh hasil dilakukan menggunakan platform *Google Colab* dengan memanfaatkan *GPU* untuk mempercepat proses komputasi. Penelitian ini menggunakan *GPU NVIDIA A100-SXM4-40GB* dengan memori 40GB. Informasi *GPU* dapat dilihat pada Gambar 3.12.

NVIDIA-SMI 550.54.15			Driver Version: 550.54.15			CUDA Version: 12.4		
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC	
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	MIG M.	
0	NVIDIA A100-SXM4-40GB	Off	00000000:00:04.0	Off	0%	0	Default	
N/A	30C	P0	42W / 400W	0MiB / 40960MiB			Disabled	
Processes:								
GPU	GI	CI	PID	Type	Process name	GPU Memory	Usage	
	ID	ID						
No running processes found								

Gambar 3.12 Informasi GPU yang digunakan

Konfigurasi dilakukan pada beberapa *hyperparameter* sebelum melakukan pelatihan guna mengoptimalkan kinerja model. Konfigurasi tersebut seperti yang terlihat pada Tabel 3.1

Tabel 3.1 Konfigurasi *Hyperparameter* Pelatihan Model

Parameter	Nilai
<i>Batch-size</i>	16
<i>Epochs</i>	100
<i>Image-Size</i>	640×640
<i>Learning Rate (LR)</i>	0.01
<i>Momentum</i>	0.937
<i>Warmup Epochs</i>	3.0
<i>Weight Decay</i>	0.0005
<i>Optimizer</i>	SGD

Proses pelatihan dilakukan dengan konfigurasi parameter yang telah ditentukan. Parameter tersebut mencakup penggunaan *batch* sebesar 16, *epochs* 100 dan ukuran citra *input* (*image-size*) 640×640 piksel. *Learning rate* (LR) awal diatur pada 0.01 dengan *momentum* sebesar 0.937. Demi menjaga kestabilan gradien saat pelatihan dimulai, diterapkan periode *warmup* selama 3 *epochs* dengan nilai *weight decay* sebesar 0.0005 guna mencegah *overfitting*. Kemudian *optimizer* yang digunakan adalah *Stochastic Gradient Descent* (SGD).

3.5 Evaluation

Setelah menyelesaikan pelatihan dan konfigurasi pada varian model YOLO11, tahap selanjutnya adalah melakukan evaluasi performa pada model. Tahap evaluasi ini memiliki tujuan untuk mengukur kemampuan setiap model yang telah dilatih dalam menjalankan tugas deteksi tingkat kematangan buah tomat. Proses evaluasi ini dilakukan menggunakan *dataset valid* yang telah dipisahkan untuk memastikan objektivitas penilaian dan validitas hasil evaluasi. Hasil dari evaluasi ini akan menjadi landasan utama dalam menentukan varian model YOLO11 yang menunjukkan kinerja paling optimal untuk selanjutnya diuji secara komprehensif pada tahap pengujian model.

Pengukuran kinerja model akan dilakukan secara menyeluruh dengan menggunakan beberapa metrik umum yang digunakan untuk evaluasi model deteksi objek. Metrik pertama adalah *precision* yang digunakan untuk menilai tingkat keakuratan dari setiap prediksi yang dihasilkan pada model. Dalam penelitian ini, *precision* mengukur seberapa banyak dari objek yang terdeteksi sebagai buah tomat dengan tingkat kematangan yang telah ditentukan sudah sesuai dengan labelnya. Metrik kedua yaitu *recall* yang bertugas menilai kemampuan model dalam mengenali kembali semua objek buah tomat yang relevan dan terdapat pada citra. *Recall* yang memiliki nilai tinggi menandakan bahwa model mampu menemukan sebagian besar buah tomat sebagai target tanpa banyak yang terlewat.

Sebagai metrik evaluasi utama yang menyimpulkan kinerja model secara keseluruhan, penelitian ini akan menggunakan *mAP50*. Metrik ini dipilih karena kemampuannya untuk memberikan gambaran performa yang komprehensif dengan menyeimbangkan *precision* dan *recall*. Istilah *mAP50* merujuk pada perhitungan nilai rata-rata dari *Average Precision* (AP) yang diperoleh pada semua kelas kematangan yang telah ditentukan dengan menerapkan ambang batas *Intersection over Union* (IoU) sebesar 0.5. Ambang batas IoU 0.5 menandakan bahwa sebuah deteksi dianggap benar jika *bounding box* pada hasil prediksi memiliki area tumpang tindih minimal 50% dengan *bounding box* yang sebenarnya (*ground truth*). Kemudian, varian model yang berhasil mencapai nilai *mAP50* tertinggi akan dianggap sebagai model dengan keseimbangan deteksi terbaik. Berdasarkan hasil evaluasi dari metrik-metrik tersebut, maka akan dipilih satu varian model yang menunjukkan kinerja paling unggul dan kemudian dilakukan tahap pengujian model.

3.6 Pengujian Model

Setelah model terbaik dipilih berdasarkan evaluasi metrik yang ditetapkan, tahap selanjutnya adalah pengujian model. Pengujian ini memiliki tujuan utama untuk menilai performa model pada data yang sepenuhnya baru dan belum pernah dilihat sebelumnya. Secara spesifik, tahap ini berfungsi untuk melihat apakah model yang telah dikembangkan memenuhi tujuan penelitian dalam mendeteksi tingkat kematangan buah tomat menggunakan model YOLO11. *Dataset* yang akan digunakan untuk pengujian adalah data citra yang terletak pada *subset test* yang telah ditentukan sebelumnya.

Kemudian, pengujian berikutnya menggunakan *input* video secara langsung dari *webcam*. *Webcam* pengujian model menggunakan “MTECH WB500” yang memiliki resolusi 1080p/30 *fps*. Pengujian ini dirancang untuk mengevaluasi performa model dalam lingkungan yang lebih kompleks. Melalui pengujian menggunakan *webcam*, akan dilihat performa secara *real-time* dengan model yang telah dipilih seperti kecepatan inferensi yang diukur dalam *frames per second* (FPS) dan stabilitas dengan kelas yang berbeda. Kombinasi hasil dari evaluasi pada *subset test* dan pengujian menggunakan *webcam* memberikan penilaian secara keseluruhan mengenai performa model. Analisis ini bertujuan untuk mengukur kemampuan operasional, mengidentifikasi batasan serta menilai kelayakan model pada sistem deteksi kematangan tomat.

BAB IV

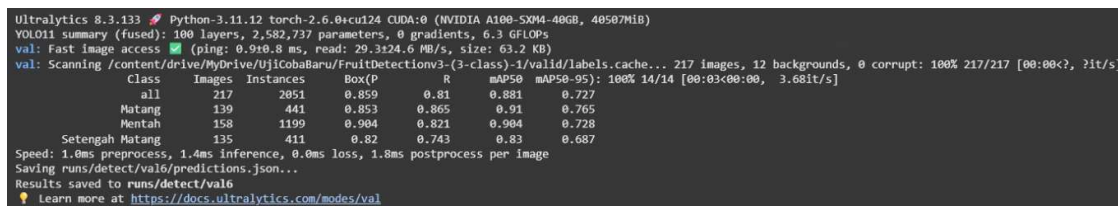
HASIL DAN PEMBAHASAN

4.1 Hasil Pelatihan Model

Pada bab ini akan memaparkan mengenai hasil dari pelatihan model yang telah dikembangkan untuk mendeteksi tingkat kematangan buah tomat yang dibagi menjadi tiga kategori yaitu mentah, setengah matang dan matang. Hasil performa model dilakukan menggunakan metrik dan kurva yang dihasilkan dari proses validasi menggunakan *framework* Ultralytics. Analisis pada metrik dan kurva tersebut akan menjelaskan sejauh mana performa model dalam mencapai tujuannya.

4.1.1 YOLO11n

Hasil model YOLO11n menunjukkan potensi yang baik berdasarkan hasil yang diperoleh. Namun, memerlukan optimalisasi lebih lanjut khususnya pada akurasi deteksi kelas “Setengah Matang” dan meminimalkan deteksi yang keliru pada latar belakang. Hasil model menggunakan validasi dapat dilihat pada Gambar 4.1 dan hasil visualisasi pelatihan dan validasi dapat dilihat pada Gambar 4.2.



```

Ultralytics 8.3.133 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (NVIDIA A100-SXM4-40GB, 40507MiB)
YOLO11 summary (fused): 100 layers, 2,582,737 parameters, 0 gradients, 6.3 GFLOPs
val: Fast Image access (ping: 0.910.8 ms, read: 20.3124.6 MB/s, size: 63.2 KB)
val: Scanning /content/drive/MyDrive/UjiCobaBaru/FruitDetectionv3-(3-class)-1/valid/labels.cache... 217 images, 12 backgrounds, 0 corrupt: 100% 217/217 [00:00<?, ?it/s]

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95
all	217	2951	0.859	0.81	0.881	0.727
Matang	139	441	0.853	0.865	0.91	0.765
Mentah	158	1199	0.904	0.821	0.904	0.728
Setengah Matang	135	411	0.82	0.743	0.83	0.687

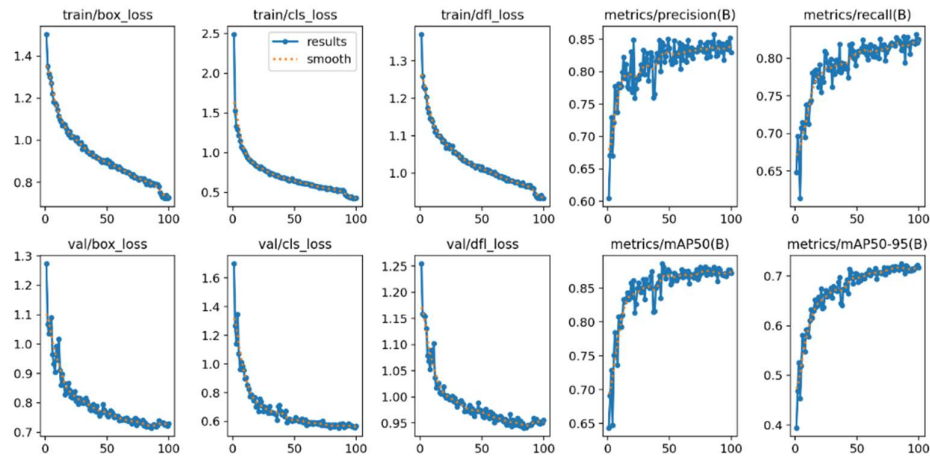
```

Speed: 1.0ms preprocess, 1.4ms inference, 0.0ms loss, 1.8ms postprocess per image
Saving runs/detect/val6/predictions.json...
Results saved to runs/detect/val6
Learn more at https://docs.ultralytics.com/modes/val

```

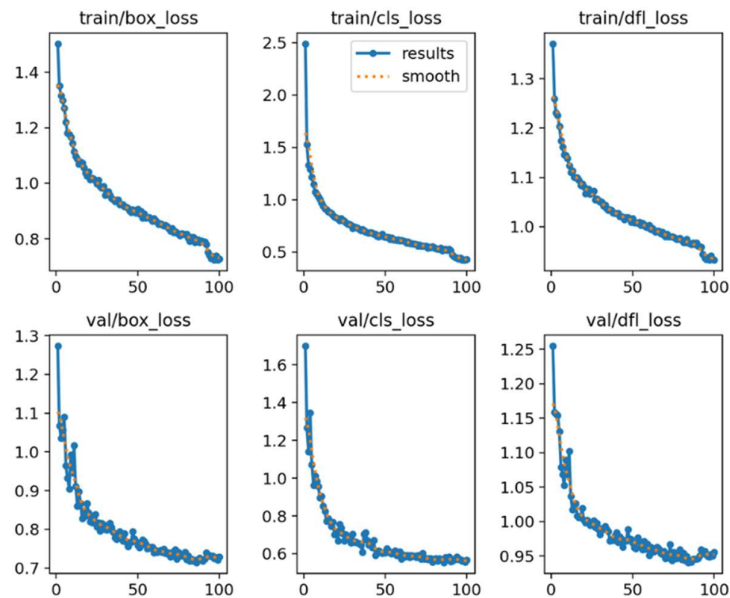
Gambar 4.1 Hasil Validasi YOLO11n

Gambar 4.1 menyajikan rangkuman hasil validasi model yang menampilkan metrik kinerja pada semua kelas yang telah ditetapkan sebelumnya. Pada varian model YOLO11n, model terdiri dari 100 layer dengan total 2.582.737 parameter dan kebutuhan komputasi sebesar 6.3 GFLOPs yang memungkinkan kecepatan inferensi rata-rata sekitar 1.4 ms per gambar. Secara keseluruhan, model menunjukkan performa yang baik dengan nilai *precision* (P) sebesar 0.859, *recall* (R) 0.81 dan *mAP50* sebesar 0.881. Nilai *mAP50* yang tinggi ini menunjukkan bahwa model akurat dalam mendeteksi dan mengklasifikasikan objek pada ambang batas IoU 0.5.



Gambar 4.2 Hasil Visualisasi Pelatihan YOLO11n

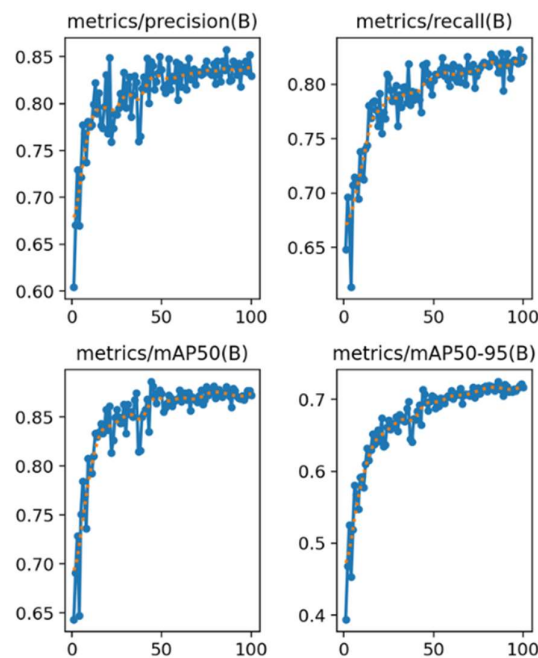
Berdasarkan hasil pelatihan dan validasi yang ditunjukkan pada Gambar 4.2 memperlihatkan tren peningkatan performa yang konsisten dan stabil selama 100 *epoch*. Nilai *loss* pada proses pelatihan yang mencakup *box loss*, *classification loss* dan *distribution focal loss* (DFL) menunjukkan penurunan seiring bertambahnya *epoch*.



Gambar 4.3 Tren penurunan *loss* YOLO11n

Pada Gambar 4.3 menunjukkan *box loss* berada pada ± 1.4 dan secara bertahap menurun hingga mencapai ± 0.7 pada akhir pelatihan. Penurunan ini mengindikasikan bahwa model semakin mampu memprediksi posisi *bounding box* dengan tingkat kesalahan yang rendah. *Classification loss* juga mengalami penurunan dari ± 2.5 menjadi ± 0.5 yang menunjukkan perbaikan substansial dalam kemampuan model membedakan kelas objek yang terdeteksi. Sementara itu, *DFL loss* menurun dari ± 1.3 menjadi ± 0.8 yang mencerminkan peningkatan presisi pada estimasi koordinat *bounding box* melalui distribusi prediksi yang lebih akurat.

Pada tahap validasi, tren memiliki pola serupa dengan hasil pelatihan yang menjadi indikasi bahwa model tidak hanya belajar dengan baik pada data latih, tetapi juga mampu mempertahankan performa tersebut pada data yang belum pernah dilihat sebelumnya. Nilai *box loss* pada data validasi berkurang dari ± 1.2 menjadi ± 0.7 , *classification loss* menurun dari ± 1.6 menjadi ± 0.5 dan *DFL loss* berkurang dari ± 1.25 menjadi ± 0.95 . Konsistensi penurunan *loss* pada kedua tahap tersebut menjadi bukti bahwa model tidak mengalami *overfitting* karena penurunan kesalahan tidak hanya terjadi pada data latih, tetapi juga pada data validasi.

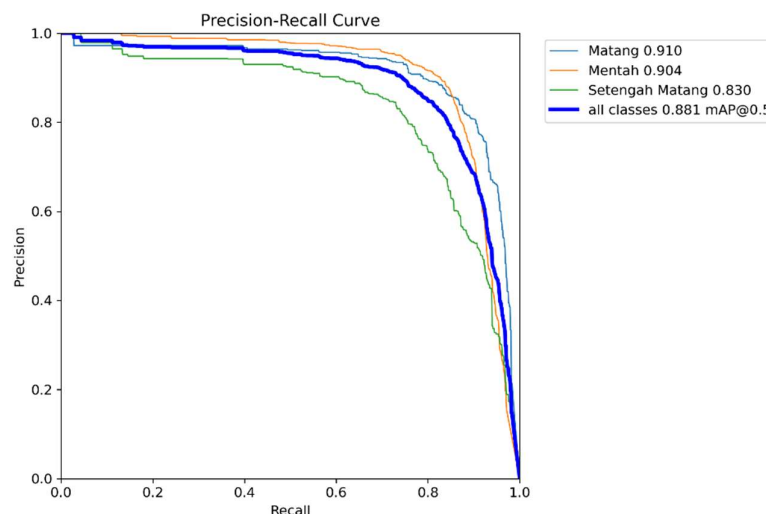


Gambar 4.4 Tren Peningkatan Kinerja YOLO11n

Gambar 4.4 menunjukkan metrik kinerja yang menunjukkan peningkatan. Nilai *precision* meningkat dari ± 0.60 menjadi ± 0.85 yang mengindikasikan bahwa model semakin jarang

memberikan prediksi positif yang keliru. Nilai *recall* meningkat dari ± 0.60 menjadi ± 0.82 yang menunjukkan kemampuan model mendeteksi objek yang relevan semakin baik dan jumlah *false positive* menurun. Peningkatan ini menandakan keseimbangan antara kemampuan model mengidentifikasi objek yang benar tanpa mengorbankan ketepatan prediksi. Selain itu, *mAP50* mengalami kenaikan dari ± 0.65 menjadi ± 0.88 yang menunjukkan bahwa model memiliki tingkat akurasi deteksi yang tinggi ketika diberikan toleransi posisi *bounding box* yang relatif longgar. Sementara itu, *mAP50-95* yang menguji performa model pada ambang batas IoU yang lebih ketat dan bervariasi meningkat dari ± 0.4 menjadi ± 0.7 yang menandakan bahwa model mampu mempertahankan akurasi deteksi pada berbagai tingkat kesulitan.

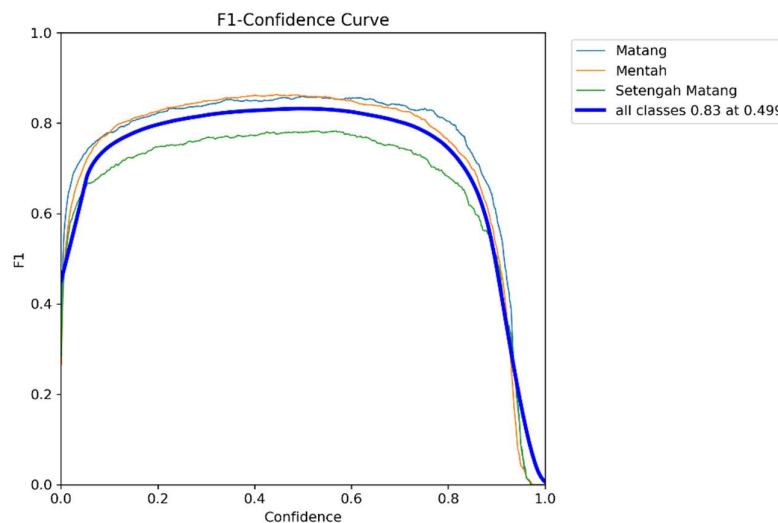
Secara keseluruhan, kombinasi penurunan *loss* yang konsisten, peningkatan metrik *precision* dan *recall* serta pencapaian nilai *mAP* yang tinggi pada berbagai ambang IoU membuktikan bahwa model YOLO11n yang dikembangkan memiliki performa deteksi objek yang andal. Model ini mampu melakukan generalisasi dengan baik, menghindari masalah *overfitting* dan mencapai keseimbangan optimal secara sensitivitas deteksi dan ketepatan deteksi.



Gambar 4.5 Kurva Hubungan *Precision* dan *Recall* YOLO11n

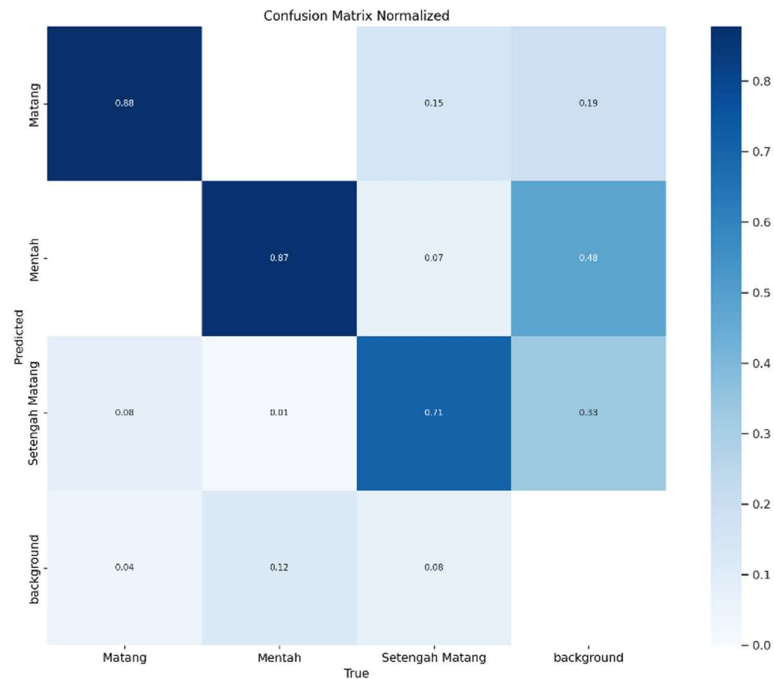
Pada Gambar 4.5 menyajikan kurva *precision-recall* yang berfungsi sebagai alat evaluasi untuk mengukur performa model yang telah dikembangkan. Kurva tersebut memvisualisasikan hubungan antara *precision* pada sumbu vertikal dan *recall* pada sumbu horizontal. Secara ideal, kurva yang mendekati sudut kanan atas menandakan performa model yang sangat baik karena

model tetap akurat bahkan saat harus menemukan lebih banyak target yang ada pada data. Pada grafik tersebut, diperlihatkan nilai AP pada setiap kelas dengan kelas matang menunjukkan performa terbaik dengan AP sebesar 0.910 dan terendah pada kelas setengah matang dengan AP sebesar 0.830. Perbedaan nilai AP ini mengindikasikan bahwa model memiliki tingkat keyakinan yang sedikit bervariasi dalam mengklasifikasikan setiap kelas. Untuk mengevaluasi kinerja model secara keseluruhan, digunakan metrik mAP yang merupakan rata-rata dari nilai AP pada semua kelas. Model berhasil mencapai nilai mAP sebesar 0.881.



Gambar 4.6 Kurva Hubungan $F1$ dan $Confidence$ YOLO11n

Gambar 4.6 menyajikan kurva $F1$ - $Confidence$ yang digunakan untuk mengevaluasi performa model pada berbagai tingkat kepercayaan (*confidence threshold*). Kurva tersebut memperlihatkan hubungan antara $F1$ -Score pada sumbu vertikal dengan nilai ambang batas kepercayaan pada sumbu horizontal yang ditetapkan untuk setiap prediksi. $F1$ -Score sendiri merupakan metrik yang menggabungkan *precision* dan *recall* untuk memberikan gambaran performa yang seimbang, sementara *confidence* mempresentasikan tingkat keyakinan atau *threshold* yang harus dilewati agar sebuah prediksi dianggap valid. Performa model secara keseluruhan mencapai titik optimalnya dengan $F1$ -Score puncak sebesar 0.83 pada saat *confidence threshold* berada di angka 0.449. Pada titik tersebut, model mampu mencapai keseimbangan terbaik antara *precision* dan *recall* secara gabungan untuk semua kelas sehingga menghasilkan performa yang optimal.



Gambar 4.7 *Confusion Matrix normalized YOLO11n*

Pada Gambar 4.7 menyajikan matriks yang membandingkan hasil prediksi model pada sumbu vertikal dengan label data sebenarnya pada sumbu horizontal. Nilai-nilai dalam matriks ini telah dinormalisasi sehingga menunjukkan proporsi prediksi yang benar dan salah untuk setiap kelas. Berdasarkan gambar tersebut, model menunjukkan performa sangat baik dalam pada kelas matang dan mentah dengan nilai masing-masing 88% dan 87%. Untuk kelas setengah matang, model menunjukkan nilai sebesar 71%.

4.1.2 YOLO11s

Hasil model varian YOLO11s menunjukkan performa menjanjikan, meskipun optimalisasi lebih lanjut diperlukan untuk meningkatkan akurasi deteksi kelas “Setengah Matang” dan meminimalkan kesalahan identifikasi latar belakang. Hasil validasi model dapat dilihat pada Gambar 4.8 dan hasil visualisasi pelatihan dan validasi terdapat pada Gambar 4.9.

```

Ultralytics 8.3.133 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (NVIDIA A100-SXM4-40GB, 40507MiB)
YOLO11s summary (fused): 100 layers, 9,413,961 parameters, 0 gradients, 21.3 GFLOPs
val: Fast image access (ping: 0.41±0.2 ms, read: 32.9±7.6 MB/s, size: 57.9 KB)
val: Scanning /content/drive/MyDrive/UjicobaBaru/FruitDetectionv3-(3-class)-1/valid/labels.cache... 217 images, 12 backgrounds, 0 corrupt: 100% 217/217 [00:00<?, ?it/s]

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95
all	217	2051	0.856	0.834	0.889	0.748
Matang	139	441	0.853	0.878	0.907	0.776
Mentah	158	1199	0.897	0.844	0.923	0.765
Setengah Matang	135	411	0.82	0.781	0.838	0.703

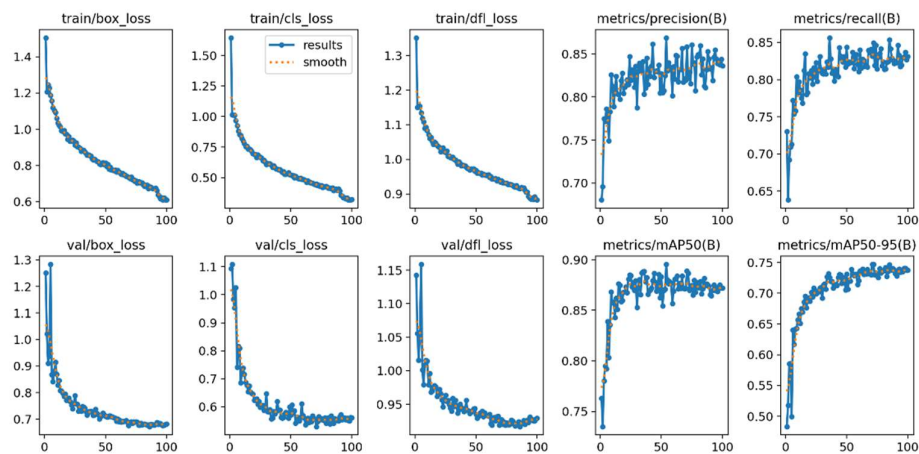
```

Speed: 1.0ms preprocess, 1.9ms inference, 0.0ms loss, 3.2ms postprocess per image
Saving runs/detect/val7/predictions.json...
Results saved to runs/detect/val7
Learn more at https://docs.ultralytics.com/modes/val

```

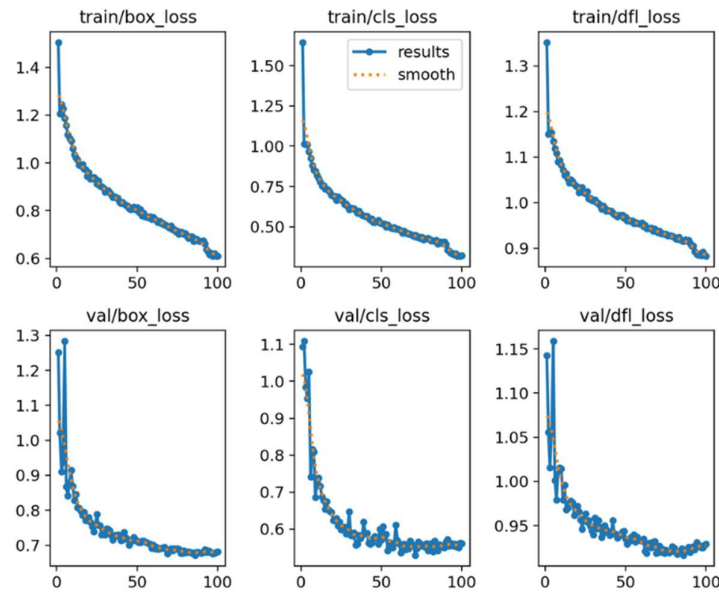
Gambar 4.8 Hasi Validasi YOLO11s

Gambar 4.8 memperlihatkan rangkuman hasil validasi model yang menampilkan metrik kinerja pada semua kelas yang telah ditetapkan sebelumnya. Model varian YOLO11s terdiri dari 100 layer dengan total 9.413.961 parameter dan memerlukan 21.3 GFLOPs dengan kecepatan inferensi rata-rata 1.9 ms per gambar. Secara keseluruhan, model menunjukkan performa yang baik dengan nilai *precision* (P) sebesar 0.856, *recall* (R) dengan 0.834 dan *mAP50* sebesar 0.889. Nilai *mAP50* yang dihasilkan pada YOLO11s lebih tinggi dari varian sebelumnya yaitu YOLO11n yang mengindikasikan bahwa model ini memiliki akurasi yang lebih unggul dalam mendeteksi serta mengklasifikasikan pada ambang batas IoU 0.5.



Gambar 4.9 Hasil Visualisasi Pelatihan YOLO11s

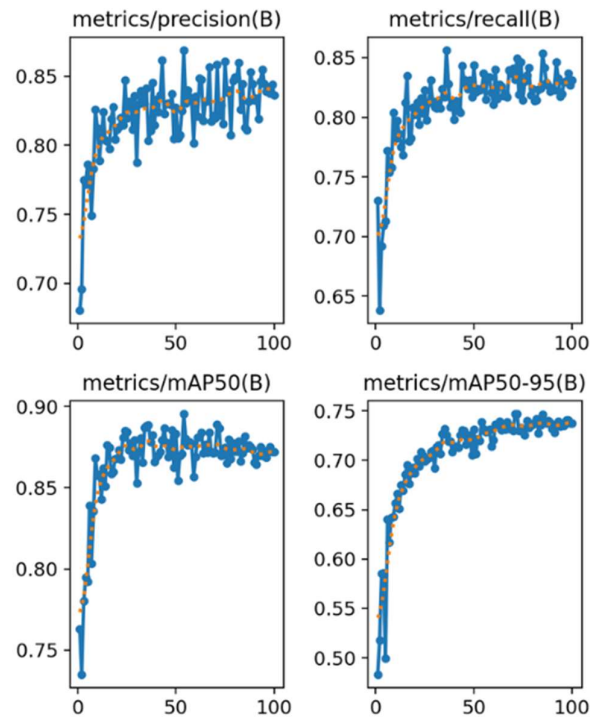
Berdasarkan hasil pelatihan dan validasi pada Gambar 4.9 memperlihatkan tren peningkatan performa sepanjang 100 *epoch*. Nilai *loss* pada proses pelatihan yang mencakup *box loss*, *classification loss* dan *DFL* menunjukkan penurunan seiring bertambahnya *epoch*.



Gambar 4.10 Tren penurunan *loss* YOLO11s

Pada Gambar 4.10, *box loss* berada pada ± 1.4 dan secara bertahap menurun hingga mencapai ± 0.6 pada akhir pelatihan. Penurunan ini mengindikasikan bahwa model mampu memprediksi posisi *bounding box* dengan tingkat kesalahan yang rendah. *Classification loss* juga mengalami penurunan tajam dari ± 1.55 menjadi ± 0.40 yang menunjukkan perbaikan dalam kemampuan model membedakan kelas objek yang terdeteksi. Sementara itu, *DFL loss* menurun dari ± 1.3 menjadi ± 0.8 yang mengindikasikan peningkatan presisi pada estimasi koordinat *bounding box* melalui distribusi prediksi yang lebih akurat.

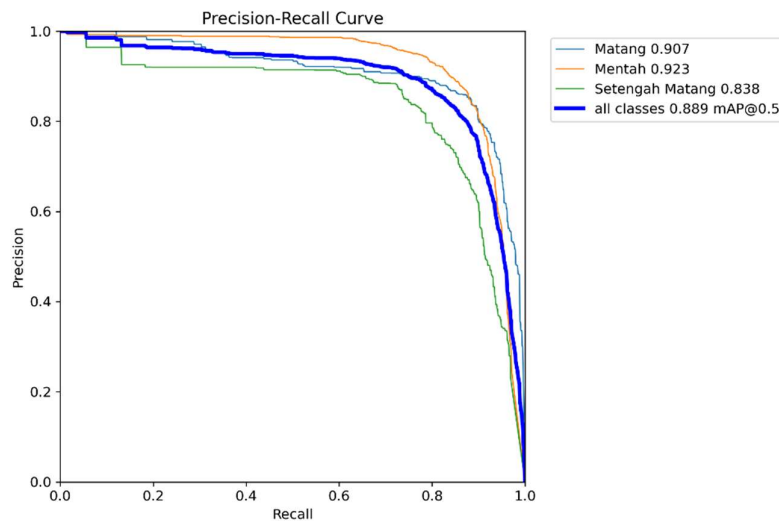
Pada tahap validasi, tren yang diamati memiliki pola serupa dengan hasil pelatihan yang menjadi indikasi bahwa model tidak hanya belajar dengan baik pada data latih, tetapi juga mampu mempertahankan performa tersebut pada data yang belum pernah dilihat sebelumnya. Nilai *box loss* pada data validasi berkurang dari ± 1.28 menjadi ± 0.68 . *classification loss* menurun dari ± 1.1 menjadi ± 0.5 dan *DFL loss* berkurang dari ± 1.15 menjadi ± 0.94 . Konsistensi penurunan *loss* pada kedua tahap ini menjadi bukti bahwa model tidak mengalami *overfitting* karena penurunan kesalahan tidak hanya terjadi pada data latih, tetapi juga terjadi pada data validasi.



Gambar 4.11 Tren Peningkatan Kinerja YOLO11s

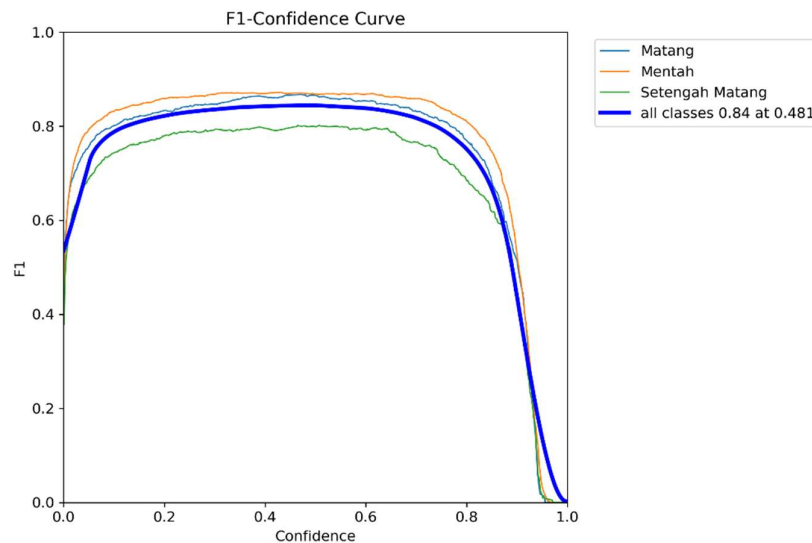
Pada Gambar 4.11, Evaluasi terhadap kinerja metrik kinerja menunjukkan peningkatan. Nilai *precision* meningkat dari ± 0.68 menjadi ± 0.85 yang berarti model semakin jarang memberikan prediksi positif yang keliru. Nilai *recall* meningkat dari ± 0.6 menjadi ± 0.83 yang mengindikasikan kemampuan model mendeteksi objek yang relevan semakin baik dan jumlah *false negative* menurun. Peningkatan ini menandakan keseimbangan antara kemampuan model mengidentifikasi objek yang benar tanpa mengorbankan ketepatan prediksi. Selain itu, *mAP50* mengalami kenaikan dari ± 0.74 menjadi ± 0.88 yang menunjukkan bahwa model memiliki tingkat akurasi deteksi yang tinggi ketika diberikan toleransi posisi *bounding box* yang relatif longgar. Sementara itu, *mAP50-95* yang menguji performa model pada ambang batas IoU yang lebih ketat dan bervariasi meningkat dari ± 0.50 menjadi ± 0.74 yang menandakan bahwa model mampu mempertahankan akurasi deteksi pada berbagai tingkat ketelitian.

Secara keseluruhan, kombinasi penurunan *loss* yang konsisten, peningkatan metrik *precision* dan *recall* serta pencapaian nilai *mAP* yang tinggi pada berbagai ambang IoU membuktikan bahwa model YOLO11s memiliki performa deteksi objek yang andal. Model ini mampu melakukan generalisasi dengan baik, menghindari masalah *overfitting* dan mencapai keseimbangan optimal antara sensitivitas deteksi dan ketepatan deteksi.



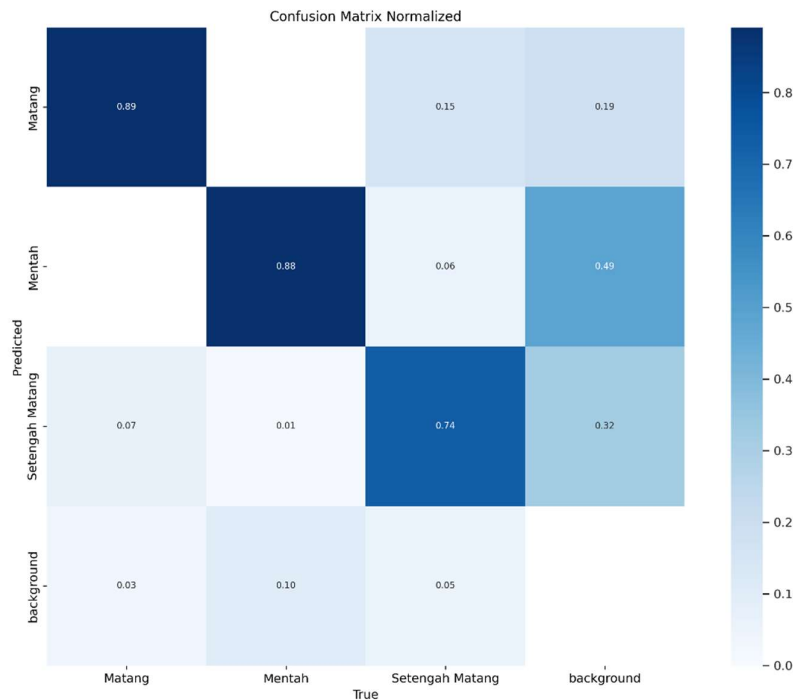
Gambar 4.12 Kurva Hubungan *Precision* dan *Recall* YOLO11s

Pada Gambar 4.12 memperlihatkan kurva *precision-recall* yang berfungsi sebagai alat evaluasi untuk mengukur performa model yang telah dikembangkan. Kurva tersebut memvisualisasikan hubungan antara *precision* pada sumbu vertikal dan *recall* pada sumbu horizontal. Secara ideal, kurva yang mendekati sudut kanan atas menandakan performa model yang sangat baik, karena model mampu mempertahankan *precision* yang tinggi dan meningkatkan jangkauan deteksi (*recall*). Pada hasil tersebut, kelas mentah menunjukkan performa terbaik dengan nilai AP sebesar 0.923, diikuti dengan kelas matang dengan nilai AP 0.907 dan performa terendah pada kelas setengah matang dengan nilai 0.838. Perbedaan nilai AP ini mengindikasikan bahwa model memiliki tingkat keyakinan yang bervariasi untuk setiap kelas. Untuk mengevaluasi kinerja model secara keseluruhan, digunakan metrik *mAP* dengan nilai yang dihasilkan sebesar 0.889. Nilai tersebut menunjukkan bahwa rata-rata model memiliki performa yang baik dalam mendeteksi semua kelas yang ada.



Gambar 4.13 Kurva Hubungan *F1* dan *Confidence* YOLO11s

Gambar 4.13 menunjukkan kurva skor *F1-Confidence* yang digunakan untuk mengevaluasi performa model pada berbagai tingkat kepercayaan (*confidence threshold*). Kurva tersebut memperlihatkan hubungan antara *F1-Score* pada sumbu vertikal dengan nilai ambang batas kepercayaan pada sumbu horizontal. Grafik tersebut menampilkan kurva untuk setiap kelas serta kurva gabungan untuk semua kelas yang ditandai dengan garis biru tebal. Berdasarkan data yang dihasilkan, performa model secara keseluruhan mencapai titik optimalnya dengan *F1-Score* puncak sebesar 0.84 pada saat *confidence threshold* berada di angka 0.481. Pada titik tersebut, model mampu mencapai keseimbangan terbaik antara *precision* dan *recall* pada gabungan untuk semua kelas sehingga menghasilkan performa klasifikasi yang paling optimal.



Gambar 4.14 *Confusion Matrix normalized YOLO11s*

Pada Gambar 4.14 menyajikan *confusion matrix normalized* yang membandingkan hasil prediksi model pada sumbu vertikal dengan label data sebenarnya pada sumbu horizontal. Berdasarkan matriks tersebut, model menunjukkan performa yang sangat baik dalam mengenali kelas matang dan mentah dengan nilai masing-masing sebesar 89% dan 88%. Untuk kelas setengah matang, model berhasil mengenali dengan benar sebanyak 74% dari total data kelas tersebut.

4.1.3 YOLO11m

Hasil model varian YOLO11m menunjukkan performa lebih baik dari dua varian model sebelumnya yaitu YOLO11n dan YOLO11s, meskipun optimalisasi lebih lanjut diperlukan untuk meningkatkan akurasi deteksi kelas “Setengah Matang” dan meminimalkan kesalahan yang keliru pada latar belakang. Hasil validasi model dapat dilihat pada Gambar 4.15 dan visualisasi pelatihan dan validasi terdapat pada Gambar 4.16.

```

Ultralytics 8.3.133 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (NVIDIA A100-SXM4-40GB, 40507MiB)
YOLO11m summary (fused): 125 layers, 20,832,345 parameters, 0 gradients, 67.7 GFLOPs
val: Fast image access (ping: 0.3±0.1 ms, read: 53.2±14.4 MB/s, size: 67.8 KB)
val: Scanning /content/drive/MyDrive/UjiCobaBaru/FruitDetectionv3-(3-class)-1/valid/labels.cache... 217 images, 12 backgrounds, 0 corrupt: 100% 217/217 [00:00<?, ?it/s]

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95
all	217	2051	0.859	0.841	0.891	0.766
Matang	139	441	0.833	0.914	0.911	0.793
Mentah	158	1199	0.909	0.847	0.921	0.781
Setengah Matang	135	411	0.837	0.761	0.84	0.724

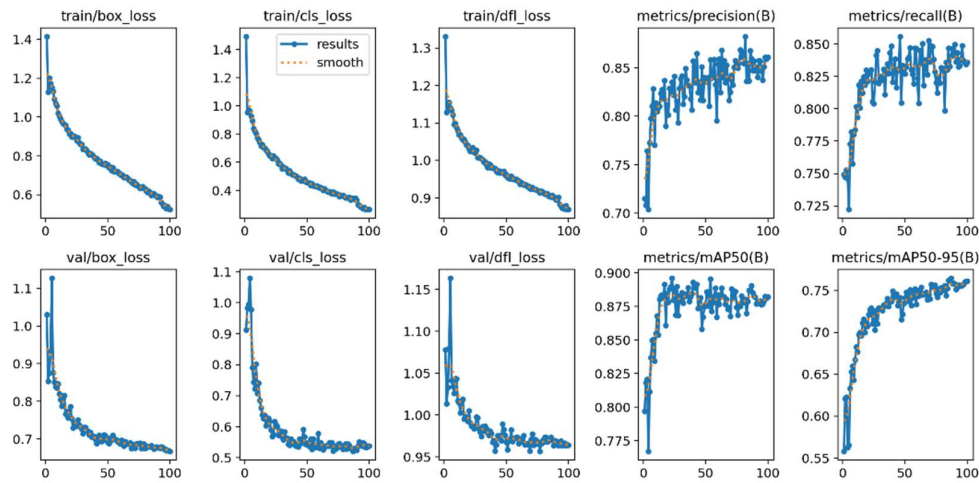
```

Speed: 1.0ms preprocess, 3.2ms inference, 0.0ms loss, 8.5ms postprocess per image
Saving runs/detect/val8/predictions.json...
Results saved to runs/detect/val8
Learn more at https://docs.ultralytics.com/modes/val

```

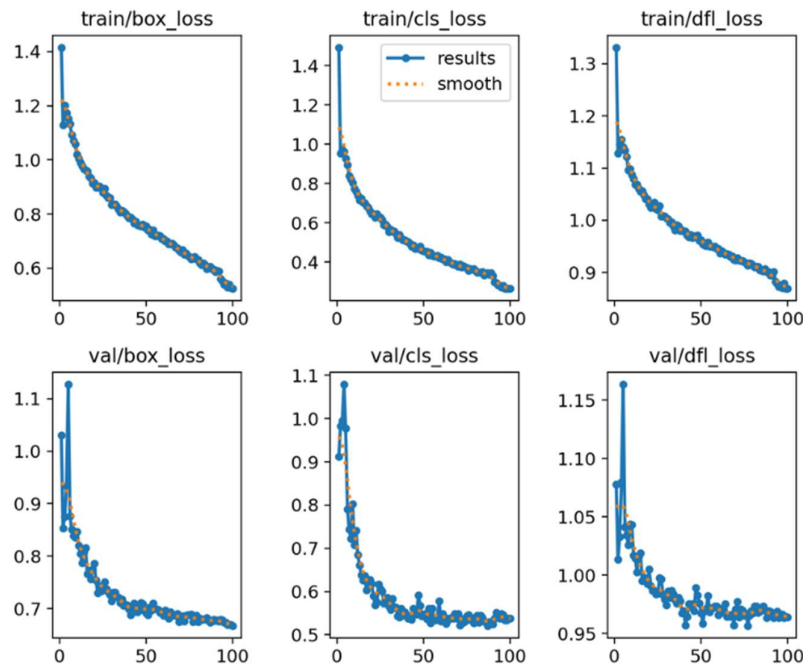
Gambar 4.15 Hasi Validasi YOLO11m

Gambar 4.15 menyajikan rangkuman hasil validasi model yang menampilkan metrik kinerja pada semua kelas yang telah ditetapkan sebelumnya. Pada varian model YOLO11m model terdiri dari 125 layer dengan total 20.032.345 parameter dan memerlukan kebutuhan komputasi sebesar 67.7 GFLOPs dengan kecepatan inferensi rata-rata 3.2 ms per gambar. Secara keseluruhan, model menunjukkan performa yang baik dengan nilai *precision* (P) sebesar 0.859, *recall* (R) dengan 0.841 dan *mAP50* sebesar 0.891. Nilai *mAP* yang dihasilkan pada YOLO11m mempunyai hasil yang lebih tinggi dari dua varian sebelumnya.



Gambar 4.16 Hasil Visualisasi Pelatihan YOLO11m

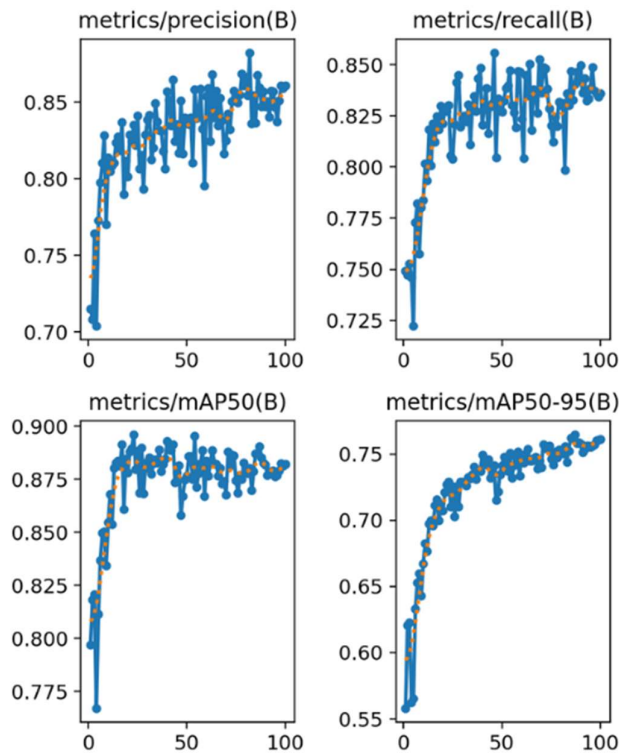
Berdasarkan hasil pelatihan dan validasi yang ditunjukkan pada Gambar 4.16 menunjukkan tren performa yang konsisten dan stabil selama 100 *epoch*. Nilai *loss* pada proses pelatihan yang terdiri dari *box loss*, *classification loss* dan *DFL loss* mengalami penurunan seiring berjalannya *epoch*.



Gambar 4.17 Tren penurunan *loss* YOLO11m

Pada Gambar 4.17, pelatihan awal pada *box loss* berada pada ± 1.4 dan menurun hingga ± 0.5 pada akhir pelatihan yang mengindikasikan bahwa model semakin akurat dalam memprediksi posisi *bounding box*. *Classification loss* turun dari ± 1.4 menjadi ± 0.3 yang menunjukkan peningkatan kemampuan model dalam membedakan kelas objek secara tepat. *DFL loss* juga mengalami penurunan dari ± 1.3 menjadi ± 0.8 yang mencerminkan peningkatan presisi pada prediksi koordinat *bounding box* melalui distribusi estimasi yang lebih baik.

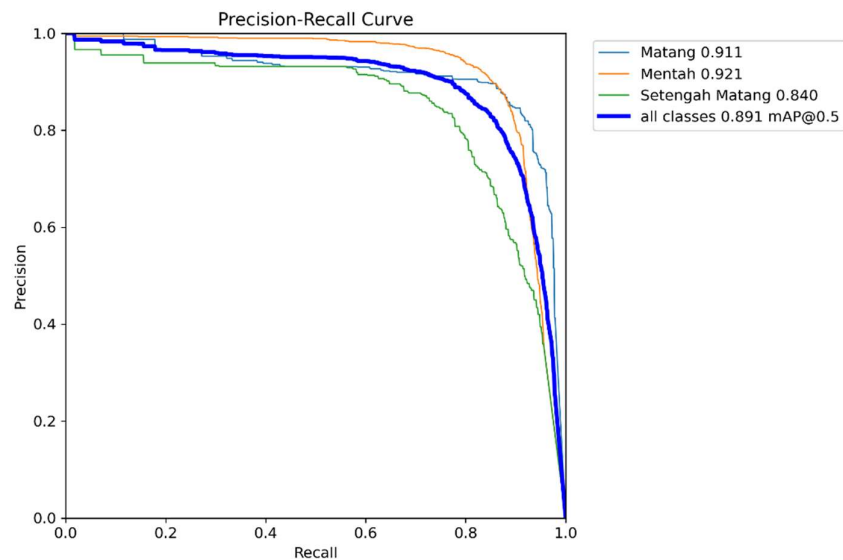
Pada tahap validasi, pola penurunan yang serupa juga terlihat. *Box loss* berkurang dari ± 1.1 menjadi ± 0.6 , *classification loss* menurun dari ± 1.08 menjadi ± 0.5 dan *DFL loss* berkurang dari ± 1.15 menjadi ± 0.97 . Konsistensi antara tren pada data latih dan data validasi ini menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami *overfitting*.



Gambar 4.18 Tren Peningkatan Kinerja YOLO11m

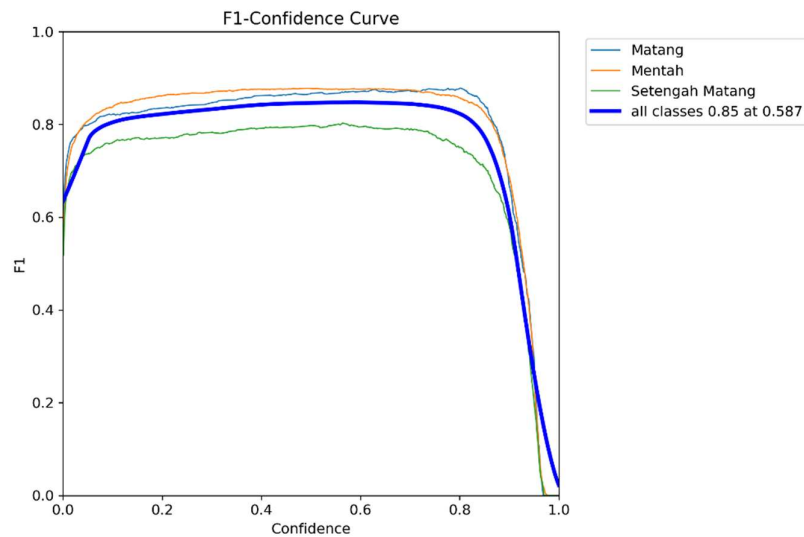
Pada Gambar 4.18, evaluasi metrik performa menunjukkan adanya peningkatan. Nilai *precision* meningkat dari ± 0.72 menjadi ± 0.85 yang berarti proporsi prediksi positif yang benar semakin tinggi dengan penurunan *false positive*. Nilai *recall* naik dari ± 0.725 menjadi ± 0.84 yang mengindikasikan bahwa model mampu mendeteksi sebagian besar objek ada dengan *false negative* yang semakin sedikit. Nilai *mAP50* mengalami peningkatan dari ± 0.775 menjadi ± 0.89 menandakan bahwa model memiliki tingkat akurasi tinggi dalam deteksi objek pada toleransi posisi *bounding box* yang cukup longgar. Sementara itu, *mAP50-95* yang mengukur akurasi pada rentang ambang IoU yang lebih ketat naik dari ± 0.55 menjadi ± 0.76 menunjukkan kemampuan model mempertahankan performa pada tingkat ketelitian yang lebih tinggi.

Secara keseluruhan, penurunan *loss* yang konsisten pada pelatihan dan validasi, peningkatan nilai *precision* dan *recall* serta tingginya nilai *mAP* pada berbagai tingkat ambang IoU menjadi bukti bahwa model YOLO11m yang dikembangkan memiliki tingkat kinerja deteksi objek yang andal. Model mampu mencapai keseimbangan yang baik antara sensitivitas deteksi dan ketepatan prediksi.



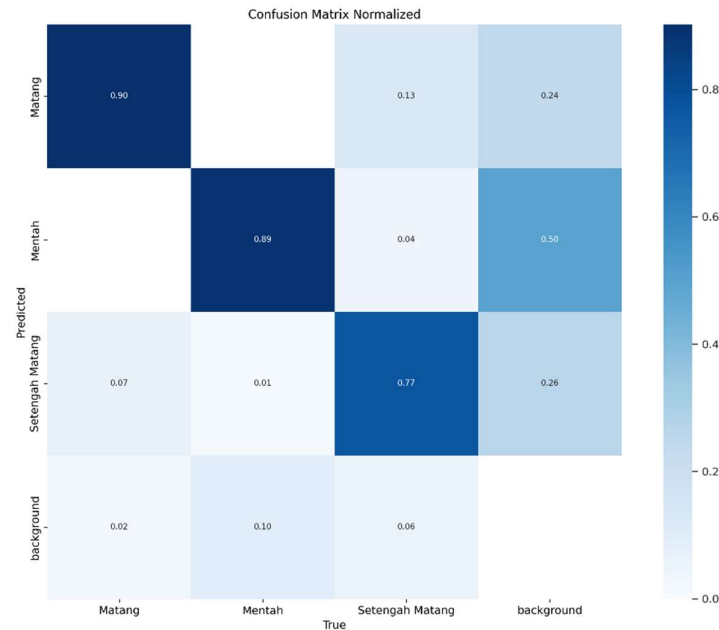
Gambar 4.19 Kurva Hubungan *Precision* dan *Recall* YOLO11m

Pada Gambar 4.19 memperlihatkan kurva *precision-recall* yang berfungsi sebagai alat evaluasi untuk mengukur performa model yang telah dikembangkan. Kurva tersebut memvisualisasikan hubungan antara *precision* pada sumbu vertikal dan *recall* pada sumbu horizontal. Hasil analisis menunjukkan nilai AP yang bervariasi untuk setiap kelas. Kelas mentah mencapai nilai AP tertinggi sebesar 0.921, diikuti oleh kelas matang dengan skor 0.911. Sementara itu, kelas setengah matang menunjukkan nilai AP terendah, yaitu 0.840. Perbedaan nilai ini menandakan kapabilitas model yang tidak seragam dalam mengenali setiap kelas. Secara keseluruhan, model varian ini menunjukkan performa yang tertinggi dari dua varian sebelumnya dengan nilai *mAP* mencapai 0.891. Nilai tersebut menegaskan bahwa model secara rata-rata memiliki kemampuan yang optimal dalam mendeteksi seluruh kelas yang telah ditentukan.



Gambar 4.20 Kurva Hubungan $F1$ dan $Confidence$ YOLO11m

Gambar 4.20 menyajikan kurva $F1$ - $Confidence$ untuk menilai performa model. Grafik tersebut mengilustrasikan hubungan antara $F1$ -Score dengan $confidence threshold$ yang ditetapkan. Di dalamnya, Visualisasi performa untuk masing-masing kelas dan performa gabungan untuk semua kelas yang ditandai dengan garis biru tebal. Berdasarkan kurva tersebut, teridentifikasi bahwa titik performa paling optimal dari model adalah 0.85 yang dicapai saat $confidence threshold$ pada nilai 0.587. Nilai tersebut menunjukkan titik keseimbangan paling efektif antara $precision$ dan $recall$ sehingga menghasilkan kemampuan yang optimal.



Gambar 4.21 *Confusion Matrix normalized YOLO11m*

Pada Gambar 4.21 disajikan *confusion matrix normalized* yang membandingkan hasil prediksi model pada sumbu vertikal dengan label data sebenarnya pada sumbu horizontal. Berdasarkan matriks tersebut, model menunjukkan performa yang sangat baik dalam mengenali kelas matang dengan nilai sebesar 90%. Untuk kelas mentah, model mampu mengenali dengan benar sebesar 89% dan untuk kelas setengah matang, model berhasil mengidentifikasi dengan benar sebanyak 77% dari total data pada kelas tersebut.

4.1.4 YOLO11l

Hasil model varian YOLO11l menunjukkan performa yang sedikit menurun dari varian model sebelumnya yaitu YOLO11m. Hasil validasi model dapat dilihat pada Gambar 4.22 dan visualisasi pelatihan dan validasi terdapat pada Gambar 4.23.

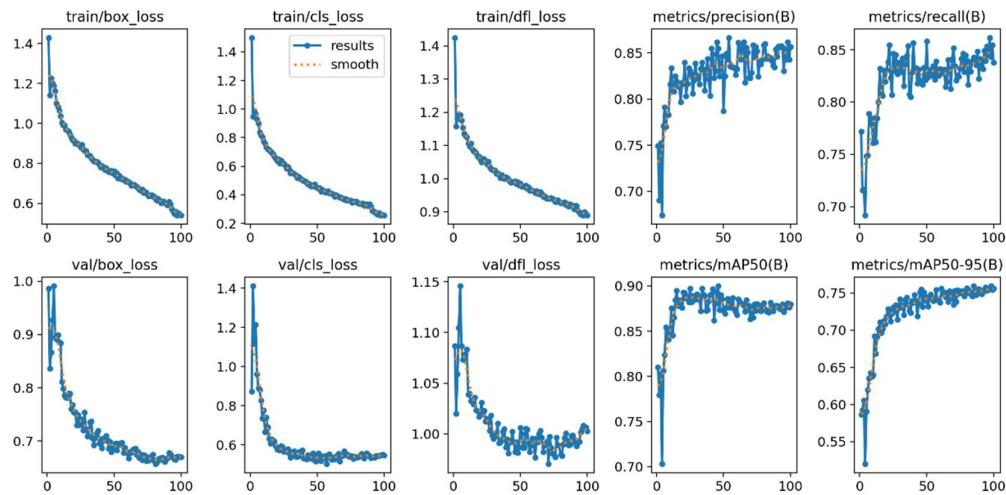
```

Ultralytics 8.3.133 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (NVIDIA A100-SXM4-40GB, 40507MiB)
YOLO11l summary (fused): 190 layers, 25,281,625 parameters, 0 gradients, 86.6 GFLOPs
val: Fast image access (ping: 0.318.1 ms, read: 52.018.5 MB/s, size: 61.9 KB)
val: Scanning /content/drive/MyDrive/UjicobaBaru/FruitDetectionv3-(3-class)-1/valid/labels.cache... 217 images, 12 backgrounds, 0 corrupt: 100% 217/217 [00:00<?, ?it/s]
Class Images Instances box(P) R mAP50 mAP50-95 100% 14/14 [00:04<00:00, 3.38it/s]
  all      217      2051    0.862    0.843    0.881    0.76
  Matang   139      441    0.828    0.914    0.882    0.769
  Mentah   158     1199    0.908    0.869    0.933    0.796
  Setengah Matang 135      411    0.85    0.746    0.83    0.716
Speed: 1.0ms preprocess, 4.0ms inference, 0.0ms loss, 8.5ms postprocess per image
Saving runs/detect/val9/predictions.json...
Results saved to runs/detect/val9
Learn more at https://docs.ultralytics.com/modes/val

```

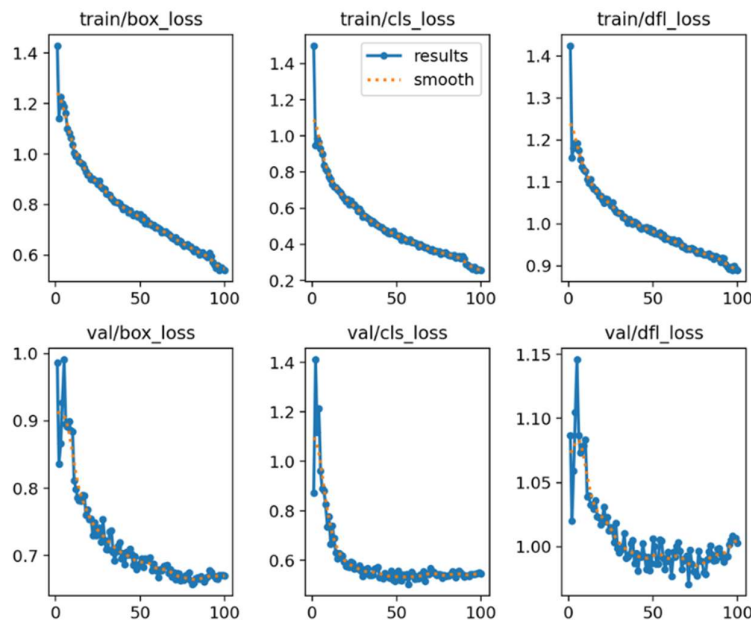
Gambar 4.22 Hasi Validasi YOLO11l

Gambar 4.22 memperlihatkan rangkuman hasil validasi model yang menampilkan metrik kinerja pada semua kelas yang telah ditetapkan sebelumnya. Model varian YOLO11l terdiri dari 190 layer dengan total 25.281.625 parameter dan memerlukan 86.6 GFLOPs dengan kecepatan inferensi rata-rata 4.0 ms per gambar. Secara keseluruhan, model menunjukkan nilai *precision* (P) sebesar 0.862, *recall* (R) dengan nilai 0.843 dan *mAP* 0.88.1. nilai *mAP* yang dihasilkan pada YOLO11l menurun dari varian YOLO11m.



Gambar 4.23 Hasil Visualisasi Pelatihan YOLO11l

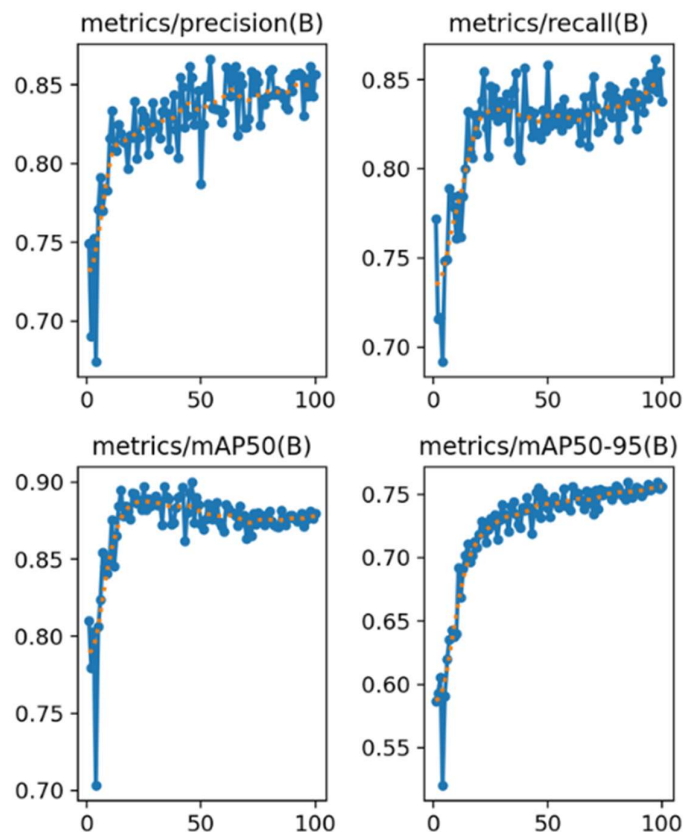
Berdasarkan hasil pelatihan dan validasi yang ditunjukkan pada Gambar 4.23 terjadi tren peningkatan performa yang konsisten dan stabil sepanjang 100 *epoch*. Nilai *loss* pada proses pelatihan yang terdiri dari *box loss*, *classification loss* dan *DFL loss* mengalami penurunan seiring berjalannya *epoch*.



Gambar 4.24 Tren penurunan *loss* YOLO11

Pada Gambar 4.24, *box loss* berada pada ± 1.4 dan menurun hingga ± 0.5 pada akhir pelatihan yang mengindikasikan bahwa model semakin akurat dalam memprediksi posisi *bounding box*. *Classification loss* turun dari ± 1.4 menjadi ± 0.3 yang menunjukkan peningkatan kemampuan model dalam membedakan kelas objek secara tepat. *DFL loss* juga mengalami penurunan dari ± 1.4 menjadi ± 0.9 yang mencerminkan peningkatan presisi pada prediksi koordinat *bounding box* melalui distribusi estimasi yang lebih baik.

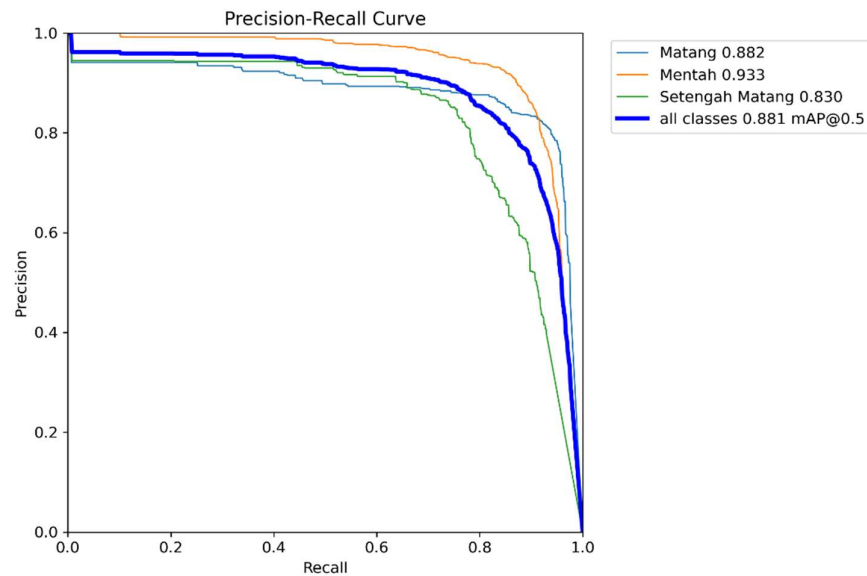
Pada tahap validasi, pola penurunan yang serupa terjadi seperti pelatihan. *Box loss* berkurang dari ± 1.0 menjadi ± 0.66 , *classification loss* menurun dari ± 1.4 menjadi ± 0.6 dan *DFL loss* berkurang dari ± 1.14 menjadi ± 0.98 . Konsistensi antara tren pada data latih dan data validasi ini menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami *overfitting*.



Gambar 4.25 Tren Peningkatan Kinerja YOLO111

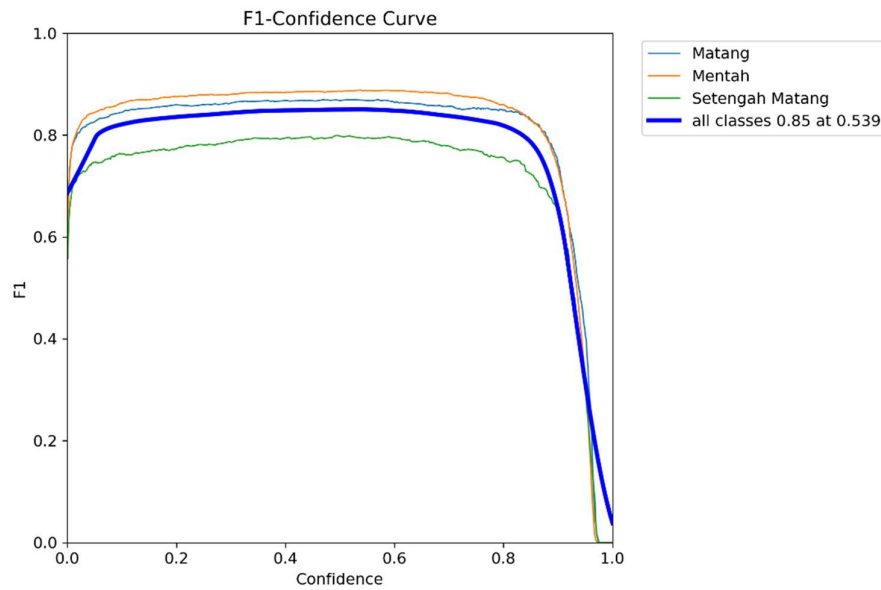
Pada Gambar 4.25, Evaluasi metrik performa menunjukkan adanya peningkatan. Nilai *precision* meningkat dari ± 0.70 menjadi ± 0.86 yang berarti proporsi prediksi positif yang benar semakin tinggi dengan penurunan *false positive*. Nilai *recall* naik dari ± 0.70 menjadi ± 0.85 yang mengindikasikan bahwa model mampu mendeteksi sebagian besar objek yang ada dengan *false negative* yang semakin sedikit. Nilai *mAP50* mengalami peningkatan dari ± 0.70 menjadi ± 0.88 menandakan bahwa model memiliki tingkat akurasi tinggi dalam deteksi objek pada toleransi posisi *bounding box* yang cukup longgar. Sementara itu, *mAP50-95* yang mengukur akurasi pada rentang ambang IoU yang lebih ketat naik dari ± 0.50 menjadi ± 0.76 menunjukkan kemampuan model mempertahankan performa pada tingkat ketelitian yang lebih tinggi.

Secara keseluruhan, penurunan *loss* yang konsisten pada pelatihan dan validasi, peningkatan nilai *precision* dan *recall* serta tingginya skor *mAP* pada berbagai tingkat ambang IoU menjadi bukti bahwa model YOLO1 yang dikembangkan memiliki kinerja deteksi objek yang andal. Model mampu mencapai keseimbangan yang baik antara sensitivitas deteksi dan ketepatan prediksi.



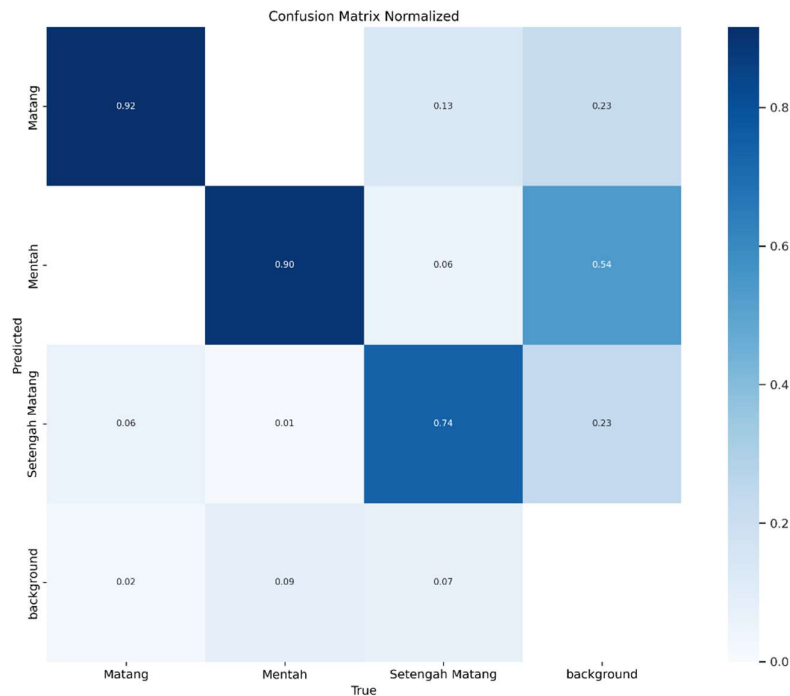
Gambar 4.26 Kurva Hubungan *Precision* dan *Recall* YOLO11

Pada Gambar 4.26, disajikan kurva *precision-recall* yang memvisualisasikan hubungan antara *precision* dan *recall*. Berdasarkan hasil evaluasi pada gambar tersebut, kelas mentah menunjukkan performa terbaik dengan nilai AP sebesar 0.933. Kelas ini diikuti oleh kelas matang dengan AP 0.882. Performa terendah tercatat pada kelas setengah matang dengan nilai AP 0.830. Perbedaan nilai AP ini mengindikasikan bahwa kemampuan model pada setiap kelas beragam. Secara komprehensif, evaluasi mendapatkan nilai *mAP50* dengan nilai 0.881 yang lebih rendah dari varian model sebelumnya.



Gambar 4.27 Kurva Hubungan *F1* dan *Confidence* YOLO11l

Pada Gambar 4.27 menunjukkan kurva *F1-Confidence* yang mengevaluasi performa model pada berbagai tingkat kepercayaan (*confidence threshold*). Kurva ini memetakan hubungan antara *F1-Score* dan nilai ambang batas kepercayaan dengan kurva gabungan untuk semua kelas yang ditandai dengan garis biru tebal. Berdasarkan data tersebut, performa model secara keseluruhan mencapai titik optimalnya dengan *F1-Score* puncak sebesar 0.85 pada saat *confidence threshold* berada di angka 0.539. Pada titik tersebut, model mampu mencapai keseimbangan yang optimal antara *precision* dan *recall* untuk gabungan semua kelas.



Gambar 4.28 *Confusion Matrix normalized YOLO11l*

Pada Gambar 4.28 di perlihatkan *normalized confusion matrix* yang menunjukkan perbandingan hasil prediksi model dengan label data yang sebenarnya. Berdasarkan matriks tersebut, model menunjukkan performa yang baik dalam mengenali kelas matang dan mentah dengan nilai masing-masing sebesar 92% dan 90%. Untuk kelas setengah matang, model menunjukkan nilai sebesar 74% dari total data aktual pada kelas tersebut.

4.2 Hasil Seluruh Varian Model

Pada tahap ini, dilakukan evaluasi terhadap empat varian model YOLO11 yaitu YOLO11n (*nano*), YOLO11s (*small*), YOLO11m (*medium*) dan YOLO11l (*large*) untuk deteksi tingkat kematangan tomat. Kinerja setiap model diukur menggunakan metrik *precision*, *recall* dan *mAP50* untuk tugas deteksi tingkat kematangan tomat. Hasil perbandingan metrik evaluasi dari keempat model tersebut disajikan pada Tabel 4.1.

Berdasarkan data pada tabel Tabel 4.1, model YOLO11m menunjukkan performa paling unggul secara keseluruhan dengan nilai *mAP50* sebesar 89.1%, *precision* 85.9% dan *recall* 84.1%. Meskipun varian YOLO11l mencatatkan nilai *precision* dan *recall* yang sedikit lebih tinggi, namun nilai *mAP50* yang didapatkan menurun menjadi 88.1% yang sama halnya dengan varian YOLO11n. Penurunan tersebut mengindikasikan bahwa peningkatan kompleksitas pada

model YOLO11l tidak memberikan keuntungan performa secara keseluruhan untuk *dataset* penelitian ini. Meskipun semua model menunjukkan hasil yang menjanjikan, ditemukan tantangan dalam akurasi deteksi pada kelas “Setengah Matang” dan adanya kesalahan identifikasi latar belakang. Isu tersebut menjadi catatan penting untuk potensi optimasi di masa mendatang.

Berdasarkan perbandingan keseimbangan antara *mAP50* dan performa yang konsisten di seluruh metrik, YOLO11m dipilih sebagai model untuk deteksi tingkat kematangan buah tomat pada penelitian ini.

Tabel 4.1 Perbandingan Hasil Performa Model

Varian Model	Kelas	<i>Precision</i>	<i>Recall</i>	<i>mAP50</i>
YOLO11n	Seluruh Kelas	0.859	0.810	0.881
	Matang	0.910	0.888	0.945
	Setengah Matang	0.741	0.638	0.776
	Mentah	0.924	0.904	0.921
YOLO11s	Seluruh Kelas	0.856	0.834	0.889
	Matang	0.900	0.891	0.937
	Setengah Matang	0.741	0.680	0.808
	Mentah	0.928	0.930	0.932
YOLO11m	Seluruh Kelas	0.859	0.841	0.891
	Matang	0.923	0.914	0.931
	Setengah Matang	0.753	0.730	0.821
	Mentah	0.900	0.880	0.921
YOLO11l	Seluruh Kelas	0.862	0.843	0.881
	Matang	0.906	0.924	0.933
	Setengah Matang	0.746	0.745	0.810
	Mentah	0.934	0.860	0.902

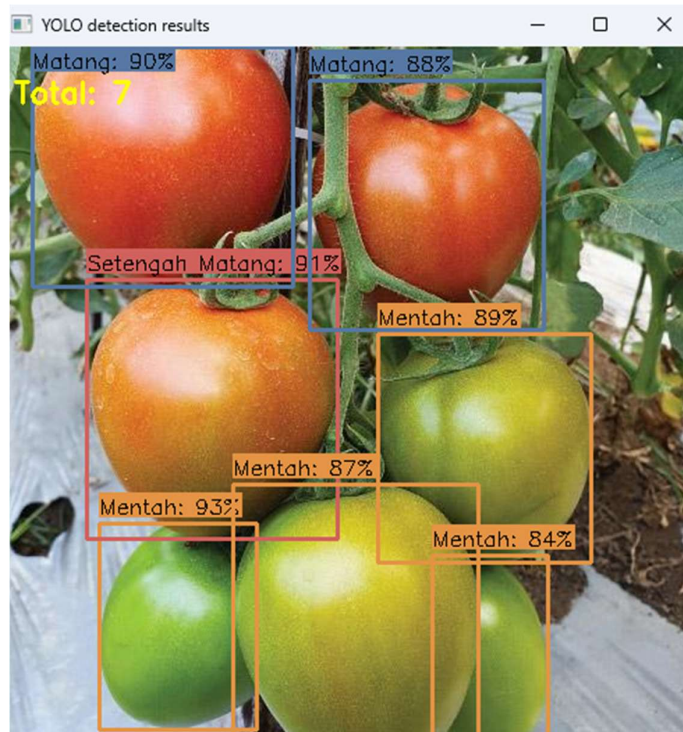
4.3 Hasil Pengujian

Pada tahap pengujian ini, dilakukan evaluasi terhadap kemampuan model deteksi objek yang telah dilatih dan dipilih pada tahap sebelumnya untuk mendeteksi tingkat kematangan buah tomat berdasarkan citra yang telah di *input*. Pengujian ini bertujuan untuk melihat performa model dalam skenario penggunaan yang sebenarnya. Dua sampel citra yaitu Gambar 4.29 dan Gambar 4.30 yang menampilkan citra tomat dengan tingkat kematangan yang berbeda. Sistem diharapkan mampu mendeteksi setiap buah tomat dalam citra dan memberikan salah satu dari tiga kategori yang telah ditentukan dengan skor kepercayaan untuk setiap deteksi.



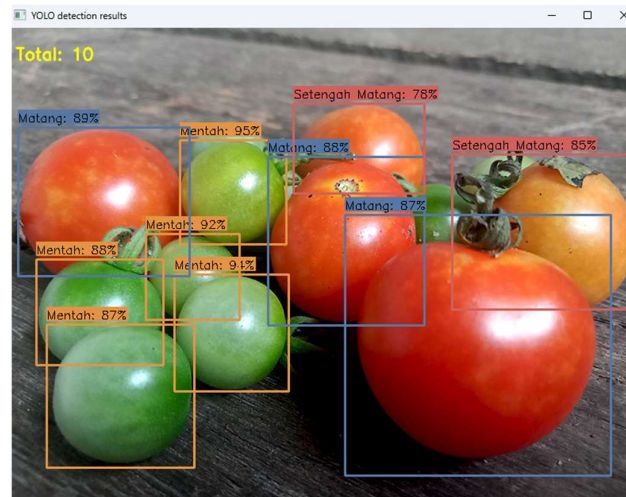
Gambar 4.29 Hasil Pengujian Gambar Satu

Pada Gambar 4.29, model berhasil mendeteksi dengan total tujuh buah tomat yang ditandai dengan sebuah kotak pembatas (*bounding box*). Fungsi dari kotak pembatas ini adalah untuk melokalisasi posisi dan cakupan area objek yang terdeteksi secara presisi pada citra. Kemudian, disematkan sebuah kategori tingkat kematangan yang disertai dengan nilai persentase yang merupakan skor kepercayaan (*confidence score*). Skor kepercayaan tersebut mempresentasikan tingkat keyakinan model terhadap kebenaran prediksinya. Dari hasil yang ditampilkan, model mampu memberikan skor kepercayaan tinggi yang berada di antara 87% hingga 95%. Hasil tersebut mengindikasikan bahwa model memiliki akurasi dan reliabilitas yang sangat baik.



Gambar 4.30 Hasil Pengujian Gambar Dua

Selanjutnya pada Gambar 4.30, model berhasil mendeteksi tujuh buah tomat dengan tiga kategori berbeda dengan dua buah terdeteksi sebagai matang, satu buah setengah matang dan empat buah sisanya sebagai mentah. Skor kepercayaan yang dicapai untuk semua kelas menunjukkan kinerja yang baik dengan rentang nilai yang beragam antara 84% hingga 93%. Hal tersebut menunjukkan bahwa model tidak hanya akurat, tetapi juga memiliki kemampuan yang baik untuk membedakan antara beberapa kematangan secara bersamaan.



Gambar 4.31 Hasil Pengujian Gambar Tiga

Pada hasil pengujian pada Gambar 4.31, model berhasil mendeteksi total sepuluh dari sebelas buah tomat yang terdistribusi ke dalam tiga kategori yaitu tiga buah terdeteksi sebagai matang, dua buah sebagai setengah matang dan lima buah sebagai mentah. Rentang skor kepercayaan yang dihasilkan berada antara 78% hingga 95%. Hasil tersebut menunjukkan performa yang baik terutama mengingat kondisi objek yang padat dan banyak mengalami tumpang tindih. Namun, tercatat satu kegagalan terdeteksi (*False Negative*) pada sebuah tomat yang mengalami tumpang tindih yang berat. Kegagalan ini disebabkan oleh minimnya fitur visual yang terlihat atau skor kepercayaannya tidak melampaui ambang batas yang ditetapkan sehingga menyoroti area potensial untuk perbaikan model di masa mendatang.



Gambar 4.32 Hasil Pengujian Menggunakan *Webcam*

Pada Gambar 4.32 menampilkan hasil pengujian yang berbeda dengan skenario sebelumnya. Pengujian ini dilakukan menggunakan *webcam* dan model berhasil mendeteksi seluruh buah tomat yang ada dengan satu buah terdeteksi sebagai matang dan dua buah sebagai setengah matang. Skor kepercayaan yang dihasilkan tinggi dengan 93% dan 95%. Kemudian, nilai *FPS* sebesar 16.09 menunjukkan bahwa model yang dikembangkan memiliki kemampuan untuk melakukan deteksi secara *real-time* dengan baik, meskipun faktor-faktor seperti kualitas *webcam* dan kondisi pencahayaan sekitar dapat mempengaruhi performa deteksi. Kinerja dengan kecepatan tersebut dapat menegaskan potensi model untuk diterapkan pada skenario operasional. Secara praktis, sistem ini dapat diimplementasikan dengan memasang kamera atau *webcam* di atas *conveyor belt* atau meja sortir di area pascapanen. Sistem kemudian menyediakan informasi klasifikasi kematangan secara visual dan *real-time* pada layar monitor. Informasi ini kemudian menjadi panduan objektif bagi siapa saja dalam rantai pasok mulai dari petani saat melakukan sortasi awal, distributor dalam proses kontrol kualitas hingga industri pengolahan pangan yang memerlukan keseragaman bahan baku. Dengan demikian, hasil pengujian ini tidak hanya memvalidasi performa teknis model, tetapi juga menunjukkan kelayakan fungsionalnya untuk menyelesaikan masalah penyortiran manual.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai pengembangan model deteksi tingkat kematangan buah tomat menggunakan model YOLO11, dapat disimpulkan bahwa model tersebut berhasil dirancang dan dikembangkan melalui serangkaian tahapan mulai dari pengumpulan data, persiapan *dataset* yang meliputi anotasi, pembagian *dataset*, *auto-orient*, penyesuaian resolusi hingga augmentasi data dan dilanjutkan dengan proses pelatihan dan evaluasi model. Dari evaluasi performa terhadap empat varian model YOLO11 yaitu YOLO11n (*nano*), YOLO11s (*small*), YOLO11m (*medium*) dan YOLO11l (*large*) dalam mendeteksi tiga kelas kematangan tomat yaitu mentah, setengah matang dan matang. Ditemukan bahwa model YOLO11m menunjukkan kinerja paling optimal dengan mencapai nilai *precision* sebesar 85.9%, *recall* 84.1% dan *mAP50* tertinggi pada 89.1% untuk keseluruhan kelas. Lebih lanjut, hasil uji coba model YOLO11m pada data citra baru dan input video secara *real-time* melalui *webcam* memperlihatkan kemampuan model untuk mendeteksi tingkat kematangan buah tomat dengan cukup baik, meskipun terdapat keterbatasan dalam menghadapi beberapa kondisi pada objek dan variasi pencahayaan serta bayangan tertentu dengan performa *real-time* pada *webcam* tercatat sekitar ± 16 *fps* selama pengujian. Hasil ini menyimpulkan bahwa teknologi yang dikembangkan dapat menjadi solusi praktis bagi petani, distributor maupun industri pengolahan pangan untuk meningkatkan objektivitas dan kecepatan dalam mengklasifikasikan kematangan tomat. Meskipun demikian, teridentifikasi adanya keterbatasan performa dalam kondisi objek yang tumpang tindih dan variasi pencahayaan yang menjadi area untuk perbaikan di masa mendatang.

5.2 Saran

Terdapat beberapa hal yang dapat menjadi saran untuk pengembangan selanjutnya. Pertama, disarankan agar penelitian berikutnya dapat memperbanyak data latih yang digunakan seperti dengan data latih yang lebih beragam. Sehingga dapat meningkatkan kinerja model dalam mengenali tingkat kematangan tomat di berbagai kondisi yang lebih bervariasi.

Kedua, melakukan pengembangan lebih lanjut pada model pengujian. Pengembangan dapat mencakup optimalisasi proses *preprocessing* seperti anotasi dan augmentasi data yang dapat membantu meningkatkan kualitas dan konsistensi model secara keseluruhan.

DAFTAR PUSTAKA

- Alhasan, R. A., & Hamza, E. K. (2023). A Novel CNN Model with Dimensionality Reduction for WSN Intrusion Detection. *Revue d'Intelligence Artificielle*, 37(5), 1121–1131. <https://doi.org/10.18280/ria.370504>
- Anwar, A., Giono, B., & Idrus, Muh. (2022). Respons Pertumbuhan dan Produksi Tanaman Tomat (*Lycopersicum esculentum* Mill.) terhadap Pupuk NPK dan Kompos. *J. Agrotan*, 8(2).
- Aras, S., Tanra, P., & Bazhar, M. (2024). Deteksi Tingkat Kematangan Buah Tomat Menggunakan YOLOv5. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(2), 623–628. <https://doi.org/10.57152/malcom.v4i2.1270>
- Badan Pusat Statistik. (2024a). *Statistik Hortikultura 2023*. 21–22. <https://www.bps.go.id/id/publication/2024/06/10/790c957ba8892f9771aeefb7/statistik-hortikultura-2023.html>
- Badan Pusat Statistik. (2024b, August 5). *Ekonomi Indonesia Triwulan II-2024 Tumbuh 3,79 Persen (Q-to-Q), Ekonomi Indonesia Triwulan II-2024 Tumbuh 5,05 Persen (Y-on-Y), dan Ekonomi Indonesia Semester I-2024 Tumbuh 5,08 Persen (C-to-C)*. <https://www.bps.go.id/id/pressrelease/2024/08/05/2381/ekonomi-indonesia-triwulan-ii-2024-tumbuh-3-79-persen--q-to-q---ekonomi-indonesia-triwulan-ii-2024-tumbuh-5-05-persen--y-on-y---dan-ekonomi-indonesia-semester-i-2024-tumbuh-5-08-persen--c-to-c-.html>
- Badgular, C. M., Poulose, A., & Gan, H. (2024). Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review. *Computers and Electronics in Agriculture*, 223. <https://doi.org/10.1016/j.compag.2024.109090>
- Che, C., Zheng, H., Huang, Z., Jiang, W., & Liu, B. (2024). Intelligent Robotic Control System Based on Computer Vision Technology. *ArXiv*, abs/2404.01116. <https://api.semanticscholar.org/CorpusID:268819306>
- Daniels, S., Suciati, N., & Fathichah, C. (2021). Indonesian Sign Language Recognition using YOLO Method. *IOP Conference Series: Materials Science and Engineering*, 1077(1), 012029. <https://doi.org/10.1088/1757-899X/1077/1/012029>

- Diwan, T., Anirudh, G., & Tembhurne, J. V. (2023). Object detection using YOLO: challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, 82(6), 9243–9275. <https://doi.org/10.1007/s11042-022-13644-y>
- FAO. (2024, October 30). *Countries by commodity*. FAOSTAT. https://www.fao.org/faostat/en/#rankings/countries_by_commodity
- Hanny Hikmayanti H, Dwi Sulistya Kusumaningrum, & Cucu Sri Cahyanti. (2021). *Identifikasi Kematangan Buah Tomat Berdasarkan Warna Menggunakan Metode Hue Saturation Value*. II(1).
- Hanum, M. (2024). Implementasi Teknik Embossing pada Pengenalan Plat Kendaraan untuk Identifikasi Otomatis Berbasis OpenCV. *JoMMiT : Jurnal Multi Media Dan IT*, 8(1). <https://doi.org/10.46961/jommit.v8i1.1361>
- Hetharua, A. D., Sumarno, S., Gunawan, I., Hartama, D., & Kirana, I. O. (2021). Alat Penyortir Buah Tomat Berdasarkan Warna Berbasis Mikrokontroller Arduino. *Jurnal Penelitian Inovatif*, 1(2), 119–130. <https://doi.org/10.54082/jupin.18>
- Hu, C., Liu, X., Pan, Z., & Li, P. (2019). Automatic detection of single ripe tomato on plant combining faster R-CNN and intuitionistic fuzzy set. *IEEE Access*, 7, 154683–154696. <https://doi.org/10.1109/ACCESS.2019.2949343>
- Jegham, N., Koh, C. Y., Abdelatti, M., & Hendawi, A. (2024). *Evaluating the Evolution of YOLO (You Only Look Once) Models: A Comprehensive Benchmark Study of YOLO11 and Its Predecessors*. <https://doi.org/10.48550/arXiv.2411.00201>
- Jocher, G., & Qiu, J. (2024). *Ultralytics YOLO11* (11.0.0). <https://github.com/ultralytics/ultralytics>
- Kaur, J., & Singh, W. (2022). Tools, techniques, datasets and application areas for object detection in an image: a review. *Multimedia Tools and Applications*, 81(27), 38297–38351. <https://doi.org/10.1007/s11042-022-13153-y>
- Kurniawan, R., Martadinata, A. T., & Cahyo, S. D. (2023). Klasifikasi Tingkat Kematangan Buah Sawit Berbasis Deep Learning dengan Menggunakan Arsitektur Yolov5. *Journal of Information System Research (JOSH)*, 5(1), 302–309. <https://doi.org/10.47065/josh.v5i1.4408>
- Liu, W., Meng, Q., Li, Z., & Hu, X. (2021). Applications of Computer Vision in Monitoring the Unsafe Behavior of Construction Workers: Current Status and Challenges. *Buildings*, 11(9). <https://doi.org/10.3390/buildings11090409>

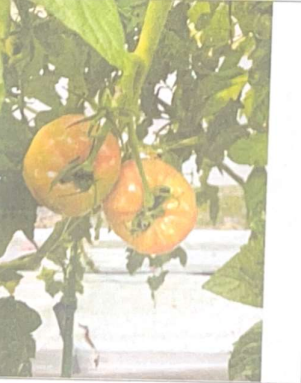
- Marpaung, F., Aulia, F., & Nabila, R. C. (2022). *Computer Vision Dan Pengolahan Citra Digital*. PUSTAKA AKSARA.
- Mputu, H. S., Abdel-Mawgood, A., Shimada, A., & Sayed, M. S. (2024). Tomato Quality Classification Based on Transfer Learning Feature Extraction and Machine Learning Algorithm Classifiers. *IEEE Access*, 12, 8283–8295. <https://doi.org/10.1109/ACCESS.2024.3352745>
- Muarofah, I. A., Jauhar Vikri, M., & Sa'ida, I. A. (2023). Real-Time Tomato Quality Detection System Using You Only Look Once (YOLOv7) Algorithm. *Ultimatics : Jurnal Teknik Informatika*, 15(2), 89.
- Muchtar, K., Chairuman, Yudha Nurdin, & Afdhal Afdhal. (2021). Pendeteksian Septoria pada Tanaman Tomat dengan Metode Deep Learning berbasis Raspberry Pi. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(1), 107–113. <https://doi.org/10.29207/resti.v5i1.2831>
- Muzammil, M. A. A., & Indraswari, R. (2024). Pengembangan Arsitektur Model YOLOv8 untuk Meningkatkan Performa Object Detection pada Varian Boks Warehouse Palletizing. *ILKOMNIKA*, 6(2), 135–146.
- Naf'an, E., Islami, F., & Gushelmi, G. (2022). *DASAR-DASAR DEEP LEARNING dan Contoh Aplikasinya*. Mitra Cendekia Media.
- Nima, R., & Shila, F. (2020). Crack classification in rotor-bearing system by means of wavelet transform and deep learning methods: an experimental investigation. *Journal of Mechanical Engineering, Automation and Control Systems*, 1(2), 102–113. <https://doi.org/10.21595/jmeacs.2020.21799>
- Prasetya, S. A., Mihuandayani, M., Abast, Y., Mangole, M., & Rahman, J. (2024). Implementasi Convolutional Neural Network untuk Klasifikasi Tingkat Kematangan Buah Nanas Menggunakan YOLOv8. *Building of Informatics, Technology and Science (BITS)*, 6(1), 567–575. <https://doi.org/10.47065/bits.v6i1.5396>
- Putra, F. S., Kusri, K., & Kurniawan, M. P. (2021). Deteksi Otomatis Jerawat Wajah Menggunakan Metode Convolutional Neural Network (CNN). *Journal of Information Technology*, 1(2), 30–34. <https://doi.org/10.46229/jifotech.v1i2.308>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. <https://arxiv.org/abs/1506.02640>

- Sacer, A. Y., Aulawi, T., Suryani, P., & Mirdhayati, I. (2022). *EFEKTIVITAS LAMA TRANSPORTASI DARAT YANG BERBEDA TERHADAP MUTU FISIK TOMAT (Lycopersicum esculentum L.)*.
- Salehi, B., Sharifi-Rad, R., Sharopov, F., Namiesnik, J., Roointan, A., Kamle, M., Kumar, P., Martins, N., & Sharifi-Rad, J. (2019). Beneficial effects and potential risks of tomato consumption for human health: An overview. In *Nutrition* (Vol. 62, pp. 201–208). Elsevier Inc. <https://doi.org/10.1016/j.nut.2019.01.012>
- Sihananto, A. N., Safitri, E. M., Maulana, Y., Fakhruddin, F., & Yudistira, M. E. (2023). Indonesian Sign Language Image Detection Using Convolutional Neural Network (CNN) Method. *Inspiration: Jurnal Teknologi Informasi Dan Komunikasi*, 13(1), 13–21. <https://doi.org/10.35585/inspir.v13i1.37>
- Tsoi, P. (2024). *YOLO11: The Cutting-Edge Evolution in Object Detection — A Brief Review of the Latest in the YOLO Series*. <https://medium.com/@paultsoi1014/yolo11-the-cutting-edge-evolution-in-object-detection-a-brief-review-of-the-latest-in-the-yolo-19065729edfe>
- Wang, S., Xiang, J., Chen, D., & Zhang, C. (2024). A Method for Detecting Tomato Maturity Based on Deep Learning. *Applied Sciences*, 14(23). <https://doi.org/10.3390/app142311111>
- White, A. D. (2022). Deep Learning for Molecules and Materials. *Living Journal of Computational Molecular Science*, 3(1), 1499. <https://doi.org/10.33011/livecoms.3.1.1499>
- 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. <https://doi.org/10.35138/paspalum.v11i1.489>
- Yanti, L., Hanafi, H., & Yassir, Y. (2024). Rancang Bangun Sistem Penyortiran Kematangan Buah Tomat Berbasis Internet of Things. *Jurnal Litek : Jurnal Listrik Telekomunikasi Elektronika*, 21(2), 89–94. <https://doi.org/10.30811/litek.v21i2.40>
- Zhang, C., Kang, F., & Wang, Y. (2022). An Improved Apple Object Detection Method Based on Lightweight YOLOv4 in Complex Backgrounds. *Remote Sensing*, 14(17). <https://doi.org/10.3390/rs14174150>
- Zhao, J., Du, C., Li, Y., Mudhsh, M., Guo, D., Fan, Y., Wu, X., Wang, X., & Almodfer, R. (2024). YOLO-Granada: a lightweight attentioned Yolo for pomegranates fruit detection. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-67526-4>

LAMPIRAN

FOTO	Keterangan	Benar/Salah
	6 Tomat Matang	✓
	2 Tomat Matang	✓
	2 Tomat Matang	✓

	<p>4 Tomat Mentah</p>	<p>✓</p>
	<p>2 Tomat Mentah</p>	<p>✓</p>
	<p>4 Tomat Mentah</p>	<p>✓</p>

	3 Tomat Setengah Matang	✓
	3 Tomat Setengah Matang	✓
	2 Tomat Setengah Matang	✓