

**VISUALISASI *DIGITAL MODEL* FASE PERTUMBUHAN TANAMAN MELON
BERBASIS DETEKSI *YOLOv8* MENGGUNAKAN *UNITY***

SKRIPSI

Oleh:

MUHAMMAD YASRIL ADIM AL AMIN
NIM. 220605110070



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

**VISUALISASI *DIGITAL MODEL* FASE PERTUMBUHAN TANAMAN
MELON BERBASIS DETEKSI *YOLOv8* MENGGUNAKAN *UNITY***

SKRIPSI

Diajukan Kepada:

Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:

MUHAMMAD YASRIL ADIM AL AMIN
220605110070

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

HALAMAN PERSETUJUAN

VISUALISASI *DIGITAL MODEL* FASE PERTUMBUHAN TANAMAN MELON BERBASIS DETEKSI *YOLOv8* MENGGUNAKAN *UNITY*

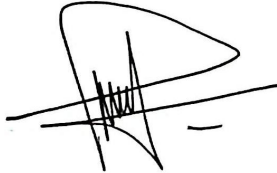
SKRIPSI

Oleh:

MUHAMMAD YASRIL ADIM AL AMIN
NIM. 220605110070

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 10 Juni 2026

Pembimbing I,



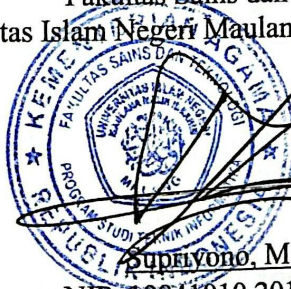
Dr. Ir. Yunifa Miftachul Arif, S.ST., M.T., SMIEEE
NIP. 19830616 201101 1 004

Pembimbing II,



Nur Fitriyah Ayu Tunjung Sari, M.Cs
NIP. 19911226 202012 2 001

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Supriyono, M.Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

VISUALISASI *DIGITAL MODEL* FASE PERTUMBUHAN TANAMAN MELON BERBASIS DETEKSI *YOLOv8* MENGGUNAKAN *UNITY*

SKRIPSI

Oleh:

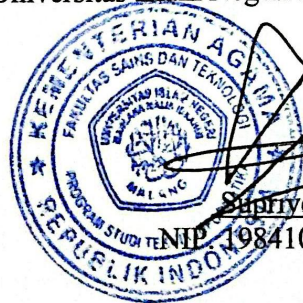
MUHAMMAD YASRIL ADIM AL AMIN
NIM. 220605110070

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 22 Juni 2026

Susunan Dewan Penguji

Ketua Penguji	: <u>Prof. Dr. Ir. Muhammad Faisal, M.T</u> NIP. 19740510 200501 1 007	()
Anggota Penguji I	: <u>Shoffin Nahwa Utama, M.T</u> NIP. 19860703 202012 1 003	()
Anggota Penguji II	: <u>Dr. Ir. Yunifa Miftachul Arif, S.ST., M.T., SMIEEE</u> NIP. 19830616 201101 1 004	()
Anggota Penguji III	: <u>Nur Fitriyah Ayu Tunjung Sari, M.Cs</u> NIP. 19911226 202012 2 001	()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Supriyono, M.Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Yasril Adim Al Amin
NIM : 220605110070
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Visualisasi *Digital Model* Fase Pertumbuhan
Tanaman Melon Berbasis Deteksi *YOLOv8*
Menggunakan *Unity*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 22 Juni 2026
Yang membuat pernyataan,



Muhammad Yasril Adim Al Amin
NIM. 220605110070

MOTTO

“If the world ain’t being to you, you gotta be nice to yourself”

-My Friend-

HALAMAN PERSEMBAHAN

Puji syukur ke hadirat Allah *Subhanahu Wa Ta'ala* atas segala rahmat, kemudahan, dan hidayah-Nya, Skripsi ini dapat terselesaikan dengan baik. Untuk itu, penulis mempersembahkan skripsi ini sebagai:

1. Tanda bakti dan rasa terima kasih yang sedalam-dalamnya kepada kedua orang tua penulis ayahanda Ir. Moch Rudy Al Amin, M.Agr, dan Ibunda Juwariyah, sebagai motivator terbesar dalam hidup penulis, yang senantiasa mengiringi setiap langkah penulis dengan doa yang tulus, kesabaran yang tak terbatas, dan kasih sayang yang tak pernah pudar. Serta kepada kedua kakak penulis, Adam Uqbah Al Amin, S.Ars, dan Qurrota A'yun Al Amin, S.T, yang selalu menjadi penyemangat dalam setiap suka maupun duka.
2. Rasa hormat dan rasa terima kasih kepada dosen pembimbing penulis yang luar biasa, Bapak Dr. Ir. Yunifa Miftachul Arif, S.ST., M.T., SMIEEE dan Ibu Nur Fitriyah Ayu Tunjung Sari, M.Cs. Selaku dosen pembimbing yang dengan penuh dedikasi dan keikhlasan telah membimbing, mengoreksi, serta memberikan arahan yang sangat berarti bagi penulis. Semoga segala ilmu dan kebaikan yang telah diberikan menjadi amal jariyah yang senantiasa mengalir.
3. Tanda rasa syukur memiliki teman seperjuangan, terkhusus rekan-rekan yang selalu bersama setiap langkah semester semasa perkuliahan penulis, yang selalu hadir dengan semangat canda dan tawa di saat paling dibutuhkan. Terimakasih telah menjadi alasan untuk terus melangkah,

semoga setiap doa baik yang kalian titipkan kembali kepada kalian berlipat ganda.

4. Tanda terima kasih kepada seluruh pihak yang tidak dapat disebutkan satu-persatu, yang telah memberikan doa, dukungan, bantuan, dan semangat dalam bentuk apapun selama proses penyelesaian skripsi ini. Semoga Allah *Subhanahu Wa Ta'ala* membalas segala kebaikan kalian dengan balasan yang jauh lebih indah

KATA PENGANTAR

Assalamu 'alaikum warahmatullahi wabarakatuh

Alhamdulillahirabbilalamin, Segala puji dan rasa syukur senantiasa penulis panjatkan atas ke hadirat Allah *Subhanahu wa Ta'ala* atas segala nikmat dan kasih sayang-Nya yang telah memudahkan penulis untuk menyelesaikan skripsi berjudul ”Visualisasi *Digital Model* Fase Pertumbuhan Tanaman Melon Berbasis Deteksi *YOLOv8* Menggunakan *Unity*” dengan baik. Shalawat serta salam penulis haturkan kepada junjungan kita Nabi besar Muhammad SAW, yang telah membawa kita dari jalan jahiliyah menuju jalan yang terang yaitu *Ahdinul islam wal iman*. Dan semoga kita semua mendapat syafaat-nya di hari akhir kelak, Amin.

Penulis mengucapkan rasa syukur dan terima kasih yang sebesar-besarnya kepada seluruh pihak yang selalu memberikan dukungan dan motivasi kepada penulis sehingga dapat menyelesaikan skripsi ini. Ucapan terima kasih ini penulis sampaikan kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. H. Agus Mulyono, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Supriyono, M.Kom, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Ir. Yunifa Miftachul Arif, S.ST., M.T., SMIEEE, selaku dosen pembimbing I dan Nur Fitriyah Ayu Tunjung Sari, M.Cs, selaku dosen

pembimbing II, yang selalu memberikan bimbingan, saran, dan dukungan yang sangat berarti bagi penulis selama proses penelitian.

5. Prof. Dr. Ir. Muhammad Faisal, M.T, selaku dosen Penguji I dan Shoffin Nahwa Utama, M.T, selaku dosen Penguji II yang telah memberikan kritik dan saran dalam proses penyelesaian penelitian ini.
6. Seluruh dosen dan staf Fakultas Sains dan Teknologi, terkhusus Program Studi Teknik Informatika yang telah memberikan ilmu, dukungan, dan bantuan administratif selama masa studi penulis.
7. Ayahanda Ir. Moch Rudy Al Amin, M.Agr, dan Ibunda Juwariyah yang memiliki peran besar dan motivator dalam perjalanan penulis selama menempuh hidup dan pendidikan perkuliahan. Serta kedua kakak penulis Adam Uqbah Al Amin, S.Ars, dan Qurrota Ayun Al Amin, S.T, yang selalu memberikan arahan positif kepada penulis.
8. Dimas Eka Putra, Muhammad Hafidul Qolbi, Muhammad Nabil Hilmi Miftachurrizqi, Ahmad Ghiffari Fadhil Saputra, M Singgi Aditya Ramadhan, dan Putri Rahayu Agustin yang senantiasa menjadi rekan berdiskusi dan bertukar pemikiran baik dalam bidang akademik maupun kehidupan sehari-hari selama masa perkuliahan hingga proses penyelesaian skripsi ini.
9. Teman-teman yang selalu hadir dan menemani penulis selama masa perkuliahan, yaitu Salman, Fairuz, Tiara, Agam, Epen, Adit, Dimas, Lefa, Egiq, serta teman-teman lain yang tidak dapat disebutkan satu persatu.

10. Teman-teman "Project Smart Greenhouse Melon", yang senantiasa membantu bertukar wawasan terkhusus dalam bidang penelitian penulis.
11. Keluarga besar "Himatif Encoder", terkhusus divisi religius periode 2023-2025, yang telah memberikan pengalaman berharga dalam organisasi selama masa perkuliahan penulis.
12. Serta seluruh pihak yang terlibat baik secara langsung maupun secara tidak langsung dari awal perkuliahan hingga penulisan skripsi ini.

Akhir Kata, penulis menyadari bahwa penulisan pada skripsi ini masih banyak kekurangan. Penulis berharap semoga penulisan skripsi ini dapat menjadi bagian dari kontribusi dalam rangka memperkuat dan mengembangkan ilmu pengetahuan lebih lanjut.

Wassalamualaikum Warahmatullahi Wabarakatuh

Malang, 22 Juni 2026

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xv
ABSTRAK	xvi
ABSTRACT	xvii
مستخلص البحث	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	6
1.3 Batasan Masalah.....	6
1.4 Tujuan Penelitian	7
1.5 Manfaat Penelitian	7
BAB II STUDI PUSTAKA.....	8
2.1 Penelitian Terkait	8
2.2 Melon	13
2.2.1 Fase Pertumbuhan Tanaman Melon.....	13
2.3 <i>Digital Model</i>	14
2.4 <i>You Only Look Once (YOLO)</i>	15
BAB III DESAIN DAN IMPLEMENTASI	17
3.1 Desain Penelitian.....	17
3.2 Pengumpulan dan Persiapan <i>Dataset</i>	18
3.2.1 Proses Anotasi (<i>Labelling</i>) <i>Dataset</i>	20
3.2.2 <i>Pre-Processing Dataset</i>	24
3.2.3 Pembagian <i>Dataset (Splitting Dataset)</i>	24
3.2.4 <i>Augmentasi Dataset</i>	25
3.3 Desain Sistem.....	26
3.3.1 Arsitektur Sistem.....	27
3.3.2 Alur Sistem	28
3.3.3 Mekanisme Penentuan Fase	29
3.3.4 Perancangan <i>Prefabs Digital Model</i>	31
3.4 Pengembangan Model.....	31
3.4.1 Arsitektur YOLOv8	32
3.4.2 Konfigurasi dan Pelatihan Model.....	35
3.4.3 Evaluasi Hasil Pelatihan.....	35
3.4.4 Ekspor ONNX Model	37
3.5 Implementasi Sistem	38
3.5.1 Integrasi ONNX <i>Model</i> ke <i>Unity</i>	38
3.5.2 Pengembangan <i>Prefab 3D Digital Model</i>	41
3.5.3 Konfigurasi <i>Scene Unity</i>	42
3.6 Skenario Pengujian Sistem.....	43
3.6.1 Skenario Pengujian <i>Tuning Parameter YOLOv8</i>	44

3.6.2 Skenario Pengujian Deteksi <i>Model YOLOv8</i>	45
3.6.3 Skenario Pengujian Komponen Visualisasi <i>Digital Model</i> (Penentuan Fase Untuk Visualisasi Fase Pertumbuhan Tanaman Melon).....	47
3.7 Evaluasi <i>User</i>	49
3.7.1 Instrumen <i>uMARS</i>	50
3.7.2 Perhitungan Skor <i>uMARS</i>	51
3.7.3 Cakupan dan Mekanisme Uji	52
BAB IV DESAIN DAN IMPLEMENTASI	54
4.1 Tata Cara Pengujian	54
4.2 Hasil Pengujian <i>Tuning Parameter YOLOv8</i>	55
4.2.1 Hasil Pengujian <i>Tuning Parameter</i> Tahap 1 (Variasi <i>Epochs</i> dan <i>Batch Size</i>)....	55
4.2.2 Hasil Pengujian <i>Tuning Parameter</i> Tahap 2 (Variasi <i>Optimizer</i>).....	56
4.2.3 Konfigurasi Parameter Terbaik	57
4.3 Hasil Pengujian Deteksi Model <i>YOLOv8</i>	61
4.4 Hasil Pengujian Komponen Visualisasi <i>Digital Model</i> (Penentuan Fase Untuk Visualisasi Fase Pertumbuhan Tanaman Melon).....	68
4.5 Evaluasi <i>User</i>	73
4.5.1 Hasil Kuisioner <i>uMARS</i>	73
4.5.2 Interpretasi Skor <i>uMARS</i>	74
4.6 Pembahasan.....	75
4.7 Integrasi Islam.....	79
4.7.1 <i>Muamalah Ma Allah</i>	79
4.7.2 <i>Muamalah Ma'annas</i>	81
BAB V KESIMPULAN DAN SARAN.....	83
5.1 Kesimpulan	83
5.2 Saran.....	84
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2. 1 Diagram Alir di <i>Digital Model</i> oleh (Kritzing et al., 2018)	15
Gambar 2. 2 Arsitektur YOLO oleh (Redmon et al., n.d.).....	15
Gambar 2. 3 Proses Deteksi pada Algoritma <i>YOLO</i> oleh (Redmon et al., n.d.)	16
Gambar 3. 1 Desain Penelitian.....	17
Gambar 3. 2 Contoh Pengambilan Gambar CCTV di waktu pagi (pukul 08:05 WIB) dan malam (22:05 WIB)	19
Gambar 3. 3 Proses Anotasi (<i>Labelling</i>) Pada Kelas <i>Early</i>	21
Gambar 3. 4 Proses Anotasi (<i>Labelling</i>) Pada Kelas <i>Late</i>	21
Gambar 3. 5 Proses Anotasi (<i>Labelling</i>) Pada Kelas Bunga.....	22
Gambar 3. 6 Proses Anotasi (<i>Labelling</i>) Pada Kelas Tumbuh.....	22
Gambar 3. 7 Proses Anotasi (<i>Labelling</i>) Pada Kelas Panen	23
Gambar 3. 8 Teknik <i>Rotate</i>	26
Gambar 3. 9 Desain Arsitektur Sistem.....	27
Gambar 3. 10 Diagram Alur Sistem.....	28
Gambar 3. 11 Arsitektur <i>YOLOv8</i> oleh (Terven & Cordova-esparza, 2024).....	33
Gambar 3. 12 Perintah Ekspor ke format <i>.onnx</i>	38
Gambar 3. 13 <i>Pseudo Code</i>	39
Gambar 3. 14 <i>Prefabs</i> Fase Melon.....	42
Gambar 3. 15 Desain <i>Scene Environment Greenhouse</i>	42
Gambar 3. 16 <i>Scene</i> Tanaman Melon	43
Gambar 4. 1 Kurva Metrik Evaluasi	58
Gambar 4. 2 Cabang Regresi dan Cabang Klasifikasi pada Bagian <i>Head</i> di Arsitektur <i>YOLOv8</i>	59
Gambar 4. 3 <i>Confusion Matrix</i>	60
Gambar 4. 4 Contoh Visual Fase <i>Early</i> Vegetatif Citra Uji Nomor 1	64
Gambar 4. 5 Contoh Visual Fase <i>Late</i> Vegetatif Citra Uji Nomor 3	65
Gambar 4. 6 Contoh Visual Fase Pembungaan Citra Uji Nomor 5	65
Gambar 4. 7 Contoh Visual Fase Petumbuhan Buah Citra Uji Nomor 7.....	66
Gambar 4. 8 Contoh Visual Fase Panen Citra Uji Nomor 9	67
Gambar 4. 9 Skenario <i>Minimum Count</i> (Skenario 1,2,3).....	70
Gambar 4. 10 Skenario <i>Fallback Majority Vote</i> (Skenario 4)	70
Gambar 4. 11 Skenario Seri dan <i>Unknown</i> (Skenario 5 dan 6)	71

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait.....	11
Tabel 3. 1 Jadwal Periode Tanam dan Pengambilan Dataset.....	18
Tabel 3. 2 Kriteria Fase Pertumbuhan Tanaman Melon	20
Tabel 3. 3 Distribusi Jumlah Gambar dan Anotasi Setiap Kelas	23
Tabel 3. 4 Distribusi Jumlah Gambar ter-Anotasi per Kelas Setelah Proses <i>Split Dataset</i>	25
Tabel 3. 5 Distribusi Jumlah Gambar ter-anotasi per Kelas sebelum proses Augmentasi dan <i>Split Dataset</i>	26
Tabel 3. 6 Kelas Fase Melon dan Indeksasi Kelas.....	30
Tabel 3. 7 Mekanisme <i>Minimum Count</i> dan Prioritas Penentuan Fase.....	30
Tabel 3. 8 Mekanisme <i>Fallback</i> dan <i>Tiebreaker</i>	31
Tabel 3. 9 Korespondensi Fase dan Prefabs	31
Tabel 3. 10 Nilai Parameter	35
Tabel 3. 11 Skenario Pengujian Tahap 1 - Variasi <i>Epochs</i> dan <i>Batch Size</i>	45
Tabel 3. 12 Skenario Pengujian Tahap 2 - Variasi <i>Optimizer</i>	45
Tabel 3. 13 Skenario Pengujian Deteksi Model YOLOv8.....	46
Tabel 3. 14 Skenario Pengujian Komponen Visualisasi Digital Model (Penentuan Fase untuk Visualisasi Fase Pertumbuhan Tanaman Melon).....	48
Tabel 3. 15 Daftar Pertanyaan Kuisisioner <i>uMARS</i>	50
Tabel 3. 16 Kategori Interpretasi Skor <i>uMARS</i>	52
Tabel 4. 1 Hasil Skenario Pengujian Tahap 1 - Variasi <i>Epochs</i> dan <i>Batch Size</i>	55
Tabel 4. 2 Hasil Skenario Pengujian Tahap 2 - Variasi <i>Optimizer</i>	56
Tabel 4. 3 Konfigurasi Parameter Terbaik Hasil Pengujain	57
Tabel 4. 4 Hasil Pengujian Deteksi Model YOLOv8.....	62
Tabel 4. 5 Rekapitulasi Hasil Pengujian Deteksi Model YOLOv8.....	67
Tabel 4. 6 Hasil Pengujian <i>Digital Model</i> (Penentuan Fase untuk Visualisasi Pertumbuhan Tanaman Melon).....	69
Tabel 4. 7 Rekapitulasi Hasil Pengujian <i>Digital Model</i>	72
Tabel 4. 8 Rekapitulasi Skor Rata-Rata per Subskala tiap Responden.....	73
Tabel 4. 9 Interpretasi Skor <i>uMARS</i>	75

ABSTRAK

Al Amin, Muhammad Yasril Adim. 2026. **Visualisasi Digital Model Fase Pertumbuhan Tanaman Melon Berbasis Deteksi YOLOv8 Menggunakan Unity**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Yunifa Miftachul Arif, M.T. (II) Nur Fitriyah Ayu Tunjung Sari, M.Cs.

Kata Kunci: Deteksi Objek, Tanaman Melon, Unity, Visualisasi Digital Model, YOLOv8.

Identifikasi fase pertumbuhan tanaman melon secara manual masih banyak dilakukan oleh petani dan peneliti, namun metode tersebut memakan waktu, tidak konsisten, dan belum mampu menghasilkan representasi visual yang intuitif terhadap kondisi pertumbuhan tanaman, terutama pada sistem budidaya *greenhouse* berskala besar. Penelitian ini bertujuan untuk membangun visualisasi *digital model* fase pertumbuhan tanaman melon berdasarkan hasil deteksi YOLOv8 yang direpresentasikan dalam lingkungan *Unity* menggunakan citra kamera CCTV yang diunggah pengguna ke dalam sistem. *Dataset* yang digunakan merupakan data *primer* berupa 199 snapshot CCTV dari *greenhouse* kelompok yang mencakup lima kelas fase pertumbuhan, yaitu *early* vegetatif, *late* vegetatif, pembungaan, pertumbuhan buah, dan panen, dengan seluruh proses *preprocessing*, anotasi, augmentasi, dan pembagian *dataset* dilakukan melalui *platform Roboflow*. Model YOLOv8 dilatih menggunakan pengujian *tuning parameter* dua tahap dan menghasilkan konfigurasi terbaik pada *epochs* = 150, *batch size* = 16, dan *optimizer* = Adam, dengan nilai *mAP@0.5* sebesar 0,8588 dan *mAP@0.5:0.95* sebesar 0,5510, serta mampu mendeteksi fase pertumbuhan tanaman melon dengan akurasi 90% dari 10 citra uji. Komponen visualisasi *digital model* menggunakan mekanisme penentuan fase yang terdiri dari mekanisme *minimum count*, *fallback majority vote*, dan *tiebreaker* berbasis urutan kematangan biologis untuk menerjemahkan *output* deteksi menjadi keputusan fase dan representasi *prefab 3D* di *Unity*, dengan tingkat keberhasilan 100% pada seluruh 6 skenario uji. Evaluasi pengguna menggunakan metode *uMARS* menghasilkan rata-rata skor 4,16 yang menempatkan aplikasi pada kategori Baik, dengan subskala *information quality* memperoleh nilai tertinggi sebesar 4,30. Hasil pengujian secara keseluruhan menunjukkan bahwa sistem visualisasi *digital model* yang dikembangkan mampu merepresentasikan perubahan fase pertumbuhan tanaman melon secara dinamis, interaktif, dan informatif berdasarkan citra kamera CCTV yang diunggah pengguna ke dalam sistem.

ABSTRACT

Al Amin, Muhammad Yasril Adim. 2026. **Digital Visualization of a Melon Growth Phase Model Based on YOLOv8 Detection Using Unity**. Thesis. Department of Informatics Engineering, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Counselor: (I) Dr. Yunifa Miftachul Arif, M.T. (II) Nur Fitriyah Ayu Tunjung Sari, M.Cs.

Keywords: Digital Model Visualization, Melon Plants, Object Detection, Unity, YOLOv8.

Manual identification of melon growth stages is still widely practiced by farmers and researchers; however, this method is time-consuming, inconsistent, and unable to produce an intuitive visual representation of plant growth conditions, particularly in large-scale greenhouse cultivation systems. This study aims to develop a digital visualization of melon plant growth phase models based on YOLOv8 detection results, represented within the Unity environment using CCTV camera images uploaded by users into the system. The dataset used consists of primary data comprising 199 CCTV snapshots from a greenhouse group covering five growth phase classes: early vegetative, late vegetative, flowering, fruit growth, and harvest. The entire process of preprocessing, annotation, data augmentation, and dataset splitting was conducted via the Roboflow platform. The YOLOv8 model was trained using a two-stage parameter tuning process and yielded the best configuration at epochs = 150, batch size = 16, and optimizer = Adam, with a mAP@0.5 of 0.8588 and an mAP@ 0.5:0.95 of 0.5510, and was able to detect melon plant growth phases with 90% accuracy across 10 test images. The model's digital visualization component employs a phase determination mechanism comprising a minimum count mechanism, a fallback majority vote, and a tiebreaker based on biological maturity sequence to translate detection outputs into phase decisions and prefab 3D representations, with a 100% success rate across all 6 test scenarios. User evaluation using the uMARS method yielded an average score of 4.16, placing the application in the "Good" category, with the information quality subscale receiving the highest score of 4.30. The overall test results show that the developed digital model visualization system is capable of representing changes in the growth phases of melon plants in a dynamic, interactive, and informative manner based on CCTV camera images uploaded by users into the system.

مستخلص البحث

الأمين، محمد ياسريل أديم. 2026. التصور الرقمي لنمو نبات البطيخ استناداً إلى تقنية الكشف YOLOv8 باستخدام Unity. أطروحة تخرج. برنامج هندسة المعلوماتية، كلية العلوم والتكنولوجيا، الجامعة الإسلامية الحكومية «مولانا مالك إبراهيم» في مالانج. المشرفان (I) د: يونيفا مفتاح العارف، ماجستير في الهندسة (II) نور فيتريه أيو تونج ساري، ماجستير في علوم الحاسوب.

الكلمات المفتاحية: التصور الرقمي للنموذج، نباتات البطيخ، YOLOv8، كشف الأجسام، Unity.

لا يزال المزارعون والباحثون يعتمدون بشكل كبير على تحديد مراحل نمو نبات البطيخ يدوياً، إلا أن هذه الطريقة تستغرق وقتاً طويلاً، وتفتقر إلى الاتساق، ولا توفر تمثيلاً بصرياً بديهياً لحالة نمو النبات، لا سيما في أنظمة زراعة الدفيئة واسعة النطاق. يهدف هذا البحث إلى إنشاء نموذج رقمي لتصوير مراحل نمو نبات البطيخ استناداً إلى نتائج الكشف بواسطة YOLOv8، والتي يتم تمثيلها في بيئة Unity باستخدام صور كاميرات المراقبة (CCTV) التي يقوم المستخدمون بتحميلها إلى النظام. تتكون مجموعة البيانات المستخدمة من بيانات أولية تتمثل في 199 لقطة من كاميرات المراقبة (CCTV) لدفيئات مجموعة تشمل خمس فئات لمراحل النمو، وهي: المرحلة الخضرية المبكرة، والمرحلة الخضرية المتأخرة، ومرحلة الإزهار، ومرحلة نمو الثمرة، ومرحلة الحصاد، حيث تم إجراء جميع عمليات المعالجة المسبقة، والتعليق، والتعزيز، وتقسيم مجموعة البيانات عبر منصة Roboflow. تم تدريب نموذج YOLOv8 باستخدام اختبار ضبط المعلومات على مرحلتين، وأسفر عن أفضل تكوين عند $epochs = 150$ ، $optimizer = Adam$ ، $batch\ size = 16$ ، مع قيمة $mAP@0.5$ تبلغ 0,8588، $mAP@0.5:0.95$ تبلغ 0,5510، كما تمكنت من اكتشاف مراحل نمو نبات البطيخ بدقة تبلغ 90% من أصل 10 صور اختبار. تستخدم مكونات التصور الرقمي للنموذج آلية لتحديد المرحلة تتألف من آلية «الحد الأدنى للعدد»، و«التصويت بالأغلبية الاحتياطية»، و«كسر التعادل» استناداً إلى تسلسل النضج البيولوجي، وذلك لترجمة نتائج الكشف إلى قرارات بشأن المرحلة وتمثيل ثلاثي الأبعاد جاهز في Unity، مع تحقيق نسبة نجاح بلغت 100% في جميع سيناريوهات الاختبار الستة. أسفرت تقييمات المستخدمين باستخدام طريقة uMARS عن متوسط درجة 4,16، مما يضع التطبيق في فئة «جيد»، حيث حصلت المقياس الفرعي لجودة المعلومات على أعلى درجة بلغت 4,30. أظهرت نتائج الاختبار بشكل عام أن نظام التصور الرقمي للنموذج الذي تم تطويره قادر على تمثيل تغيرات مراحل نمو نبات البطيخ بطريقة ديناميكية وتفاعلية وغنية بالمعلومات استناداً إلى صور كاميرات المراقبة التي يقوم المستخدمون بتحميلها إلى النظام.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Identifikasi fase pertumbuhan tanaman merupakan salah satu aspek penting dalam proses budidaya yang perlu dilakukan secara akurat dan terstruktur. Dalam praktiknya, petani maupun peneliti masih banyak mengandalkan observasi manual untuk menentukan fase pertumbuhan tanaman, mulai dari fase vegetatif hingga panen (Boahen, 2024). Metode manual tersebut memakan waktu, tidak selalu menghasilkan pengamatan yang konsisten, dan belum mampu merepresentasikan kondisi pertumbuhan tanaman secara visual dan intuitif bagi pengguna. Tantangan ini semakin kompleks ketika budidaya dilakukan dalam skala besar pada sistem pertanian modern berbasis *greenhouse* dan hidroponik, di mana kebutuhan akan identifikasi fase pertumbuhan yang akurat dan dapat dipahami secara visual menjadi semakin mendesak (Schneider et al., 2024).

Salah satu komoditas yang relevan untuk dikaji dalam konteks tersebut adalah melon (*Cucumis melo L.*), yaitu tanaman hortikultura bernilai ekonomi tinggi yang tumbuh optimal pada iklim tropis dan subtropis dengan suhu 25-30°C (Moing et al., 2020). Melon dapat dibudidayakan secara konvensional maupun dalam sistem modern seperti *greenhouse* dan hidroponik. Selain nilai ekonominya, melon dipilih sebagai objek penelitian karena memiliki fase pertumbuhan yang dapat diidentifikasi secara visual melalui perubahan morfologi tanaman dari fase

vegetatif hingga panen, sehingga cocok untuk dikaji menggunakan pendekatan deteksi berbasis citra.

Dalam sistem budidaya melon modern, salah satu kebutuhan mendasar adalah kemampuan merepresentasikan fase pertumbuhan tanaman secara visual dan berbasis data citra, sehingga kondisi pertumbuhan tanaman dapat dipahami secara lebih intuitif dan mudah diinterpretasikan oleh pengguna. Dalam hal ini, *computer vision* dan *deep learning* dapat dimanfaatkan untuk menentukan fase pertumbuhan tanaman melon melalui deteksi objek berbasis citra. Meskipun demikian, sebagian besar penelitian masih berfokus pada deteksi jenis tanaman secara umum, identifikasi penyakit, atau estimasi parameter tunggal seperti tinggi tanaman dan luas daun. Penelitian yang mengintegrasikan hasil deteksi tersebut ke dalam sistem visual yang mampu merepresentasikan kondisi nyata fase pertumbuhan tanaman melon dalam bentuk digital masih relatif terbatas.

Untuk menjawab permasalahan tersebut, penelitian ini mengusulkan penggunaan konsep digital model sebagai sarana representasi visual. Digital model merupakan representasi digital dari suatu objek atau sistem fisik yang dibangun berdasarkan data nyata, namun tanpa adanya koneksi data otomatis secara langsung antara entitas fisik dan digitalnya. Digital model mengandalkan pembaruan data secara manual berdasarkan hasil pengamatan atau pengukuran dari dunia nyata (Kritzinger et al., 2018). Dalam penelitian ini, digital model diwujudkan melalui representasi visual tanaman melon yang berubah mengikuti fase pertumbuhannya berdasarkan data hasil deteksi citra yang diunggah pengguna ke dalam sistem.

Penerapan metode *You Only Look Once* (YOLO) digunakan untuk mendeteksi objek tanaman melon dari citra kamera CCTV yang diunggah pengguna ke dalam sistem, kemudian hasil deteksi tersebut dimanfaatkan untuk menentukan fase pertumbuhan tanaman melon. *YOLO* dirancang untuk memproses data berbasis citra secara cepat dan akurat. Model ini memiliki kemampuan mengekstraksi fitur visual secara hierarkis, mulai dari pola sederhana seperti garis dan tekstur hingga pola yang lebih kompleks (Redmon et al., n.d.), seperti bentuk daun atau kemunculan bunga. Melalui proses *training*, *YOLO* diharapkan mampu mengenali perbedaan visual pada setiap fase pertumbuhan melon. Informasi hasil deteksi inilah yang selanjutnya menjadi dasar pembentukan representasi visual dalam bentuk *digital model*, sehingga model digital yang dibangun tidak bersifat statis, melainkan berubah mengikuti fase pertumbuhan tanaman berdasarkan citra yang diunggah pengguna ke dalam sistem.

Visualisasi representasi digital tersebut dibangun menggunakan *game engine Unity*. Sebagai salah satu *game engine* yang populer dan fleksibel, *Unity* memiliki potensi besar untuk merealisasikan visualisasi dalam bentuk model digital yang interaktif (Keil et al., 2021). *Unity* mendukung integrasi data berbasis model *machine learning*, rendering visual berkualitas tinggi, serta animasi yang memungkinkan pertumbuhan tanaman divisualisasikan dari satu fase ke fase berikutnya dengan tingkat realisme yang baik. Fitur-fitur ini memudahkan pengguna untuk mengamati perubahan fase pertumbuhan melon secara langsung, tidak hanya dalam bentuk data numerik atau tabel, tetapi juga dalam bentuk visualisasi yang menyerupai kondisi sebenarnya.

Upaya meningkatkan efisiensi pertanian melalui teknologi cerdas merupakan bagian dari tanggung jawab manusia dalam mengelola dan memanfaatkan sumber daya alam yang telah Allah ciptakan. Dalam ajaran Islam, manusia sebagai khalifah di muka bumi diberi amanah untuk menjaga, memelihara, dan memanfaatkan alam secara bijaksana. Salah satu bentuk pelaksanaan amanah tersebut adalah mengoptimalkan teknologi untuk mendukung pengelolaan pertanian yang lebih efektif, efisien, dan berkelanjutan, termasuk dalam pemantauan serta penentuan fase pertumbuhan tanaman. Hal ini sejalan dengan firman Allah *Subhanahu Wa Ta'ala*. dalam Surah Al-An'am ayat 141:

وَهُوَ الَّذِي أَنشَأَ جَنَّاتٍ مَّعْرُوشَاتٍ وَغَيْرَ مَعْرُوشَاتٍ وَالنَّخْلَ وَالزَّرْعَ مُخْتَلِفًا أَكْلُهُ وَالزَّيْتُونَ وَالرَّمَانَ مُتَشَابِهًا وَغَيْرَ مُتَشَابِهٍ كُلُوا مِنْ ثَمَرِهِ إِذَا أَثْمَرَ وَآتُوا حَقَّهُ يَوْمَ حَصَادِهِ وَلَا تُسْرِفُوا إِنَّهُ لَا يُحِبُّ الْمُسْرِفِينَ

“Dan Dialah yang menjadikan tanaman-tanaman yang merambat dan yang tidak merambat, pohon kurma, tanaman yang beraneka ragam rasanya, zaitun dan delima yang serupa (bentuk dan warnanya) dan tidak serupa (rasanya). Makanlah buahnya apa-bila ia berbuah dan berikanlah haknya (zakatnya) pada waktu memetik hasilnya, tapi janganlah berlebih-lebihan. Sesungguhnya Allah tidak menyukai orang-orang yang berlebihan”(QS. An-An'am: 141).

Menurut Tafsir Al-Muyassar yang diterbitkan oleh Kementerian Agama Saudi Arabia, manusia diperintahkan untuk memanfaatkan dan mengelola apa yang telah Allah ciptakan di muka bumi dengan penuh tanggung jawab dan tidak berlebihan (Suri & Nirwana AN, 2022). Dalam konteks penelitian ini, pemanfaatan teknologi visual dalam bentuk *digital model* dan kecerdasan buatan untuk memantau fase pertumbuhan tanaman melon merupakan salah satu bentuk pengelolaan sumber daya pertanian secara lebih efisien, akurat, dan bertanggung jawab sebagaimana yang diperintahkan.

Penelitian yang dilakukan oleh (Oktarina et al., 2025) merancang sistem klasifikasi otomatis menggunakan model *ResNet50* untuk mengidentifikasi kondisi tanaman pare (*bitter melon*) dalam lingkungan akuaponik. Penelitian tersebut menunjukkan performa yang tinggi pada data pelatihan, dengan nilai *precision* 0,92 dan *recall* 1,00. Meskipun demikian, penelitian tersebut masih berfokus pada klasifikasi kondisi tanaman dan belum mengarah pada deteksi fase pertumbuhan tanaman melon yang terintegrasi dengan visualisasi dalam bentuk *digital model*.

Selanjutnya, penelitian yang dilakukan oleh (Murtaza et al., 2023) mengembangkan sistem visualisasi data interaktif menggunakan *Unity 3D framework* yang memungkinkan penyajian data dari file CSV dalam lingkungan tiga dimensi imersif. Sistem ini memanfaatkan *Python API* dan *headset Meta Quest 2* untuk merepresentasikan data secara visual dalam ruang virtual. Namun, penelitian tersebut belum menyinggung visualisasi 3D fase tanaman melon dan penggunaan algoritma *deep learning* untuk deteksi fase pertumbuhan tanaman melon berbasis citra guna memvisualisasikan perubahan pertumbuhan tanaman secara dinamis.

Selain itu, pendekatan klasifikasi fase pertumbuhan tanaman berbasis fenotipe secara otomatis juga telah dikaji oleh (Kim et al., 2025) Penelitian ini menggabungkan YOLOv8, *K-means clustering*, dan regresi CNN berbasis ResNet-18 untuk mengklasifikasikan fase pertumbuhan basil dalam lingkungan multi-tanaman dengan akurasi lebih dari 98%. Pendekatan tersebut mampu menghasilkan klasifikasi fase yang akurat dan reproduibel, tetapi masih berfokus pada satu jenis tanaman herba dan belum mengintegrasikan hasil deteksi ke dalam platform

visualisasi tiga dimensi interaktif untuk merepresentasikan pertumbuhan tanaman melon secara imersif.

Berdasarkan uraian tersebut, masih terdapat celah penelitian pada integrasi antara deteksi fase pertumbuhan tanaman melon berbasis citra menggunakan *YOLOv8* dengan *digital model* yang mampu memvisualisasikan perubahan pertumbuhan secara dinamis dan interaktif berdasarkan citra yang diunggah pengguna. Oleh karena itu, penelitian ini bertujuan membangun visualisasi *digital model* pertumbuhan tanaman melon berdasarkan hasil deteksi *YOLOv8* yang direpresentasikan dalam lingkungan *Unity*, sehingga fase pertumbuhan tanaman melon dapat dipahami secara lebih akurat, terstruktur, dan mudah dipahami.

1.2 Rumusan Masalah

Bagaimana melakukan visualisasi perubahan fase pertumbuhan tanaman melon dalam bentuk *digital model* berdasarkan hasil deteksi *YOLOv8* menggunakan *Unity* dari citra kamera CCTV yang diunggah pengguna ?

1.3 Batasan Masalah

Untuk memperoleh hasil yang optimal, penelitian ini menetapkan beberapa batasan, diantaranya :

1. Daerah penelitian yang dilakukan hanya di *greenhouse* kelompok yang berada di desa tirtomoyo, kecamatan pakis, kabupaten malang.
2. Penelitian ini berfokus pada visualisasi perubahan fase pertumbuhan tanaman melon jenis “*Golden Melon*” meliputi fase *early* vegetatif, *late* vegetatif, pembungaan, pertumbuhan buah, dan panen berdasarkan hasil

deteksi *YOLOv8* tanpa mencakup analisis data lain, seperti estimasi ukuran buah, atau berat tanaman dan tanpa mempertimbangkan aspek gamifikasi atau interaktifitas lainnya yang lebih mendalam.

3. Representasi visual terbatas pada perubahan fase pertumbuhan tanaman melon di *greenhouse* kelompok secara umum berdasarkan hasil deteksi *YOLOv8* dari citra kamera CCTV yang diunggah pengguna tanpa mencakup perbedaan fase setiap tanaman.
4. Pengambilan citra untuk *dataset* dilakukan menggunakan kamera CCTV yang diposisikan pada sudut atas *greenhouse*, dengan tujuan agar seluruh tanaman di dalam greenhouse dapat tercakup dalam citra yang digunakan.

1.4 Tujuan Penelitian

Berdasarkan masalah diatas maka tujuan dari penelitian untuk mengetahui cara melakukan visualisasi perubahan fase pertumbuhan tanaman melon dalam bentuk *digital model* berdasarkan hasil deteksi *YOLOv8* Menggunakan *Unity* dari citra kamera CCTV yang diunggah pengguna.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan solusi inovatif untuk mempermudah pemantauan fase pertumbuhan tanaman melon jenis “*Golden Melon*” berdasarkan hasil deteksi *YOLOv8* dalam bentuk *digital model* melalui citra kamera CCTV yang diunggah pengguna, sehingga mendukung efisiensi dan akurasi dalam pengelolaan pertanian dan menampilkan visualisasi pertumbuhan tanaman yang lebih dinamis dan interaktif.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Penelitian terkait dikaji untuk memahami perkembangan topik penelitian sebelumnya yang dapat dijadikan acuan serta sebagai pembandingan dengan penelitian yang sedang dilakukan. Kajian difokuskan pada dua kelompok penelitian, yaitu penelitian yang menerapkan teknologi visual *digital model* sebagai representasi visual objek fisik, serta penelitian yang menggunakan algoritma *deep learning* berbasis *YOLO* untuk deteksi objek menggunakan *unity engine*.

Penelitian (Murtaza et al., 2023) mengembangkan sistem visualisasi data interaktif menggunakan *Unity 3D framework* yang memungkinkan penyajian data dari file CSV dalam lingkungan tiga dimensi imersif. Sistem ini memanfaatkan Python API dan headset Meta Quest 2 untuk merepresentasikan data secara visual dalam ruang virtual. Hasil pengembangan menunjukkan bahwa Unity 3D efektif sebagai platform representasi visual data secara interaktif.

(Kim et al., 2025) mengembangkan pipeline terintegrasi untuk klasifikasi fase pertumbuhan tanaman basil berbasis fenotipe tingkat instance pada lingkungan multi-tanaman menggunakan citra *top-view* dari sembilan *growth chamber* dengan total 8.519 citra. *Pipeline* yang diusulkan menggabungkan YOLOv8 untuk deteksi tanaman secara otomatis, *K-means clustering* untuk pengurutan posisi individu, serta model regresi CNN berbasis ResNet-18 untuk mengestimasi jumlah pasangan daun sebagai indikator fase pertumbuhan yang dikonversi ke tiga level berdasarkan

skala BBCH. Hasil evaluasi menunjukkan $mAP@0.5$ sebesar 0,995 pada tahap deteksi, MAE sebesar 0,13, R^2 sebesar 0,96, dan akurasi klasifikasi fase pertumbuhan melebihi 98% dengan $F1-score$ sebesar 0,98.

(Buyuksalih et al., 2017) mengkaji penerapan Unity Game Engine sebagai platform pemodelan dan visualisasi 3D untuk merepresentasikan objek fisik secara digital. Penelitian ini membahas keunggulan dan tantangan penggunaan Unity dalam konteks pemodelan representasi digital objek fisik, termasuk integrasi basis data dan performa pada dataset berskala besar. Hasil penelitian menunjukkan bahwa Unity mampu menjadi platform visualisasi digital model yang efektif, namun implementasinya menghadapi kendala pada dataset besar.

(Arif & Harini, 2021) mengembangkan sistem *Automatic Scenario Control* (ASC) dalam serious game pariwisata untuk memvisualisasikan rekomendasi destinasi wisata di Mojokerto, Jawa Timur menggunakan Unity *game engine*. Sistem ini menggabungkan *Hierarchical Finite State Machine* (HFSM) pada tahap perancangan skenario berbasis tantangan sesuai jenis atraksi wisata, serta *Dynamic Weight TOPSIS* (DWT) pada tahap seleksi skenario berdasarkan ekspektasi pemain terhadap kriteria destinasi. Hasil pengujian pada 32 wisatawan potensial menunjukkan akurasi 0,78, *precision* 0,80, *recall* 0,96, dan $F1-score$ 0,87, dengan skor rata-rata aspek media 7,8 dan aspek materi 7,9 dari skala 10.

(Santoso et al., 2021) mengimplementasikan kombinasi *CNN* dan *YOLOv4-tiny* untuk sistem pendeteksi uang kertas rupiah berbasis *real-time* yang ditujukan bagi penyandang *low vision*. Model dilatih menggunakan 1.400 gambar dalam 14

kelas dan menghasilkan *mAP* 97,5% pada data uji serta 88% pada variasi kondisi fisik dan pencahayaan.

(Aini et al., 2024) mengembangkan model klasifikasi bunga menggunakan *Convolutional Neural Network* (CNN) dengan pendekatan dari *scratch* dan *transfer learning* berbasis ResNet50, serta optimasi *Nadam* untuk meningkatkan efisiensi pelatihan. *Dataset* yang digunakan adalah Oxford17 (17 jenis bunga, 1.360 citra) dan Oxford102 (102 jenis bunga, 8.189 citra) dengan pembagian 80% data latih dan 20% data uji. Hasil penelitian menunjukkan akurasi 60% dan 84% pada dataset Oxford17 masing-masing untuk pendekatan *scratch* dan *transfer learning*, sedangkan pada *dataset* Oxford102 diperoleh akurasi 42% dan 64%, dengan CNN *transfer learning* yang lebih unggul dibandingkan SVM dan ANN pada dataset Oxford17.

(Jupiter et al., 2023) mengimplementasikan *CNN* dan *YOLO* untuk mendeteksi jenis kendaraan dari citra video lalu lintas menggunakan *dataset* 484 gambar dalam empat kelas. Model *YOLO* menghasilkan *precision* 0,936, *recall* 0,914, dan *F1-score* 0,924, dan disimpulkan unggul dalam kecepatan deteksi *real-time* dibandingkan *CNN* yang lebih akurat pada citra penuh yang stabil.

(Roslan & Ibrahim, 2023) mengimplementasikan Faster R-CNN dengan jaringan pralatih FPN30 untuk deteksi dan pengenalan buah secara otomatis pada dua skenario dataset, yaitu *inter-class* (apel dan jeruk) dan *intra-class* (apel *Fuji* dan *Royal Gala*). *Dataset inter-class* terdiri dari 250 citra latih dan 24 citra uji, sedangkan dataset *intra-class* terdiri dari 290 citra latih dan 31 citra uji yang dianotasi menggunakan platform Roboflow. Hasil evaluasi menunjukkan nilai

Average Precision (AP) sebesar 91,443 pada *dataset inter-class* dan AP 57,486 pada *dataset intra-class*, dengan model yang mampu mendeteksi dan melokalisasi objek buah secara akurat melalui *bounding box*.

Rangkuman seluruh penelitian terkait yang dijadikan acuan dalam pengembangan visualisasi *digital model* fase pertumbuhan tanaman melon berbasis *YOLOv8* disajikan pada Tabel 2.1 berikut.

Tabel 2.1 Penelitian Terkait

No.	Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian
1.	(Murtaza et al., 2023)	<i>Development of Interactive Data Visualization System in Unity 3D Framework</i>	Visualisasi data 3D berbasis <i>Unity</i> , <i>Python API</i> , <i>VR Meta Quest 2</i> .	Unity 3D efektif sebagai platform visualisasi data interaktif dalam lingkungan imersif.
2.	(Kim et al., 2025)	<i>Instance-level Phenotype-based Growth Stage Classification of Basil in Multi-plant Environments</i>	YOLOv8 (deteksi tanaman), <i>K-means clustering</i> , CNN ResNet-18 regresi (estimasi pasangan daun), 8.519 citra, 9 <i>growthchamber</i> .	$mAP@0.5 = 0,995$, akurasi klasifikasi fase >98%, MAE = 0,13, $R^2 = 0,96$. <i>pipeline</i> otomatis non-destruktif pada lingkungan multi-tanaman.
3.	(Buyuksalih et al., 2017)	<i>3D Modelling and Visualization Based on the Unity Game Engine – Advantages and Challenges</i>	Pemodelan dan visualisasi 3D berbasis <i>Unity Game Engine</i> .	Unity efektif sebagai platform digital model, kendala pada dataset besar.
4.	(Arif & Harini, 2021)	<i>An Automatic Scenario Control in Serious Game to Visualize Tourism Destinations Recommendation</i>	HFSM (perancangan skenario), <i>Dynamic Weight TOPSIS</i> (seleksi skenario), <i>Unity game engine</i> .	Akurasi 0,78, <i>precision</i> 0,80, <i>recall</i> 0,96, <i>F1-score</i> 0,87; skor aspek media 7,8/10, aspek materi 7,9/10.
5.	(Santoso et al., 2021)	<i>Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network dan Algoritma YOLO dalam Sistem Pendeteksi Uang Kertas Rupiah bagi Penyandang Low Vision</i>	CNN dan YOLOv4-tiny, 1.400 citra, 14 kelas.	mAP 97,5% (data uji), mAP 88% pada variasi kondisi.
6.	(Aini et al., 2024)	<i>Applying Convolutional Neural Network and Nadam</i>	CNN (<i>scratch & ResNet50 transfer learning</i>), Nadam <i>optimization</i> , <i>dataset</i>	Oxford17: akurasi 60% (<i>scratch</i>), 84% (<i>transfer learning</i>); Oxford102:

No.	Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian
		<i>Optimization in Flower Classification</i>	Oxford17 & Oxford102.	42% (<i>scratch</i>), 64% (<i>transfer learning</i>).
7.	(Jupiter et al., 2023)	<i>Implementasi Algoritma CNN dan YOLO untuk Mendeteksi Jenis Kendaraan pada Jalan Raya</i>	CNN dan YOLO, 484 citra, 4 kelas.	YOLO: <i>precision</i> 0,936, <i>recall</i> 0,914, <i>F1</i> 0,924.
8.	(Roslan & Ibrahim, 2023)	<i>Fruit Detection and Recognition Using Faster RCNN with FPN30 Pre-trained Network</i>	Faster R-CNN, FPN30 <i>pre-trained network</i> , <i>dataset inter-class</i> dan <i>intra-class</i> buah.	AP 91,443 (<i>inter-class</i>), AP 57,486 (<i>intra-class</i>); model akurat dalam deteksi dan lokalisasi objek buah.

Berdasarkan penelitian terkait, digital model terbukti efektif sebagai landasan representasi visual objek fisik dalam lingkungan virtual, baik melalui visualisasi data interaktif berbasis Unity, pemodelan 3D objek fisik, maupun representasi pertumbuhan tanaman secara prosedural. Studi-studi tersebut menegaskan nilai tambah digital model dalam merepresentasikan objek fisik ke dalam bentuk visual digital. Namun, celah yang masih terbuka adalah pengenalan fase pertumbuhan berbasis deteksi citra sebagai state yang dapat dipetakan langsung ke representasi 3D pada visualisasi digital model.

Selanjutnya, berdasarkan penelitian terkait, penerapan algoritma YOLO memiliki kinerja yang baik dalam deteksi objek secara *real-time*. Penelitian oleh (Santoso et al., 2021) yang mengimplementasikan *YOLOv4-tiny* untuk deteksi uang kertas rupiah menghasilkan nilai *mean Average Precision* (mAP) mencapai 97,5% pada data uji, bahkan pada variasi kondisi fisik dan pencahayaan tetap mencapai mAP 88%. Sementara itu, penelitian (Jupiter et al., 2023) mengungkapkan bahwa YOLO unggul dalam kecepatan deteksi objek secara *real-time*, bahkan ketika sebagian objek tertutup, dengan nilai *precision* 0,936, *recall* 0,914, dan *F1-score*

0,924. Dengan demikian, YOLO dinilai unggul untuk mendeteksi objek pada citra yang memerlukan kecepatan inferensi tinggi dalam kondisi lingkungan yang dinamis.

2.2 Melon

Melon (*Cucumis melo L.*) merupakan tanaman hortikultura dari famili *Cucurbitaceae* yang bernilai ekonomi tinggi dan relatif mudah dibudidayakan. Tanaman semusim ini tumbuh merambat dengan sistem perakaran serabut serta memiliki keragaman varietas berdasarkan bentuk buah, warna daging, dan tingkat kemanisan (Wei et al., 2023). Secara agroklimat, melon tumbuh optimal pada iklim tropis dan subtropis dengan suhu 25-30°C, intensitas cahaya yang cukup, dan media tanam berdrainase baik (Moing et al., 2020). Melon dapat dibudidayakan secara konvensional maupun dalam sistem modern seperti *greenhouse* dan hidroponik yang memungkinkan pengendalian lingkungan tumbuh secara lebih terkontrol.

2.2.1 Fase Pertumbuhan Tanaman Melon

Fase pertumbuhan tanaman melon berlangsung bertahap dan dipengaruhi oleh faktor internal seperti genetik dan hormon pertumbuhan, serta faktor eksternal seperti cahaya, suhu, kelembapan, dan ketersediaan nutrisi (Rif'an et al., 2024). Secara umum, pertumbuhan melon terbagi menjadi lima fase utama sebagai berikut.

a. Fase Semai

Fase semai dimulai sejak biji berkecambah hingga terbentuk dua daun sejati. Pada fase ini media tanam yang ideal bersifat gembur, lembap, dan tidak tergenang.

b. Fase Vegetatif (Pembentukan Daun dan Pertumbuhan)

Pada fase vegetatif, tanaman membentuk akar, batang, daun, percabangan, dan sulur. Fase ini membutuhkan asupan air dan unsur hara, terutama nitrogen, dalam jumlah yang lebih besar dibandingkan fase lainnya.

c. Fase Pembungaan

Pada fase pembungaan, tanaman mulai menghasilkan bunga jantan dan bunga betina sebagai tanda peralihan dari fase vegetatif ke fase generatif.

d. Fase Generatif (Pembentukan dan Pembesaran Buah)

Setelah penyerbukan berhasil, buah muda terbentuk dan berkembang pada fase pembentukan serta pembesaran buah. Fase ini ditandai dengan pertumbuhan ukuran buah yang pesat hingga mencapai ukuran optimalnya.

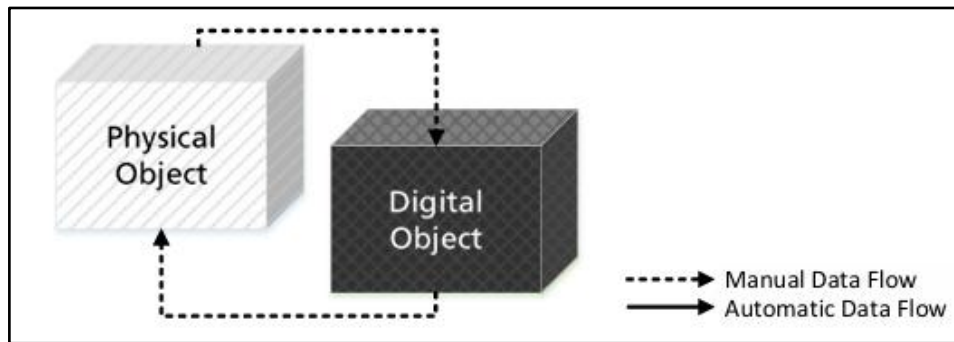
e. Fase Panen (Pematangan Buah)

Pada fase pematangan buah, pertumbuhan ukuran buah mulai melambat dan terjadi akumulasi gula, pembentukan aroma khas, serta perubahan warna kulit dan daging buah tergantung varietasnya. Panen umumnya dilakukan 50-65 hari setelah penyerbukan, tergantung varietas dan kondisi budidaya.

2.3 Digital Model

Digital Model merupakan representasi digital dari suatu objek fisik yang ada maupun yang direncanakan, namun tidak melibatkan pertukaran data otomatis antara objek fisik dan objek digital. Seluruh pertukaran data dilakukan secara manual, sehingga perubahan kondisi pada objek fisik tidak berdampak langsung pada objek digital, begitu pula sebaliknya. Contoh dari kategori ini mencakup model simulasi pabrik yang direncanakan, model matematis produk baru, atau

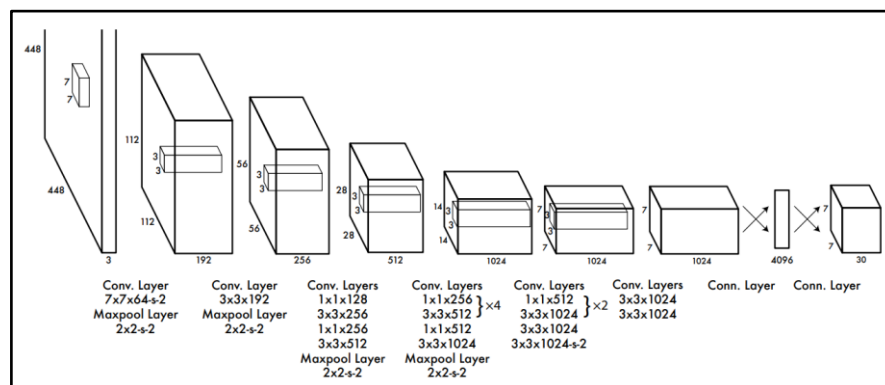
representasi lain dari objek fisik yang tidak menggunakan integrasi data otomatis (Kritzing et al., 2018). Ilustrasi aliran data pada *Digital Model* dapat dilihat pada Gambar 2.1 berikut.



Gambar 2. 1 Diagram Alir di *Digital Model* oleh (Kritzing et al., 2018)

2.4 *You Only Look Once (YOLO)*

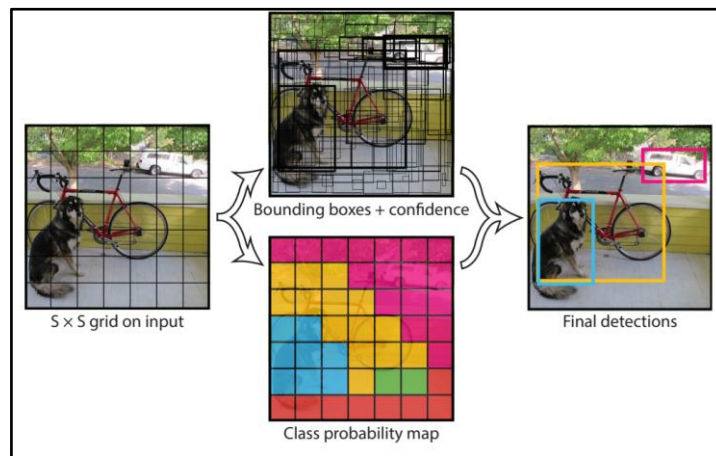
Algoritma *YOLO* (*You Only Look Once*) merupakan algoritma *computer vision* yang dirancang untuk mendeteksi objek secara cepat dan efisien dalam satu kali proses inferensi. Arsitektur *YOLO* ditunjukkan pada Gambar 2.2 berikut.



Gambar 2. 2 Arsitektur *YOLO* oleh (Redmon et al., n.d.)

Berbeda dengan pendekatan deteksi objek konvensional yang memproses gambar secara bertahap, *YOLO* memproses seluruh gambar sekaligus dalam satu jaringan syaraf tunggal. *YOLO* membagi gambar *input* menjadi grid $S \times S$, di mana setiap *sel grid* bertanggung jawab mendeteksi objek yang pusatnya jatuh pada sel

tersebut. Setiap sel memprediksi *bounding box* B beserta *confidence score* dan probabilitas kelas kondisional C . *Confidence score* menggambarkan seberapa akurat prediksi kotak terhadap objek yang terdeteksi, apabila tidak ada objek pada sel tersebut, nilai *confidence* akan bernilai nol. Proses deteksi *YOLO* ditunjukkan pada Gambar 2.3 berikut.



Gambar 2. 3 Proses Deteksi pada Algoritma *YOLO* oleh (Redmon et al., n.d.)

Penentuan prediksi akhir didasarkan pada *class confidence score* yang mengukur tingkat kepercayaan terhadap klasifikasi dan lokalisasi objek secara bersamaan, sebagaimana ditunjukkan pada Persamaan 2.1.

$$Pr(Class_i | Object) \times Pr(Object) \times IoU_{pred}^{truth} = Pr(Class_i) \times IoU_{pred}^{truth} \quad (2.1)$$

Keterangan:

$Pr(Class_i | Object)$ = Probabilitas kondisional kelas i
 $Pr(Object)$ = probabilitas kelas i

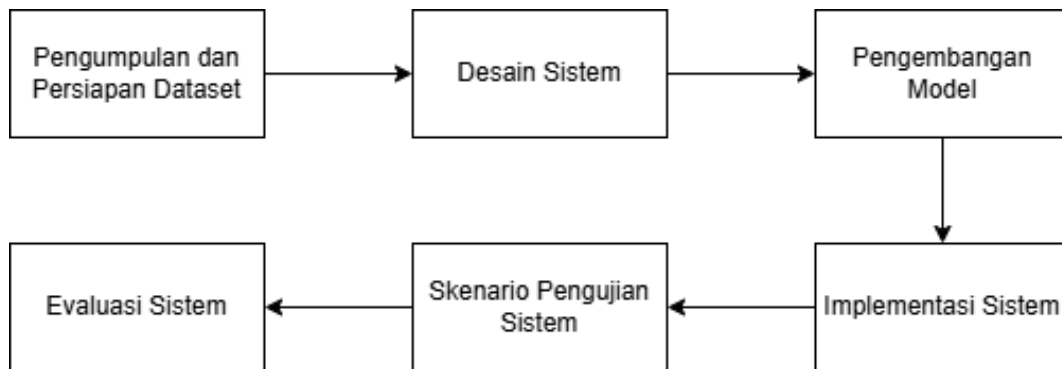
Dari persamaan tersebut, akan mendapatkan nilai *confidence* dari kelas spesifik. Nilai *class confidence score* merepresentasikan probabilitas kelas yang muncul dalam kotak prediksi sekaligus mengukur seberapa akurat posisi kotak yang diprediksi terhadap objek yang sebenarnya.

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Desain Penelitian

Desain penelitian merupakan rangkaian tahapan yang disusun secara sistematis untuk memastikan penelitian berjalan secara efektif dan terarah, mulai dari pengumpulan *dataset* hingga evaluasi secara keseluruhan. Alur desain penelitian ini disajikan pada Gambar 3.1 berikut.



Gambar 3. 1 Desain Penelitian

Tahap awal meliputi pengumpulan dan persiapan *dataset*, yang mencakup pengambilan citra melalui kamera CCTV, *pre-processing*, anotasi, augmentasi, serta pembagian *dataset* menggunakan platform *Roboflow*.

Setelah *dataset* disiapkan, tahap berikutnya adalah perancangan sistem (desain sistem). Pada tahap ini, peneliti menyusun arsitektur sistem visualisasi *digital model* secara menyeluruh, meliputi alur *input* citra kamera CCTV oleh pengguna, proses deteksi menggunakan model *YOLOv8*, mekanisme penentuan fase untuk menentukan fase pertumbuhan, serta keluaran berupa visualisasi *prefab 3D* di *Unity*.

Tahap selanjutnya adalah pengembangan model, yaitu pelatihan *YOLOv8* menggunakan *dataset* yang telah disiapkan di *Google Colab*, evaluasi hasil pelatihan berdasarkan metrik akurasi, dan ekspor model ke format *Open Neural Network Exchange (ONNX)* untuk integrasi dengan *Unity*. Setelah model berhasil dikembangkan, dilakukan implementasi sistem melalui integrasi model *ONNX* ke *Unity Sentis*, pengembangan *prefab 3D digital model* di *Blender*, serta konfigurasi *scene Unity* secara keseluruhan.

Untuk memastikan sistem berfungsi dengan baik, dilakukan pengujian yang mencakup akurasi deteksi model *YOLOv8*, logika penentuan fase pertumbuhan, dan visualisasi *prefab digital model* berdasarkan hasil deteksi di *Unity*. Tahap terakhir adalah evaluasi pengguna menggunakan metode *User Version of the Mobile Application Rating Scale (uMARS)* untuk menilai kualitas aplikasi dari sudut pandang pengguna pada platform mobile.

3.2 Pengumpulan dan Persiapan *Dataset*

Dataset yang digunakan merupakan data primer berupa *snapshot CCTV* dari *greenhouse* kelompok, yang mencakup lima kelas fase pertumbuhan tanaman melon, yaitu *early vegetatif*, *late vegetatif*, pembungaan, pertumbuhan buah, dan panen. *Dataset* diambil dari *log citra CCTV* yang tersimpan di *server* pada periode tanam pertama dan periode tanam kedua. Jadwal periode tanam dan pengambilan *dataset* dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Jadwal Periode Tanam dan Pengambilan *Dataset*

No.	Periode Tanam	Waktu Tanam	Waktu Panen	Hitungan HST	Gambar yang Dikumpulkan
1.	Periode tanam pertama	22-08-2025	31-10-2025	69 HST	168 Gambar
2.	Periode tanam kedua	05-12-2025	17-02-2026	74 HST	171 Gambar
Total Gambar					339 Gambar

Berdasarkan Tabel 3.1, pengumpulan *dataset* dilakukan selama dua periode tanam, yaitu periode pertama dari 22 Agustus 2025 hingga 31 Oktober 2025 (69 HST) dan periode kedua dari 5 Desember 2025 hingga 17 Februari 2026 (74 HST), dengan total 339 gambar yang diperoleh dari *log server CCTV greenhouse*. Pengambilan data hanya dilakukan pada waktu pagi (pukul 08:00 WIB) dan siang hari (pukul 13.05 WIB) karena kondisi pencahayaan alami pada kedua waktu tersebut lebih memadai untuk menangkap karakteristik visual tanaman, seperti bentuk daun, kemunculan bunga, dan perubahan warna buah. Gambar malam hari tidak digunakan karena pencahayaan di dalam greenhouse yang kurang baik tidak dapat merepresentasikan informasi warna secara akurat, khususnya warna kuning keemasan sebagai penanda utama fase panen. Perbedaan kondisi pencahayaan antara pagi dan malam pada hari yang sama diilustrasikan pada Gambar 3.2.



Gambar 3. 2 Contoh Pengambilan Gambar CCTV di waktu pagi (pukul 08:05 WIB) dan malam (22:05 WIB)

Setelah mengecualikan pengambilan citra CCTV pada malam hari, total dataset yang dikumpulkan berjumlah 199 gambar yang mencakup fase *early vegetatif*, *late vegetatif*, pembungaan, pertumbuhan buah, dan panen. Kelima kelas tersebut dipilih untuk memastikan seluruh fase pertumbuhan terwakili secara lengkap sebagai *state* dalam sistem visualisasi *digital model*. Seluruh tahapan

persiapan *dataset* meliputi *pre-processing dataset*, anotasi, augmentasi, dan pembagian *dataset* dilakukan menggunakan platform *Roboflow*. Rangkuman penentuan kelas dan contoh citra per kelas disajikan pada Tabel 3.2 berikut.

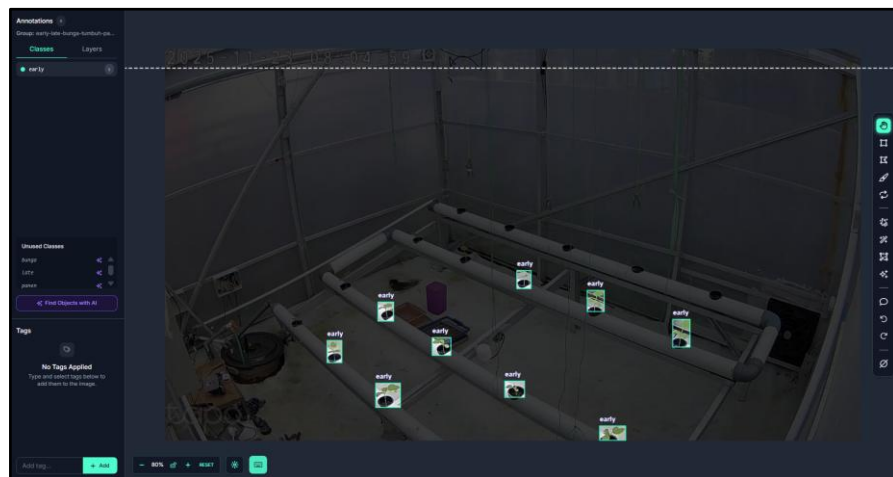
Tabel 3.2 Kriteria Fase Pertumbuhan Tanaman Melon

No.	Kriteria Fase Pertumbuhan Melon	Citra gambar	Deskripsi
1.	Fase early vegetatif		Ditandai daun yang belum berbentuk sempurna dan masih berukuran kecil.
2.	Fase late vegetatif		Ditandai dengan daun sudah lebat dan sudah membentuk menjari sempurna, tetapi belum memiliki bunga.
3.	Fase Pembungaan (bunga)		Ditandai dengan tanaman mulai mengeluarkan bunga
4.	Fase pertumbuhan buah (tumbuh)		Ditandai dengan terbentuknya buah melon yang masih membesar, tetapi warna buah masih kehijauan.
5.	Fase panen		Ditandai saat buah melon telah berwarna kuning keemasan sempurna sesuai karakteristik varietas "golden melon".

3.2.1 Proses Anotasi (*Labelling*) *Dataset*

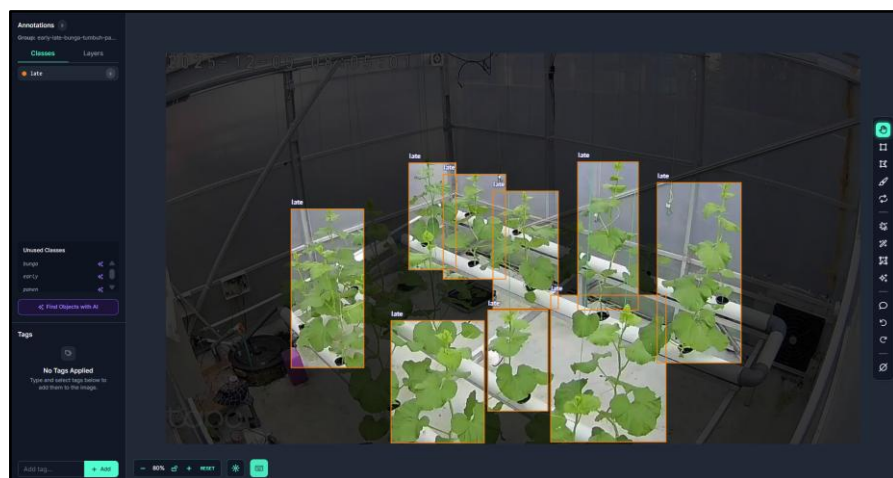
Dataset yang sudah dikumpulkan, selanjutnya dilakukan proses anotasi atau pelabelan objek pada setiap gambar menggunakan *tool* yang tersedia di *Roboflow* dengan metode manual *semi-otomatis*. Setiap gambar diberi *bounding box* yang

merepresentasikan objek tanaman, kemudian diberi *label* sesuai nama kelas fase pertumbuhannya. Proses ini dilakukan secara teliti untuk memastikan setiap bounding box memiliki posisi dan ukuran yang tepat sesuai batas visual objek tanaman pada gambar. Contoh proses anotasi objek pada kelas *early* disajikan pada Gambar 3.3 berikut.



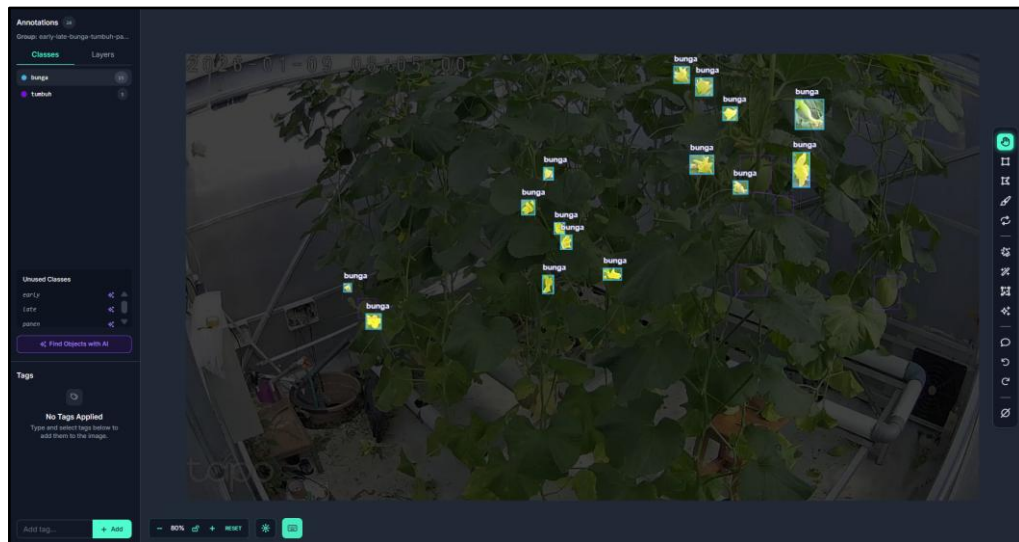
Gambar 3. 3 Proses Anotasi (*Labelling*) Pada Kelas *Early*

Gambar 3.3 diatas meruapakan proses anotasi untuk kelas *early*, Dimana dengan memberikan *bounding box* pada keseluruhan satu tanaman yang memiliki bentuk daun belum berbentuk sempurna dan masih berukuran kecil.



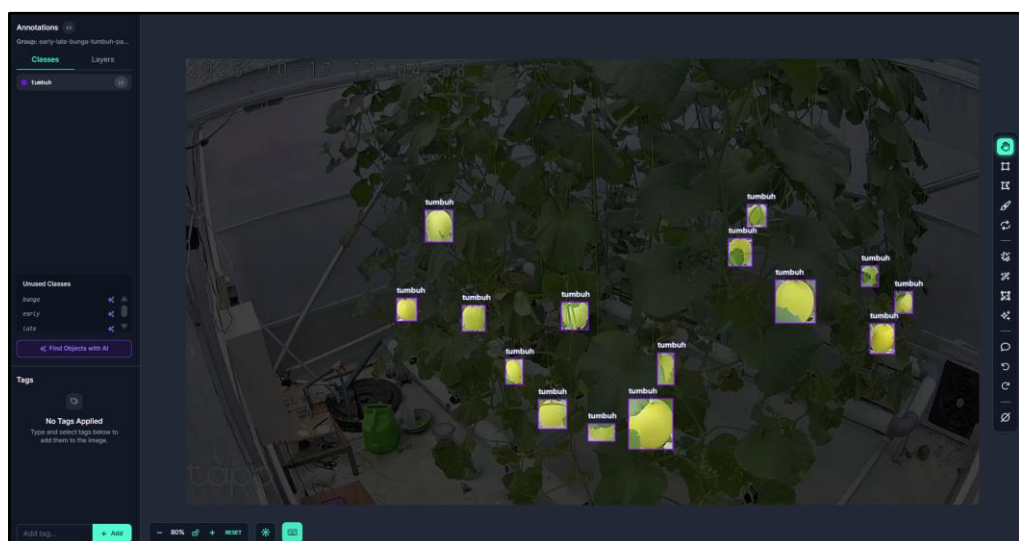
Gambar 3. 4 Proses Anotasi (*Labelling*) Pada Kelas *Late*

Gambar 3.4 diatas merupakan proses anotasi untuk kelas *late*, Dimana dengan memberikan *bounding box* pada keseluruhan satu tanaman yang memiliki daun sudah berbentuk menjari sempurna.



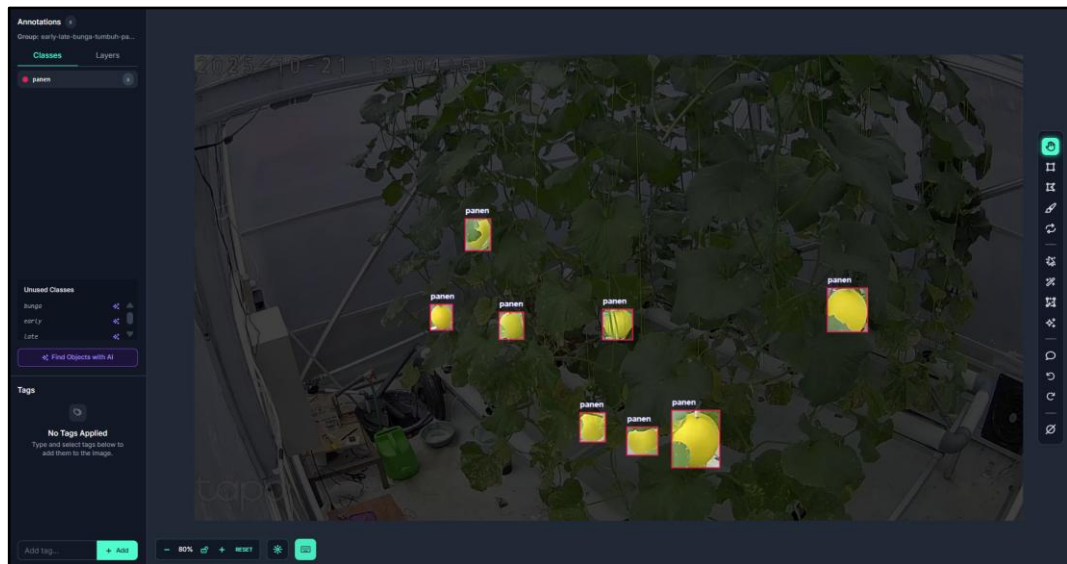
Gambar 3. 5 Proses Anotasi (*Labelling*) Pada Kelas Bunga

Gambar 3.5 diatas merupakan proses anotasi untuk kelas bunga, Dimana dengan memberikan *bounding box* pada objek bunga pada setiap tanaman yang sudah tumbuh bunga jantan atau bunga betina.



Gambar 3. 6 Proses Anotasi (*Labelling*) Pada Kelas Tumbuh

Gambar 3.6 diatas merupakan proses anotasi untuk kelas tumbuh, Dimana dengan memberikan *bounding box* pada objek buah tanaman melon yang masih berwarna kehijauan dan berukuran kecil.



Gambar 3. 7 Proses Anotasi (*Labelling*) Pada Kelas Panen

Gambar 3.7 diatas merupakan proses anotasi untuk kelas panen, Dimana dengan memberikan *bounding box* pada objek buah tanaman melon yang sudah berwarna kuning keemasan sesuai dengan karakteristik varietas jenis “golden melon”.

Distribusi jumlah gambar yang memiliki anotasi pada setiap kelas dan jumlah anotasi pada seluruh *dataset* ditunjukkan pada Tabel 3.3 berikut.

Tabel 3.3 Distribusi Jumlah Gambar dan Anotasi Setiap Kelas

No.	Kelas	Gambar ter-Anotasi	Jumlah Anotasi (BB)
1.	Fase <i>early</i> vegetatif	35 Gambar	303
2.	Fase <i>late</i> vegetatif	32 Gambar	283
3.	Fase Pembungaan (bunga)	62 Gambar	418
4.	Fase pertumbuhan buah (tumbuh)	119 Gambar	533
5.	Fase panen	66 Gambar	237
Jumlah Total Anotasi (BB)			1.774

Dataset terdiri dari total 1.774 anotasi *bounding box* yang terdistribusi ke dalam 5 kelas. Satu gambar dapat memiliki anotasi lebih dari satu kelas, sehingga jumlah gambar per kelas (314 total) lebih besar dari jumlah gambar unik atau gambar asli dataset yang berjumlah 199 gambar.

3.2.2 *Pre-Processing Dataset*

Setelah proses pemberian *label* (anotasi) dilakukan, selanjutnya *pre-processing dataset* dilakukan menggunakan *platform Roboflow* untuk memastikan seluruh citra memiliki format, ukuran, dan kualitas yang seragam sebelum digunakan dalam proses pelatihan model *YOLOv8*. Teknik *pre-processing* yang digunakan adalah sebagai berikut.

1. *Resize*

Resize merupakan teknik untuk mengubah dimensi citra agar seluruh data memiliki ukuran yang seragam dan konsisten. Dimensi citra diubah menjadi 640x640 piksel.

2. *Auto-Orient*

Auto-Orient merupakan teknik untuk mendeteksi dan memperbaiki orientasi citra secara otomatis berdasarkan metadata *EXIF* (*Exchangeable Image File Format*), sehingga seluruh citra dalam *dataset* memiliki orientasi yang sesuai dengan kondisi sebenarnya sebelum tahap pelatihan berlangsung.

3.2.3 *Pembagian Dataset (Splitting Dataset)*

Setelah tahapan *pre-processing dataset* selesai, *dataset* dibagi ke dalam data *training*, *validation*, dan *testing* dengan rasio 80:10:10. Rasio ini dipilih karena umum digunakan pada penelitian *object detection* berbasis *YOLO* dengan jumlah

data yang terbatas. Pembagian dilakukan secara otomatis menggunakan *platform roboflow* dengan memperhatikan pemerataan distribusi kelas pada setiap bagian, Distribusi jumlah gambar ter-anotasi per kelas setelah proses *split dataset* disajikan pada Tabel 3.4 berikut.

Tabel 3.4 Distribusi Jumlah Gambar ter-Anotasi per Kelas Setelah Proses *Split Dataset*

No.	Kelas	Train	Valid	Test	Total
1.	Fase <i>early</i> vegetatif	25	6	4	35
2.	Fase <i>late</i> vegetatif	25	4	3	32
3.	Fase Pembungaan (bunga)	45	8	9	62
4.	Fase pertumbuhan buah (tumbuh)	94	12	13	119
5.	Fase panen	51	5	10	66
Total Gambar (<i>Unique</i>)		159	20	20	199

Data *training* digunakan untuk melatih model, data *validation* untuk memantau performa selama pelatihan dan mencegah *overfitting*, serta data *testing* untuk mengukur kinerja akhir model menggunakan data yang belum pernah dilihat sebelumnya.

3.2.4 Augmentasi *Dataset*

Augmentasi data dilakukan menggunakan platform *Roboflow* untuk memperbanyak jumlah data citra gambar mengingat jumlah data *primer* yang diperoleh masih terbatas. Teknik augmentasi hanya diterapkan pada data *training*, sedangkan data *validation* dan *testing* tetap menggunakan data asli tanpa augmentasi. Teknik augmentasi yang digunakan adalah *rotate*, Teknik ini dipilih karena penerapannya tidak mengubah karakteristik visual objek tanaman secara signifikan sehingga label kelas tetap valid setelah augmentasi dilakukan.

Teknik *rotate* merotasi citra secara acak dalam rentang -15° hingga $+15^{\circ}$, serta rotasi tetap sebesar 90° searah jarum jam (*Clockwise*), 90° berlawanan arah

jarum jam (*Counter-Clockwise*), dan 180° (*Upside Down*). Setiap gambar menghasilkan 3 output per contoh pelatihan, sehingga jumlah data bertambah secara signifikan. Contoh teknik *rotate* disajikan pada Gambar 3.8 berikut.



Gambar 3. 8 Teknik *Rotate*

Distribusi total gambar pada *dataset* setelah dilakukan proses augmentasi dan *split dataset* dapat dilihat pada Tabel 3.5 berikut.

Tabel 3.5 Distribusi Jumlah Gambar ter-anotasi per Kelas sebelum proses *Augmentasi* dan *Split Dataset*

Split	Jumlah Gambar
<i>Train (output 3x per gambar)</i>	477
<i>Valid</i>	20
<i>Test</i>	20
Total <i>Dataset</i>	517

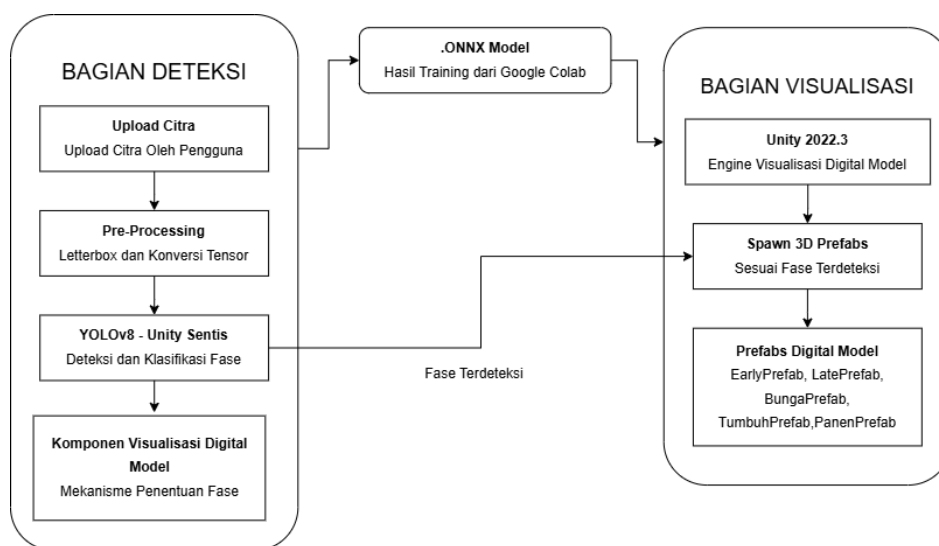
3.3 Desain Sistem

Desain sistem dalam penelitian ini disusun untuk menggambarkan alur perancangan sistem visualisasi *digital model* yang mampu merepresentasikan fase pertumbuhan tanaman melon secara visual berdasarkan hasil deteksi *YOLOv8* terhadap citra kamera CCTV yang diunggah pengguna ke dalam sistem. Perancangan ini mencakup empat aspek utama yang saling berkaitan, yaitu arsitektur umum sistem, alur sistem, mekanisme penentuan fase pertumbuhan, dan

perancangan *prefab digital model* sebagai representasi visual setiap fase pertumbuhan tanaman melon.

3.3.1 Arsitektur Sistem

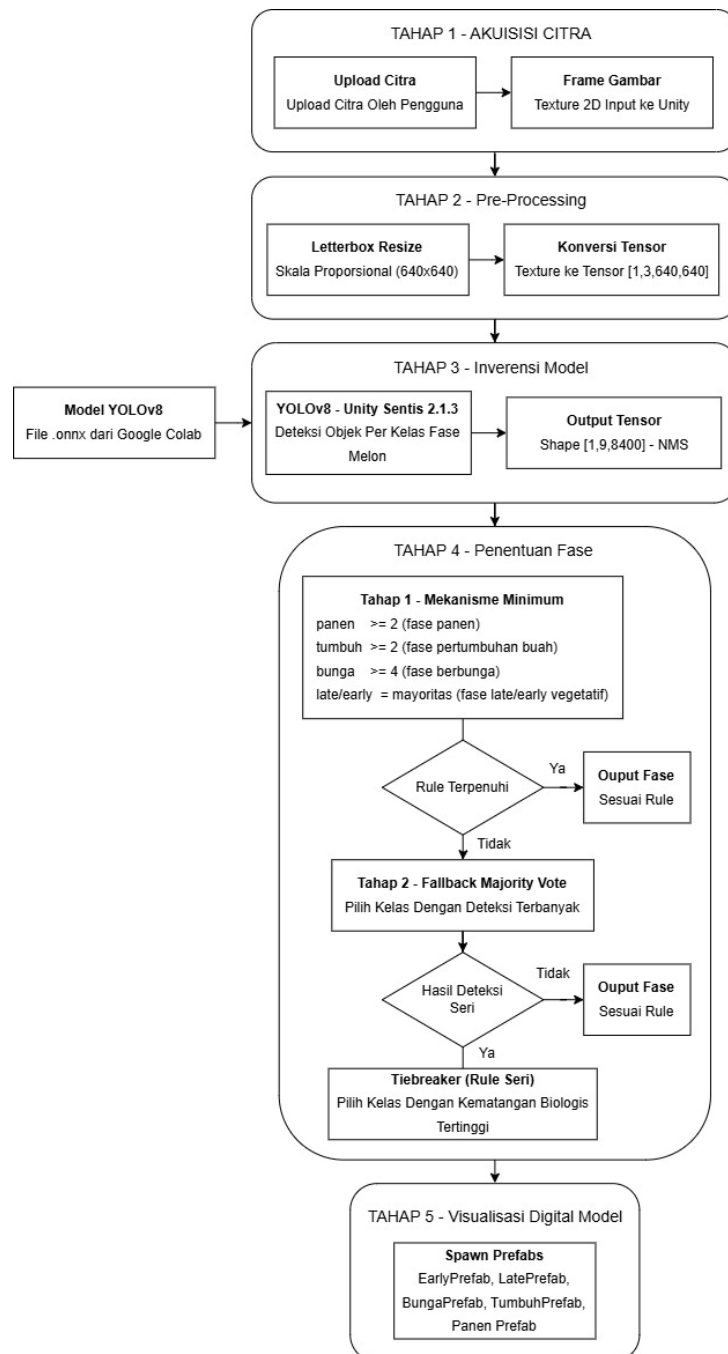
Arsitektur sistem terdiri dari dua bagian utama yang terintegrasi: deteksi dan visualisasi. Pada bagian deteksi, citra dari kamera CCTV diproses melalui tahap pre-processing (letterbox dan konversi tensor), lalu diinferensi menggunakan model YOLOv8 berformat ONNX yang diintegrasikan dengan Unity Sentis. Hasil deteksi kemudian divalidasi melalui mekanisme penentuan fase, mencakup mekanisme minimum, fallback, dan tiebreaker guna meningkatkan keandalan sistem. Pada bagian visualisasi, fase yang terdeteksi direpresentasikan secara dinamis menggunakan prefab 3D di Unity 2022.3, yang menampilkan lima fase pertumbuhan tanaman melon: early vegetatif, late vegetatif, pembungaan, pertumbuhan buah, dan panen. Gambaran arsitektur sistem disajikan pada Gambar 3.9.



Gambar 3. 9 Desain Arsitektur Sistem

3.3.2 Alur Sistem

Alur sistem menggambarkan alur kerja sistem secara keseluruhan mulai dari akuisisi gambar hingga visualisasi *digital model* berdasarkan citra yang diunggah pengguna. Diagram alur sistem disajikan pada Gambar 3.10 berikut.



Gambar 3. 10 Diagram Alur Sistem

Berdasarkan Gambar 3.10, alur sistem terdiri dari lima tahap utama yang saling terintegrasi sebagai berikut.

1. Tahap Akuisisi Gambar, citra CCTV yang diupload pengguna menghasilkan *frame* gambar yang dikirim ke *Unity* sebagai format *Texture2D*.
2. Tahap *Preprocessing*, Gambar diproses melalui Teknik *letterbox* untuk menyesuaikan dimensi citra secara proporsional dan dikonversi ke bentuk *tensor* (baris, kolom, saluran) sebelum dimasukkan ke model.
3. Tahap Inferensi Model, model *YOLOv8 (.onnx)* yang didapatkan dari proses *training*. Diolah melalui *Unity Sentis* untuk memproses *tensor* dan menghasilkan output deteksi yang kemudian difilter menggunakan *Non-Maximum Suppression (NMS)* guna menghilangkan deteksi yang tumpang tindih.
4. Tahap Penentuan Fase, di mana sistem menerapkan tiga mekanisme penentuan fase yaitu, *minimum count* dengan prioritas kematangan biologis tanaman, *fallback majority vote* apabila mekanisme pertama tidak menghasilkan keputusan, dan *tiebreaker* jika terdapat jumlah deteksi yang sama, dengan prioritas kematangan biologis tanaman.
5. Tahap Visualisasi *Digital Model*, salah satu dari lima *prefab* (*EarlyPrefab*, *LatePrefab*, *BungaPrefab*, *TumbuhPrefab*, *PanenPrefab*) ditampilkan pada *spawn point* di *Unity* sesuai fase yang dihasilkan.

3.3.3 Mekanisme Penentuan Fase

Penentuan fase pertumbuhan tanaman melon dalam sistem ini menggunakan beberapa mekanisme untuk menjelaskan logika pengambilan keputusan

berdasarkan hasil deteksi model. Keputusan akhir tidak ditentukan hanya berdasarkan satu deteksi, melainkan berdasarkan akumulasi jumlah deteksi pada masing-masing kelas dalam satu frame sehingga hasil yang diperoleh lebih stabil dan representatif. Penjabaran indeks kelas fase melon disajikan pada Tabel 3.6 berikut.

Tabel 3.6 Kelas Fase Melon dan Indeksasi Kelas

No	Kelas Fase	Indeks ke ()
1	<i>early</i>	1
2	<i>late</i>	2
3	bunga	0
4	tumbuh	4
5	panen	3

Pada tahap pertama, sistem menerapkan mekanisme *minimum count* (jumlah minimum) dengan urutan prioritas berdasarkan tingkat kematangan biologis, sehingga fase yang lebih matang diprioritaskan dalam penentuan hasil akhir. Fase *late* dan *early* tidak memiliki ambang *minimum* tersendiri karena keduanya ditentukan berdasarkan dominasi jumlah deteksi secara relatif, mengingat kedua fase ini memiliki karakteristik visual yang paling mirip sehingga ambang batas *minimum* dinilai kurang stabil. Penjabaran mekanisme *minimum count* disajikan pada Tabel 3.7 berikut.

Tabel 3. 7 Mekanisme *Minimum Count* dan Prioritas Penentuan Fase

Prioritas	Kondisi Deteksi	Keputusan Fase	keterangan
1	Panen ≥ 2	Panen	Fase paling matang diprioritaskan jika memenuhi jumlah <i>minimum</i>
2	Tumbuh ≥ 2	Pertumbuhan buah	Dipilih jika panen tidak memenuhi syarat
3	Bunga ≥ 4	Pembungaan	Membutuhkan ambang lebih tinggi agar keputusan lebih stabil
4	<i>Late</i> > <i>Early</i>	<i>Late</i> vegetatif	Dipilih jika jumlah deteksi <i>late</i> lebih dominan
5	<i>Early</i> > <i>Late</i>	<i>Early</i> vegetatif	Dipilih jika jumlah deteksi <i>early</i> lebih dominan

Apabila tidak ada kelas yang memenuhi mekanisme *minimum count*, sistem masuk ke tahap kedua yaitu *fallback majority vote* (alternatif suara mayoritas), di mana kelas dengan jumlah deteksi terbanyak dipilih sebagai hasil akhir. Jika terdapat jumlah deteksi yang sama, sistem menggunakan mekanisme *tiebreaker* (penentu) berdasarkan urutan kematangan biologis *Early* vegetative, *Late* vegetative, Pembungaan, Pertumbuhan buah, Panen. Penjabaran mekanisme *fallback* dan *tiebreaker* disajikan pada Tabel 3.8 berikut.

Tabel 3. 8 Mekanisme *Fallback* dan *Tiebreaker*

Tahap	Kondisi	Keputusan	Keterangan
<i>Fallback Majority Vote</i>	Tidak ada kelas yang memenuhi <i>minimum count</i>	Pilih kelas dengan jumlah deteksi terbanyak	Digunakan sebagai mekanisme cadangan
<i>Tiebreaker</i>	Terdapat jumlah deteksi yang sama	Pilih kelas dengan kematangan biologis tertinggi	Urutan kelas: <i>early</i> , <i>late</i> , bunga, tumbuh, panen

3.3.4 Perancangan *Prefabs Digital Model*

Setiap fase pertumbuhan tanaman melon direpresentasikan oleh satu *prefab* khusus yang dirancang agar hasil deteksi fase dapat langsung diterjemahkan ke dalam visualisasi *digital model* yang sesuai. Rangkuman korespondensi antara fase dan *prefab* disajikan pada Tabel 3.9 berikut.

Tabel 3. 9 Korespondensi Fase dan Prefabs

Fase	Nama Prefabs	Representasi Visual
<i>Early</i> vegetatif	<i>EarlyPrefab</i>	Tanaman muda, kecil, dan daun belum menjari
<i>Late</i> vegetatif	<i>LatePrefab</i>	Tanaman berkembang, dan daun menjari sempurna
Pembungaan	<i>BungaPrefab</i>	Tanaman berbunga
Pertumbuhan buah	<i>TumbuhPrefab</i>	Buah mulai terbentuk dan masih berwarna hijau
Panen	<i>PanenPrefab</i>	Buah siap panen dan berwarna kuning

3.4 Pengembangan Model

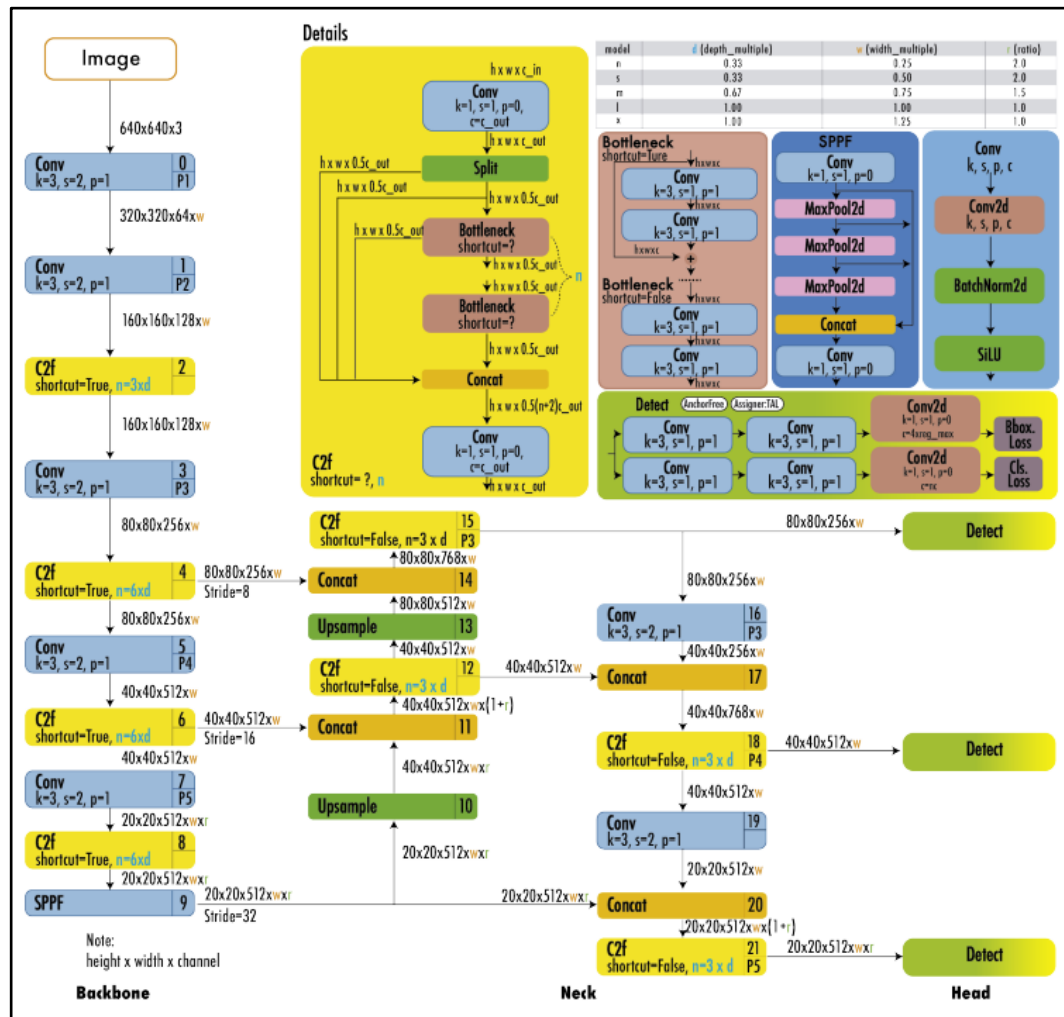
Subbab ini menjelaskan tahapan pengembangan model deteksi, mulai dari *dataset* yang telah siap digunakan untuk pelatihan hingga menghasilkan model

akhir dalam format *.onnx*. Proses pengembangan terdiri atas tiga tahap utama, yaitu konfigurasi dan pelatihan model, evaluasi hasil pelatihan, serta *ekspor* model ke format *ONNX*. Setiap tahap dilakukan secara berurutan untuk memastikan model yang dihasilkan memiliki performa yang baik dan kompatibel untuk diintegrasikan ke dalam sistem *Unity*.

3.4.1 Arsitektur YOLOv8

Arsitektur deteksi objek yang digunakan dalam penelitian ini adalah YOLOv8 (*You Only Look Once* versi 8), yang dipilih karena merupakan pengembangan terbaru dari varian YOLO. YOLOv8 dipilih karena bersifat *anchor-free*, sehingga tidak memerlukan proses penentuan *anchor box* secara manual seperti pada versi YOLO terdahulu, serta memiliki struktur *head* yang telah didekopling (*decoupled head*), yang memisahkan proses klasifikasi kelas dan regresi *bounding box* ke dalam dua cabang berbeda untuk meningkatkan akurasi deteksi (Terven & Cordova-esparza, 2024).

Menurut (Terven & Cordova-esparza, 2024), arsitektur YOLOv8 tersusun atas tiga komponen utama yang saling terintegrasi, yaitu *backbone*, *neck*, dan *head*. Ketiga komponen tersebut bekerja secara berurutan untuk mengubah citra input menjadi prediksi akhir berupa kelas objek dan posisi *bounding box*. Gambaran arsitektur YOLOv8 secara lengkap disajikan pada Gambar 3.11 berikut.



Gambar 3. 11 Arsitektur YOLOv8 oleh (Terven & Cordova-esparza, 2024)

Bagian *backbone* berfungsi mengekstraksi fitur visual dari citra input melalui serangkaian lapisan konvolusi. *Backbone* YOLOv8 menggunakan struktur turunan dari CSPDarknet53, dengan modul C2f (*Cross Stage Partial* dengan dua cabang konvolusi). Modul C2f dirancang untuk memperkaya aliran gradien antar lapisan sekaligus menjaga efisiensi parameter, sehingga proses ekstraksi fitur menjadi lebih ringan tanpa mengurangi kemampuan representasi model. Pada bagian akhir backbone terdapat modul SPPF (*Spatial Pyramid Pooling Fast*), yang menggabungkan beberapa operasi *pooling* dengan ukuran *receptive field* berbeda

untuk memperkaya informasi konteks serta meningkatkan kemampuan model dalam mengenali objek pada berbagai skala ukuran.

Bagian *neck* berfungsi menggabungkan fitur-fitur yang telah diekstraksi pada level resolusi berbeda dari *backbone*. YOLOv8 mengadopsi struktur PAN-FPN (*Path Aggregation Network - Feature Pyramid Network*), yang melakukan penggabungan fitur secara dua arah, dari resolusi rendah ke tinggi dan sebaliknya, sehingga informasi semantik dari fitur level dalam dan informasi detail spasial dari fitur level dangkal dapat saling melengkapi. Struktur ini menghasilkan tiga skala deteksi keluaran (P3, P4, dan P5) yang masing-masing bertanggung jawab mendeteksi objek berukuran kecil, sedang, dan besar.

Bagian *head* bertugas menghasilkan prediksi akhir berupa kelas objek dan koordinat *bounding box* berdasarkan fitur yang telah diproses oleh *neck*. Berbeda dengan versi YOLO sebelumnya yang menggunakan pendekatan *anchor-based*, YOLOv8 menerapkan pendekatan *anchor-free* pada *head*-nya, di mana model langsung memprediksi titik tengah dan ukuran *bounding box* tanpa bergantung pada kotak referensi (*anchor box*) yang ditentukan sebelumnya. Selain itu, *head* YOLOv8 bersifat *decoupled*, yaitu memisahkan cabang klasifikasi kelas dan cabang regresi *bounding box* ke dalam dua jalur perhitungan yang berbeda, sehingga kedua tugas tersebut dapat dioptimalkan secara independen dan menghasilkan prediksi yang lebih akurat.

Ketiga skala deteksi yang dihasilkan oleh *neck* (P3, P4, dan P5) pada akhirnya menghasilkan total 8.400 kandidat *bounding box* pada resolusi input

640x640 piksel, sebagaimana ditunjukkan pada bentuk *output tensor* [1, 9, 8400] yang dijelaskan lebih lanjut pada bagian implementasi sistem.

3.4.2 Konfigurasi dan Pelatihan Model

Versi YOLOv8 yang digunakan selama proses pelatihan adalah YOLOv8n (*nano*) dipilih sebagai model deteksi utama karena memiliki ukuran yang ringan dan kecepatan inferensi yang tinggi, sehingga sesuai untuk implementasi pada perangkat dengan sumber daya terbatas (Terven & Cordova-esparza, 2024). Model dilatih menggunakan beberapa parameter utama, yaitu *epoch*, *imgsz*, *batch*, *optimizer*, dan *patience*. Konfigurasi ini dipilih untuk mencapai keseimbangan antara akurasi model dan efisiensi komputasi, dengan nilai awal masing-masing parameter disajikan pada Tabel 3.10 berikut.

Tabel 3.10 Nilai Parameter

Parameter	Nilai Parameter
<i>Epoch</i>	100
<i>imgsz</i>	640 (sesuai dengan dataset)
<i>batch</i>	32
<i>optimizer</i>	<i>AdamW</i> (default)
<i>patience</i>	50 (default)

Nilai *epochs* dan *batch size* selanjutnya divariasikan pada tahap pengujian *tuning parameter* untuk menentukan kombinasi terbaik.

3.4.3 Evaluasi Hasil Pelatihan

Hasil pelatihan model YOLOv8n dievaluasi menggunakan beberapa metrik yang mengacu pada skema evaluasi COCO (*Common Objects in Context*). Skema ini dipilih karena menjadi standar yang umum digunakan dalam evaluasi model deteksi objek, sehingga hasil yang diperoleh dapat dibandingkan secara konsisten

dengan penelitian sejenis. Seluruh metrik evaluasi didasarkan pada nilai IoU (*Intersection over Union*), yaitu ukuran kesesuaian antara *bounding box* prediksi dan *ground truth* yang dihitung menggunakan Persamaan 3.1.

$$IoU = \frac{\text{Area tumpang tindih (Intersection)}}{\text{Area Gabungan (Union)}} = \frac{\text{Area}(B_p \cap B_{gt})}{\text{Area}(B_p \cup B_{gt})} \quad (3.1)$$

Nilai IoU ini selanjutnya menjadi dasar perhitungan *Average Precision* (AP) per kelas, yang kemudian dirata-ratakan untuk menghasilkan nilai *mAP*. Metrik evaluasi yang digunakan diuraikan sebagai berikut.

a. *Precision*

Precision mengukur proporsi prediksi positif yang benar dari seluruh prediksi positif yang dihasilkan model. *Precision* dihitung menggunakan Persamaan 3.2 sebagai berikut.

$$Precision = \frac{\text{True Positive (TP)}}{\text{True Positive} + \text{False Positive (TP + FP)}} \quad (3.2)$$

b. *Recall*

Recall mengukur kemampuan model dalam mendeteksi seluruh objek yang benar-benar ada pada data. *Recall* dihitung menggunakan Persamaan 3.3 sebagai berikut.

$$Recall = \frac{\text{True Positive (TP)}}{\text{True Positive} + \text{False Negative (TP + FN)}} \quad (3.3)$$

c. *F1-Score*

F1-Score merupakan rata-rata antara *Precision* dan *Recall* yang digunakan untuk mengukur keseimbangan antara keduanya dalam satu nilai tunggal. Metrik ini digunakan karena jumlah data per kelas fase pertumbuhan tanaman

melon tidak selalu seimbang. *F1-Score* dihitung menggunakan Persamaan 3.4 sebagai berikut.

$$F1 - Score = \frac{Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (3. 4)$$

Keterangan

TP = True Positive

FP = False Positive

FN = False Negative

d. COCO *mAP*

COCO *mean Average Precision* (COCO *mAP*) adalah metrik evaluasi standar pada deteksi objek yang diperkenalkan dalam skema evaluasi COCO. Nilai ini diperoleh dengan menghitung rata-rata AP pada beberapa *IoU threshold*, yaitu 0,50 hingga 0,95 dengan interval 0,05. Dengan demikian, COCO *mAP* tidak hanya menilai kemampuan model dalam menemukan objek, tetapi juga mengevaluasi ketepatan *bounding box* secara lebih menyeluruh. Semakin tinggi nilai COCO *mAP*, semakin baik performa deteksi model. COCO *mAP* dapat dinyatakan dengan Persamaan 3. 5 sebagai berikut.

$$mAP = \frac{1}{T} \sum_{t=1}^T \frac{1}{C} \sum_{c=1}^C AP_c(IoU_t) \quad (3. 5)$$

Keterangan:

T = jumlah *IoU threshold* (10 threshold dari 0.50-0.95 interval 0.05)

C = jumlah kelas

AP_c(IoU_t) = nilai AP untuk kelas ke-*c* pada *threshold* IoU ke-*t*

3.4.4 Ekspor ONNX Model

Setelah melalui proses pelatihan, model menghasilkan file berformat *.pt* (*PyTorch*) yang kemudian dikonversi ke format *.onnx* menggunakan perintah berikut di *Google Colab* pada Gambar 3.12 berikut.

```
model.export(format="onnx", imgsz=640, simplify=True)
```

Gambar 3. 12 Perintah Ekspor ke format .onnx

Perintah tersebut menghasilkan model ONNX dengan ukuran *input* 640x640 piksel dan struktur yang telah disederhanakan untuk meningkatkan efisiensi inferensi. Setelah konversi selesai, *file .onnx* diverifikasi menggunakan *Netron* untuk memastikan kompatibilitas struktur model dengan *Unity Sentis* sebelum diintegrasikan ke dalam sistem.

3.5 Implementasi Sistem

Implementasi sistem menjelaskan tahapan penerapan rancangan ke dalam bentuk aplikasi yang dapat dijalankan. Pada tahap ini, seluruh komponen yang telah dirancang diwujudkan menjadi sistem yang terintegrasi, mulai dari model deteksi, *aset visual 3D*, hingga *environment* aplikasi di *Unity*, yang dilakukan secara bertahap agar setiap komponen dapat dikembangkan dan dihubungkan sesuai alur kerja yang telah ditentukan.

3.5.1 Integrasi ONNX *Model* ke *Unity*

Tahap pertama implementasi adalah integrasi model ONNX ke dalam *Unity* menggunakan library *Unity Sentis*. Model YOLOv8 hasil pelatihan yang telah dikonversi ke format .onnx diimpor ke dalam proyek *Unity* agar dapat menerima *input* berupa citra, memproses inferensi deteksi objek, dan menghasilkan *output* berupa fase pertumbuhan tanaman berdasarkan citra yang diunggah pengguna. Alur kerja sistem ini dirancang sesuai dengan diagram yang telah dijelaskan pada Sub-bab 3.3.2, dengan *input* berupa citra dari kamera CCTV dalam bentuk *Texture2D*

oleh pengguna dan *output* berupa visualisasi *prefab 3D* fase pertumbuhan yang *spawn* pada *scene Unity*. *Pseudo code* alur kerja model pada sistem deteksi secara lengkap disajikan pada Gambar 3.13 berikut.

```

INPUT: citra dari kamera CCTV (Texture2D) oleh pengguna
OUTPUT: prefab 3D fase pertumbuhan ter-spawn di scene
// Tahap 1 Preprocessing Citra
Hitung skala proporsional agar citra muat dalam 640x640
Buat kanvas 640x640 berisi padding hitam (letterbox)
Salin piksel citra ke tengah kanvas sesuai skala
Konversi kanvas ke tensor [1, 3, 640, 640]
// Langkah 2 Inferensi Model YOLOv8
Kirim tensor model YOLOv8 ke GPU untuk diinferensi
Terima output tensor [1, 9, 8400] dari model
Salin output dari GPU ke CPU untuk diproses
// Langkah 3 Parsing Kandidat Deteksi dan Filtering NMS
UNTUK setiap kandidat dari 8.400 bounding box:
    Ambil koordinat box dan skor per kelas
    JIKA skor tertinggi < confThreshold (0.25) : LEWATI
    Simpan sebagai kandidat deteksi valid
// Langkah 4 Filtering Metode Non-Maximum Suppression (NMS)
UNTUK setiap kelas {bunga, early, late, panen, tumbuh}:
    Urutkan kandidat kelas tersebut dari skor tertinggi
    SELAMA masih ada kandidat
        Ambil kandidat teratas sebagai deteksi final
        Hapus kandidat lain yang nilai IoU-nya > 0.45
// Langkah 4 Penentuan Fase
Hitung jumlah deteksi untuk setiap kelas fase
    JIKA jumlah panen  $\geq 2$ : spawn prefab PanenPrefab;
    JIKA jumlah tumbuh  $\geq 2$ : spawn prefab TumbuhPrefab;
    JIKA jumlah bunga  $\geq 4$ : spawn prefab BungaPrefab;
    JIKA late > early: spawn prefab LatePrefab;
    JIKA early > late: spawn prefab EarlyPrefab;
    JIKA tidak ada yang memenuhi: spawn prefab kelas dengan deteksi
        terbanyak;
    JIKA tidak ada deteksi sama sekali: tidak spawn prefab (Unknown);
// Langkah 5 Spawn Prefab
Hapus objek 3D sebelumnya dari scene
Pilih prefab sesuai fase yang terdeteksi
Spawn prefab di setiap titik yang telah ditentukan
SELESAI

```

Gambar 3. 13 Pseudo Code

Berdasarkan *pseudo code* pada Gambar 3.13, alur kerja model pada sistem terdiri atas lima langkah utama. Langkah pertama adalah *preprocessing* citra, di mana citra *input* kamera CCTV yang berformat *Texture2D* yang diunggah

pengguna diubah ukurannya secara proporsional agar dapat dimuat dalam dimensi 640x640 piksel. Proses ini menggunakan teknik *letterboxing*, yaitu menempatkan citra yang telah diskalakan di tengah kanvas berukuran 640x640 dengan pengisi (*padding*) berwarna hitam pada sisi-sisi yang tersisa. Citra hasil *letterboxing* kemudian dikonversi menjadi *tensor* berdimensi [1, 3, 640, 640] sebagai format *input* yang diterima model YOLOv8.

Langkah kedua adalah inferensi model, di mana *tensor input* dikirimkan ke *GPU* untuk diproses oleh model YOLOv8 menggunakan *Unity Sentis*. Model menghasilkan *output* berupa *tensor* berdimensi [1, 9, 8400], yang merepresentasikan 8.400 kandidat *bounding box* dengan masing-masing memiliki 9 nilai, yaitu empat koordinat box dan lima skor kelas. *Output tensor* kemudian disalin kembali ke *CPU* untuk diproses lebih lanjut.

Langkah ketiga adalah *parsing* dan penyaringan awal kandidat deteksi. Setiap kandidat dari 8.400 *bounding box* dievaluasi dengan mengambil koordinat kotak dan skor tertinggi dari seluruh kelas. Kandidat yang memiliki skor tertinggi di bawah nilai *confidence threshold* sebesar 0,25 diabaikan, sementara kandidat yang memenuhi ambang batas disimpan sebagai deteksi valid untuk diproses pada tahap berikutnya.

Langkah keempat adalah proses *filtering* menggunakan metode NMS. Setiap kandidat kelas (*early*, *late*, bunga, tumbuh, dan panen) diurutkan berdasarkan skor dari tertinggi ke terendah, kemudian kandidat dengan skor tertinggi dipilih sebagai deteksi final dan kandidat lain dengan nilai IoU melebihi 0,45 terhadap deteksi tersebut dihapus untuk mencegah deteksi ganda. Setelah

filtering, fase pertumbuhan dominan ditentukan berdasarkan mekanisme *minimum* dimana fase panen dan tumbuh dipilih jika terdapat minimal dua deteksi, fase bunga jika minimal empat deteksi, serta fase *late* atau *early* dipilih berdasarkan jumlah deteksi terbanyak di antara keduanya. Apabila tidak ada kelas yang memenuhi kondisi tersebut, kelas dengan deteksi terbanyak dipilih sebagai *fallback*. Jika tidak terdapat deteksi sama sekali, sistem tidak melakukan *spawn prefab* dan kondisi dinyatakan sebagai *Unknown*.

Langkah kelima adalah *spawn prefab* 3D. Objek 3D yang sebelumnya telah di-*spawn* pada *scene* dihapus terlebih dahulu untuk menghindari penumpukan objek. Selanjutnya, *prefab* yang sesuai dengan fase pertumbuhan yang terdeteksi dipilih dan di-*spawn* pada setiap titik koordinat yang telah ditentukan di dalam *scene Unity*, sehingga visualisasi *digital model* fase pertumbuhan tanaman dapat ditampilkan kepada pengguna.

3.5.2 Pengembangan *Prefab 3D Digital Model*

Tahap kedua adalah pengembangan *prefab 3D digital model*. Setiap fase pertumbuhan tanaman melon dimodelkan dalam bentuk objek 3D menggunakan *Blender3D*, mencakup fase *early* vegetatif, *late* vegetatif, pembungaan, pertumbuhan buah, dan panen. Aset yang telah selesai dimodelkan diekspor ke format *FBX*, kemudian diimpor ke Unity sebagai *prefab* yang ditampilkan sesuai hasil mekanisme penentuan fase sehingga visualisasi digital model dapat mencerminkan kondisi nyata tanaman secara akurat. *Prefab* kelima fase pertumbuhan ditunjukkan pada Gambar 3.14, berurutan dari paling kiri *PanenPrefab* hingga paling kanan *EarlyPrefab*.



Gambar 3.14 *Prefabs Fase Melon*

3.5.3 Konfigurasi *Scene Unity*

Tahap ketiga adalah konfigurasi *scene* di *Unity* yang mencakup pengaturan seluruh komponen sistem agar model deteksi, *aset visual 3D*, dan antarmuka pengguna dapat berjalan dalam satu alur yang utuh. Pengembangan *scene unity* dikhususkan untuk mobile platform. Gambaran *scene* utama ditunjukkan pada Gambar 3.15 dan Gambar 3.16 berikut.



Gambar 3.15 *Desain Scene Environment Greenhouse*

Gambar 3.15 menunjukkan tampilan *environment* visualisasi *greenhouse* yang digunakan sebagai latar lingkungan dalam sistem visualisasi *digital model* fase pertumbuhan tanaman melon berbasis deteksi YOLOv8, merepresentasikan kondisi lingkungan *greenhouse* sebagai konteks utama dalam sistem visual.



Gambar 3.16 Scene Tanaman Melon

Gambar 3.16 menunjukkan tampilan *scene* utama di dalam *greenhouse* yang merupakan area visualisasi tanaman melon pada *platform mobile*. Pada *scene* ini pengguna dapat memantau fase pertumbuhan tanaman melon melalui tampilan *prefab 3D* yang berubah secara dinamis sesuai hasil deteksi YOLOv8 berdasarkan citra yang diunggah pengguna.

3.6 Skenario Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh komponen sistem berfungsi sesuai dengan tujuan penelitian. Pengujian mencakup tiga skenario utama, yaitu pengujian *tuning parameter* YOLOv8 untuk menentukan konfigurasi model terbaik, pengujian deteksi *model* YOLOv8 untuk mengukur ketepatan model

dalam mengenali fase pertumbuhan tanaman melon, serta pengujian komponen visualisasi *digital model* untuk memastikan logika penentuan fase dan visualisasi *prefab* berjalan sesuai mekanisme yang telah dirancang.

3.6.1 Skenario Pengujian *Tuning Parameter YOLOv8*

Pengujian *tuning parameter* dilakukan untuk menganalisis pengaruh variasi konfigurasi *hyperparameter* terhadap performa model *YOLOv8* dalam mendeteksi fase pertumbuhan tanaman melon. Tujuan pengujian ini adalah menentukan kombinasi parameter yang menghasilkan performa deteksi terbaik berdasarkan nilai metrik evaluasi yang terukur secara kuantitatif.

Pengujian *tuning parameter* pada penelitian ini dilakukan dalam dua tahap. Tahap pertama bertujuan untuk menentukan kombinasi *epochs* dan *batch size* terbaik dengan *optimizer* tetap (parameter *default YOLOv8*) menggunakan *AdamW optimizer*. Tahap kedua dilakukan setelah kombinasi *epochs* dan *batch size* terbaik diperoleh dari tahap pertama, kemudian nilai tersebut digunakan sebagai konfigurasi tetap untuk menguji variasi *optimizer* guna menentukan algoritma optimasi yang paling optimal untuk *dataset* fase pertumbuhan tanaman melon yang digunakan. Pendekatan dua tahap ini dipilih karena lebih efisien dibandingkan menguji seluruh kombinasi parameter sekaligus, serta menghasilkan alur analisis yang lebih sistematis dan mudah diinterpretasikan.

Setiap konfigurasi parameter dilatih menggunakan *dataset* yang sama, yaitu *dataset primer* citra tanaman melon yang telah melalui proses anotasi dan augmentasi sebagaimana dijelaskan pada subbab sebelumnya. Performa setiap konfigurasi dievaluasi menggunakan metrik *Precision*, *Recall*, *F1-Score*,

$mAP@0.5$, dan $mAP@0.5:0.95$ yang masing-masing telah dijelaskan pada Sub-Subbab 3.4.3.

Skenario pengujian tahap pertama, yaitu variasi *epochs* dan *batch size*, disajikan pada Tabel 3.11 berikut.

Tabel 3.11 Skenario Pengujian Tahap 1 - Variasi *Epochs* dan *Batch Size*

No	<i>Epochs</i>	<i>Batch Size</i>	<i>Optimizer</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	$mAP@0.5$	$mAP@0.5:0.95$
1	50	16	<i>AdamW</i>					
2	50	32	<i>AdamW</i>					
3	100	16	<i>AdamW</i>					
4	100	32	<i>AdamW</i>					
5	150	16	<i>AdamW</i>					
6	150	32	<i>AdamW</i>					

Skenario pengujian tahap kedua, yaitu variasi optimizer menggunakan kombinasi *epochs* dan *batch size* terbaik dari tahap pertama, disajikan pada Tabel 3.12 berikut.

Tabel 3.12 Skenario Pengujian Tahap 2 – Variasi *Optimizer*

No	<i>Epochs</i>	<i>Batch Size</i>	<i>Optimizer</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	$mAP@0.5$	$mAP@0.5:0.95$
1	<i>terbaik</i>	<i>terbaik</i>	SGD					
2	<i>terbaik</i>	<i>terbaik</i>	Adam					
3	<i>terbaik</i>	<i>terbaik</i>	AdamW					

3.6.2 Skenario Pengujian Deteksi *Model YOLOv8*

Pengujian deteksi *model YOLOv8* dilakukan untuk mengukur kemampuan model terbaik hasil pengujian *tuning parameter* dalam mengenali fase pertumbuhan tanaman melon secara tepat melalui alur sistem yang telah diimplementasikan di *Unity*. Pengujian dilakukan dengan mengunggah citra kamera CCTV ke dalam sistem, kemudian sistem memproses citra tersebut melalui seluruh alur yang telah diimplementasikan, mulai dari *preprocessing letterbox*, inferensi *model ONNX* via

Unity Sentis, hingga penentuan fase melalui mekanisme penentuan fase sebagaimana dijelaskan pada Sub-bab 3.5.1.

Citra uji yang digunakan merupakan *data testing (test)* dari hasil pembagian *dataset* yang belum pernah digunakan selama proses pelatihan maupun validasi model. Citra uji terdiri dari 2 citra setiap fase pertumbuhan tanaman melon, yaitu *early vegetatif*, *late vegetatif*, pembungaan, pertumbuhan buah, dan panen, dengan total 10 citra uji. Selanjutnya, fase yang terdeteksi oleh sistem dibandingkan dengan fase asli pada masing-masing citra uji, dikategorikan "Benar" apabila keduanya sesuai dan "Salah" apabila tidak sesuai. Skenario pengujian deteksi *model YOLOv8* disajikan pada Tabel 3.13 berikut.

Tabel 3.13 Skenario Pengujian Deteksi *Model YOLOv8*

No	Gambar Uji	Fase Asli	Fase Terdeteksi	Status
1		<i>Early vegetatif</i>		
2		<i>Early vegetatif</i>		
3		<i>Late vegetatif</i>		
4		<i>Late vegetatif</i>		

No	Gambar Uji	Fase Asli	Fase Terdeteksi	Status
5		Pembungaan		
6		Pembungaan		
7		Pertumbuhan Buah		
8		Pertumbuhan Buah		
9		Panen		
10		Panen		

3.6.3 Skenario Pengujian Komponen Visualisasi *Digital Model* (Penentuan Fase Untuk Visualisasi Fase Pertumbuhan Tanaman Melon)

Sebagai komponen dalam sistem visualisasi berbasis *digital model*, penentuan fase berperan dalam menentukan visualisasi perubahan pertumbuhan tanaman melon berdasarkan hasil deteksi *YOLOv8* terhadap citra yang diunggah

pengguna. Pengujian komponen ini dilakukan dengan mensimulasikan berbagai kombinasi jumlah deteksi *label* pada setiap kelas, kemudian mencatat fase yang dihasilkan sistem dan membandingkannya dengan fase yang diharapkan berdasarkan mekanisme penentuan fase yang telah dirancang.

Pengujian ini mencakup tiga mekanisme utama, yaitu *minimum count* untuk memastikan sistem memprioritaskan fase yang lebih matang sesuai urutan prioritas biologis, *fallback majority vote* untuk memastikan sistem memilih kelas dengan jumlah deteksi terbanyak ketika tidak ada kelas yang memenuhi mekanisme *minimum*, serta skenario seri (*tiebreaker*) untuk memastikan sistem memilih fase dengan tingkat kematangan biologis tertinggi ketika dua atau lebih kelas memiliki jumlah deteksi yang sama. Setiap skenario dipilih secara representatif untuk membuktikan bahwa masing-masing mekanisme berjalan dengan benar tanpa harus menguji seluruh kemungkinan kombinasi nilai. Skenario pengujian penentuan fase disajikan pada Tabel 3.14 berikut.

Tabel 3.14 Skenario Pengujian Komponen Visualisasi *Digital Model* (Penentuan Fase untuk Visualisasi Fase Pertumbuhan Tanaman Melon)

No	Skenario	<i>early</i>	<i>late</i>	bunga	tumbuh	panen	Fase/Prefab Diharapkan	Fase/Prefab Dihasilkan	Status
Minimum Count (jumlah minimal)									
1	Satu fase memenuhi <i>minimum count</i>	0	0	0	0	2	Panen (<i>PanenPrefab</i>)		
2	Dua fase memenuhi <i>minimum count</i> , prioritas kematangan biologis	0	0	0	2	2	Panen (<i>PanenPrefab</i>)		
3	Tiga fase memenuhi <i>minimum count</i> , prioritas kematangan biologis	0	0	4	2	2	Panen (<i>PanenPrefab</i>)		
Fallback Majority Vote (Berdasarkan jumlah deteksi ketika tidak memenuhi mekanisme <i>minimum count</i>)									
4	Tidak ada yang memenuhi <i>minimum count</i> , <i>spawn</i> dengan	0	0	2	1	0	Pembungaan (<i>BungaPrefab</i>)		

No	Skenario	early	late	bunga	tumbuh	panen	Fase/Prefab Diharapkan	Fase/Prefab Dihasilkan	Status
	jumlah deteksi terbanyak								
Skenario Seri (Berdasarkan urutan kematangan biologis)									
5	Seri dua fase, prioritas kematangan biologis	0	0	1	1	0	Pertumbuhan Buah (<i>TumbuhPrefab</i>)		
6	Tidak ada deteksi sama sekali	0	0	0	0	0	Unknown (Tidak muncul <i>prefab</i>)		

3.7 Evaluasi User

Pengujian ini bertujuan untuk menilai kualitas aplikasi visualisasi perubahan fase pertumbuhan tanaman melon yang dikembangkan menggunakan *Unity* pada *platform mobile*, dengan memanfaatkan hasil deteksi model *YOLOv8*. Aspek yang dievaluasi meliputi *engagement*, *functionality*, *aesthetics*, *information quality*, dan *subjective quality* yang dirasakan pengguna selama penggunaan aplikasi.

Metode evaluasi yang digunakan adalah *User Version of the Mobile Application Rating Scale (uMARS)*, yang dikembangkan oleh (Stoyanov et al., 2016). *uMARS* merupakan adaptasi dari *Mobile Application Rating Scale (MARS)* yang sebelumnya memerlukan keahlian profesional di bidang *mHealth*, dan telah disederhanakan agar dapat digunakan langsung oleh pengguna akhir tanpa latar belakang teknis khusus. Instrumen ini dipilih karena dirancang secara khusus untuk mengukur kualitas aplikasi *mobile* dari sudut pandang pengguna akhir, sehingga lebih relevan dan representatif dibandingkan instrumen evaluasi *usability* generik yang tidak mempertimbangkan karakteristik unik *platform mobile*, seperti keterbatasan layar, interaksi sentuh, dan mobilitas penggunaan (Stoyanov et al., 2016). Dengan demikian, penggunaan *uMARS* dalam penelitian ini diharapkan

mampu menghasilkan data evaluasi yang lebih akurat dan mencerminkan pengalaman nyata pengguna dalam berinteraksi dengan aplikasi *mobile* yang dikembangkan.

3.7.1 Instrumen *uMARS*

Instrumen *uMARS* terdiri atas 20 pernyataan yang dikelompokkan ke dalam lima subskala, yaitu *Engagement* (5 item), *Functionality* (4 item), *Aesthetics* (3 item), *Information Quality* (4 item), dan *Subjective Quality* (4 item). Setiap pernyataan dinilai menggunakan skala *Likert* 5 poin, yaitu Sangat Tidak Setuju = 1, Tidak Setuju = 2, Netral = 3, Setuju = 4, dan Sangat Setuju = 5.

Subskala *Engagement* mengukur seberapa menarik dan interaktif aplikasi bagi pengguna. Subskala *Functionality* mengukur kemudahan penggunaan, navigasi, dan kinerja teknis aplikasi di perangkat *mobile*. Subskala *Aesthetics* menilai kualitas tampilan visual dan desain antarmuka. Subskala *Information Quality* mengevaluasi akurasi, kejelasan, dan relevansi informasi yang disajikan. Subskala *Subjective Quality* mengukur kepuasan dan penilaian menyeluruh pengguna terhadap aplikasi.

Daftar pernyataan *uMARS* yang digunakan dalam penelitian ini diadaptasi dari versi bahasa Inggris (Stoyanov et al., 2016) dan diterjemahkan ke dalam Bahasa Indonesia untuk keperluan penelitian ini. Daftar pernyataan tersebut disajikan pada Tabel 3.15 berikut.

Tabel 3.15 Daftar Pertanyaan Kuisioner *uMARS*

Nomor	Subskala	Pertanyaan
1	<i>Engagement</i>	Aplikasi ini menyenangkan dan menghibur untuk digunakan
2	<i>Engagement</i>	Aplikasi ini menarik dan membuat saya ingin menggunakannya.
3	<i>Engagement</i>	Aplikasi ini dapat disesuaikan dengan kebutuhan saya.

Nomor	Subskala	Pertanyaan
4	<i>Engagement</i>	Aplikasi ini memungkinkan saya melakukan hal-hal sesuai dengan yang saya inginkan.
5	<i>Engagement</i>	Saya kemungkinan akan berbagi informasi dari aplikasi ini kepada orang lain.
6	<i>Functionality</i>	Aplikasi ini berjalan dengan baik dan tanpa masalah teknis.
7	<i>Functionality</i>	Aplikasi ini mudah dipelajari cara penggunaannya.
8	<i>Functionality</i>	Saya dapat menjadi terampil menggunakan aplikasi ini dengan cepat.
9	<i>Functionality</i>	Aplikasi ini memiliki semua fitur yang saya harapkan ada.
10	<i>Aesthetics</i>	Tampilan antarmuka aplikasi ini menarik secara visual.
11	<i>Aesthetics</i>	Desain layar aplikasi ini berkualitas tinggi.
12	<i>Aesthetics</i>	Aplikasi ini memiliki desain visual yang konsisten.
13	<i>Information Quality</i>	Informasi dalam aplikasi ini ditulis dengan baik dan relevan dengan tujuan aplikasi.
14	<i>Information Quality</i>	Informasi yang terdapat dalam aplikasi ini akurat dan benar.
15	<i>Information Quality</i>	Aplikasi ini berasal dari sumber yang dapat dipercaya.
16	<i>Information Quality</i>	Informasi dalam aplikasi ini ditampilkan dengan jelas.
17	<i>Subjective Quality</i>	Anda berpikir akan sering menggunakan aplikasi ini dalam 12 bulan ke depan.
18	<i>Subjective Quality</i>	Saya akan merekomendasikan aplikasi ini kepada orang yang membutuhkannya.
19	<i>Subjective Quality</i>	Saya bersedia membayar untuk menggunakan aplikasi ini.
20	<i>Subjective Quality</i>	Penilaian bintang keseluruhan terhadap aplikasi ini (1–5).

3.7.2 Perhitungan Skor *uMARS*

Perhitungan skor *uMARS* dilakukan dengan menghitung nilai rata-rata dari setiap subskala secara terpisah. Skor rata-rata masing-masing subskala diperoleh menggunakan Persamaan 3. 6 berikut.

$$S_k = \frac{\sum x_i}{n_k} \quad (3. 6)$$

Keterangan:

S_k = Skor rata-rata subskala ke-k

$\sum x_i$ = Jumlah skor jawaban responden pada seluruh item dalam subskala ke-k

n_k = Jumlah item dalam subskala ke-k

Selanjutnya, skor total *uMARS* diperoleh dengan menghitung rata-rata dari kelima subskala sebagaimana ditunjukkan pada Persamaan 3. 7 berikut.

$$uMARS = \frac{(S_{Eng} + S_{Fun} + S_{Aes} + S_{Inf} + S_{Sub})}{5} \quad (3. 7)$$

Keterangan:

$uMARS$ = Skor total $uMARS$ pada rentang 1-5
 S_{Eng} = Skor rata-rata subskala *Engagement*
 S_{Fun} = Skor rata-rata subskala *Functionality*
 S_{Aes} = Skor rata-rata subskala *Aesethic*
 S_{Inf} = Skor rata-rata subskala *Information Quality*
 S_{Sub} = Skor rata-rata subskala *Subjective Quality*

Interpretasi skor $uMARS$ mengacu pada kategori yang dikemukakan oleh Stoyanov et al. (2016), di mana skor $\geq 3,00$ dinyatakan *acceptable* (dapat diterima), skor $\geq 4,00$ dikategorikan *good* (baik), dan skor $\geq 4,50$ dikategorikan *excellent* (sangat baik). Panduan interpretasi lengkap disajikan pada Tabel 3.16 berikut.

Tabel 3.16 Kategori Interpretasi Skor $uMARS$

Rentang Skor	Kategori	Pertanyaan
1,00 – 1,99	Sangat Buruk	Aplikasi tidak memadai dan perlu perombakan menyeluruh
2,00 – 2,99	Buruk	Aplikasi memiliki banyak kekurangan yang harus diperbaiki
3,00 – 3,99	Cukup / Dapat Diterima	Aplikasi memenuhi standar minimal yang dapat diterima
4,00 – 4,49	Baik	Aplikasi berkualitas baik dan layak digunakan
4,50 – 5,00	Sangat Baik	Aplilasi berkualitas sangat tinggi dan sangat direkomendasikan

3.7.3 Cakupan dan Mekanisme Uji

Pengujian dilakukan terhadap 35 responden yang mewakili calon pengguna sistem dengan kriteria responden yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Mahasiswa atau dosen di bidang teknologi informasi, pertanian, atau agroteknologi, atau masyarakat umum yang memiliki ketertarikan terhadap penerapan teknologi dalam pertanian.
2. Bersedia mengikuti seluruh proses evaluasi, mulai dari uji aplikasi di perangkat *mobile* hingga pengisian kuesioner serta memberikan umpan balik terhadap sistem.

3. Memiliki perangkat smartphone berbasis *Android* yang mendukung instalasi dan pengoperasian aplikasi *mobile* yang dikembangkan.

Pengujian dilakukan dalam dua tahap utama. Tahap pertama adalah demonstrasi dan eksplorasi sistem, yaitu responden diberi kesempatan untuk mencoba aplikasi secara langsung di perangkat *mobile* masing-masing, mulai dari mengunggah citra kamera CCTV tanaman melon melalui antarmuka aplikasi, melihat hasil deteksi fase pertumbuhan dari model *YOLOv8*, hingga melihat visualisasi fase pertumbuhan hasil deteksi pada antarmuka *Unity*. Responden diberikan kebebasan untuk mengeksplorasi seluruh fitur aplikasi dengan panduan singkat mengenai cara mengunggah citra, guna memperoleh penilaian yang autentik terhadap kemudahan dan kenyamanan penggunaan aplikasi.

Tahap kedua adalah pengisian kuesioner *uMARS*. Setelah menyelesaikan eksplorasi aplikasi, responden diminta menjawab seluruh 20 pernyataan menggunakan skala *Likert* 1 sampai 5. Hasil jawaban kemudian dihitung dan diinterpretasikan sesuai prosedur yang telah dijelaskan pada Sub Bab 3.7.2.

BAB IV

DESAIN DAN IMPLEMENTASI

4.1 Tata Cara Pengujian

Pengujian dalam penelitian ini dalam empat tahap, yaitu pengujian model *YOLOv8*, pengujian deteksi model, pengujian komponen visualisasi, dan evaluasi pengguna menggunakan *uMARS*. Pertama dilakukan pengujian pada model *YOLOv8* melalui dua tahap *tuning parameter*. Tahap pertama menguji variasi *epochs* dan *batch size*, sementara tahap kedua menguji variasi *optimizer* dengan kombinasi parameter terbaik dari tahap pertama. Metrik evaluasi yang digunakan termasuk *Precision*, *Recall*, *F1-Score*, *mAP@0.5*, dan *mAP@0.5:0.95* untuk menentukan konfigurasi terbaik yang kemudian diterapkan pada tahap deteksi.

Pada tahap pengujian deteksi, citra uji dimasukkan ke dalam sistem untuk mengukur kemampuan model dalam mendeteksi fase pertumbuhan tanaman melon. Hasil deteksi dibandingkan dengan fase asli pada citra uji untuk menilai akurasi deteksi. Selanjutnya, pengujian komponen visualisasi *digital model* dilakukan untuk memverifikasi keakuratan mekanisme penentuan fase dan *spawn prefab*. Pengujian ini juga mensimulasikan berbagai kombinasi jumlah deteksi untuk memastikan mekanisme penentuan fase seperti *minimum count*, *fallback majority vote*, dan *tiebreaker* berfungsi sesuai desain sistem.

Tahap terakhir adalah evaluasi pengalaman pengguna menggunakan metode *User Version of the Mobile Application Rating Scale (uMARS)*. Responden yang menguji aplikasi *mobile* memberikan penilaian melalui kuesioner untuk menilai

kemudahan penggunaan, efisiensi, dan kepuasan terhadap sistem yang dikembangkan. Dengan demikian, pengujian dilakukan secara terstruktur, mencakup evaluasi komponen sistem dan integrasi keseluruhan serta pengalaman pengguna.

4.2 Hasil Pengujian *Tuning Parameter YOLOv8*

Subbab ini menyajikan hasil pengujian *tuning parameter* model *YOLOv8* yang dilakukan dalam dua tahap. Tahap pertama membahas hasil variasi *epochs* dan *batch size* untuk menentukan kombinasi terbaik, sedangkan tahap kedua membahas hasil variasi *optimizer* menggunakan kombinasi *epochs* dan *batch size* terbaik yang diperoleh dari tahap pertama. Seluruh hasil pengujian dievaluasi menggunakan metrik *Precision*, *Recall*, *F1-Score*, *mAP@0.5*, dan *mAP@0.5:0.95*.

4.2.1 Hasil Pengujian *Tuning Parameter Tahap 1 (Variasi Epochs dan Batch Size)*

Pengujian tahap pertama dilakukan dengan memvariasikan nilai *epochs* dan *batch size* pada enam konfigurasi, sementara *optimizer* ditetapkan menggunakan *AdamW* sebagai nilai default. Tujuan pengujian ini adalah menentukan kombinasi *epochs* dan *batch size* yang menghasilkan performa deteksi terbaik terhadap *dataset* fase pertumbuhan tanaman melon. Hasil pengujian tahap pertama disajikan pada Tabel 4.1 berikut.

Tabel 4. 1 Hasil Skenario Pengujian Tahap 1 - Variasi *Epochs* dan *Batch Size*

No	<i>Epochs</i>	<i>Batch Size</i>	<i>Optimizer</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>mAP@0.5</i>	<i>mAP@0.5:0.95</i>
1	50	16	<i>AdamW</i>	0.7592	0.8280	0.7921	0.8547	0.5307
2	50	32	<i>AdamW</i>	0.7955	0.7823	0.7889	0.8583	0.5057
3	100	16	<i>AdamW</i>	0.7589	0.8112	0.7842	0.8523	0.5229
4	100	32	<i>AdamW</i>	0.7561	0.8142	0.7840	0.8545	0.5435
5	150	16	<i>AdamW</i>	0.7960	0.8136	0.8047	0.8771	0.5468

No	Epochs	Batch Size	Optimizer	Precision	Recall	F1-Score	mAP@0.5	mAP@0.5:0.95
6	150	32	AdamW	0.8156	0.7948	0.8051	0.8699	0.5596

Kesimpulan hasil pengujian pada Tabel 4.1 menunjukkan bahwa Skenario 5 dengan konfigurasi *Epoch* 150 dan *Batch Size* 16 merupakan kombinasi parameter terbaik, dengan nilai *Precision* = 0.7960, *Recall* = 0.8136, *F1-Score* = 0.8047, *mAP@0.5* = 0.8771, dan *mAP@0.5:0.95* = 0.5468. Konfigurasi ini mencatatkan *mAP@0.5* tertinggi di antara seluruh skenario dengan selisih 0.0248 terhadap konfigurasi terendah (Skenario 3, *mAP@0.5* = 0.8523), serta *Recall* yang cukup baik untuk meminimalkan objek yang terlewat, Meskipun *F1-Score* Skenario 5 hampir identik dengan Skenario 6 (selisih 0.0004), pemilihan *Batch Size* 16 juga sesuai dan tetap dipertahankan karena frekuensi *gradient update* yang lebih tinggi akan memperjelas perbedaan perilaku antar *optimizer* pada pengujian tahap kedua. Oleh karena itu, kombinasi ini ditetapkan sebagai konfigurasi tetap untuk tahap selanjutnya.

4.2.2 Hasil Pengujian *Tuning Parameter* Tahap 2 (Variasi *Optimizer*)

Pengujian tahap kedua dilakukan dengan memvariasikan jenis *optimizer* pada tiga konfigurasi, sementara nilai *epochs* dan *batch size* ditetapkan berdasarkan hasil terbaik tahap pertama yaitu *epochs* = 150 dan *batch size* = 16. Tujuan pengujian ini adalah menentukan algoritma optimasi yang paling optimal untuk melatih model *YOLOv8* pada *dataset* fase pertumbuhan tanaman melon yang digunakan. Hasil pengujian tahap kedua disajikan pada Tabel 4.2 berikut.

Tabel 4. 2 Hasil Skenario Pengujian Tahap 2 - Variasi *Optimizer*

No	Epochs	Batch Size	Optimizer	Precision	Recall	F1-Score	mAP@0.5	mAP@0.5:0.95
1	150	16	AdamW	0.7960	0.8136	0.8047	0.8771	0.5468

No	Epochs	Batch Size	Optimizer	Precision	Recall	F1-Score	mAP@0.5	mAP@0.5:0.95
2	150	16	Adam	0.8096	0.8194	0.8145	0.8588	0.5510
3	150	16	SGD	0.8344	0.7888	0.8109	0.8696	0.5613

Berdasarkan hasil pengujian pada Tabel 4.2, *optimizer* terbaik diperoleh pada skenario percobaan nomor 2 yaitu *Adam*, yang menghasilkan nilai *Precision* = 0.8096, *Recall* = 0.8194, *F1-Score* = 0.8145, *mAP@0.5* = 0.8588, dan *mAP@0.5:0.95* = 0.5510. Hasil ini menunjukkan bahwa *optimizer Adam* lebih optimal dibandingkan *optimizer* lainnya dalam melatih model *YOLOv8* pada *dataset* yang digunakan, ditunjukkan dengan *F1-Score* tertinggi sebesar 0.8145 dan *Recall* terbaik sebesar 0.8194 yang mencerminkan keseimbangan *Precision* dan *Recall* paling baik serta kemampuan model dalam meminimalkan objek yang terlewat, sehingga ditetapkan sebagai bagian dari konfigurasi parameter akhir.

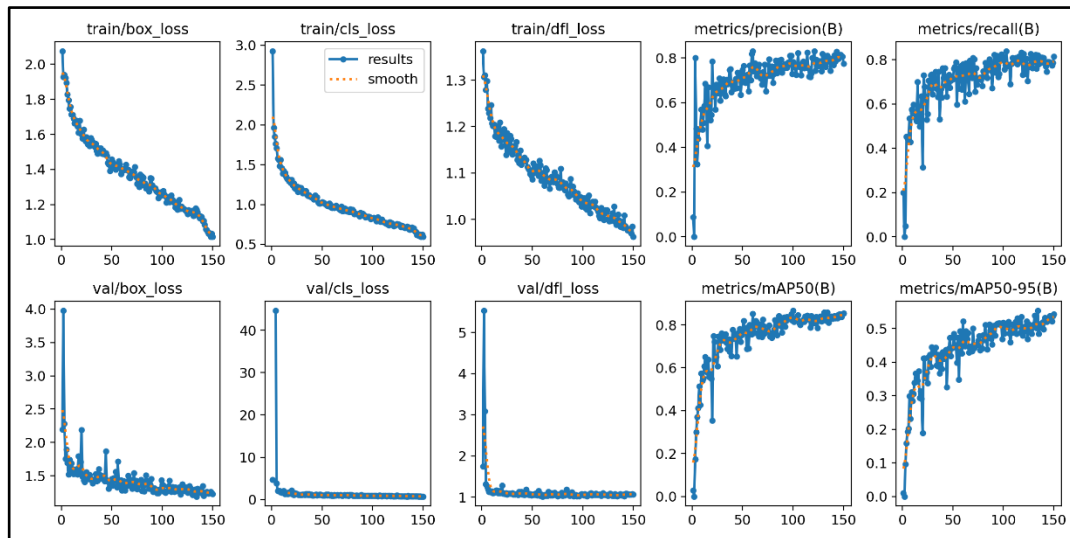
4.2.3 Konfigurasi Parameter Terbaik

Berdasarkan hasil pengujian dua tahap yang telah dilakukan, konfigurasi parameter terbaik model *YOLOv8* pada penelitian ini diperoleh dari kombinasi hasil terbaik tahap pertama dan tahap kedua. Konfigurasi tersebut disajikan pada Tabel 4.3 berikut.

Tabel 4. 3 Konfigurasi Parameter Terbaik Hasil Pengujain

Parameter	Nilai (yang diuji)	Nilai Terbaik
Epochs	50, 100, 150	150
Batch Size	16, 32	16
Optimizer	AdamW, Adam, SGD	Adam
Precision	-	0.8096
Recall	-	0.8194
F1-Score	-	0.8145
mAP@0.5	-	0.8588
mAP@0.5:0.95	-	0.5510

Untuk memperjelas performa model *YOLOv8* pada konfigurasi parameter terbaik, berikut ditampilkan grafik hasil pelatihan yang mencakup kurva metrik evaluasi yang mencakup perbandingan *training loss* dan *validation loss* dan grafik *confusion matrix* pada Gambar 4.1 berikut:

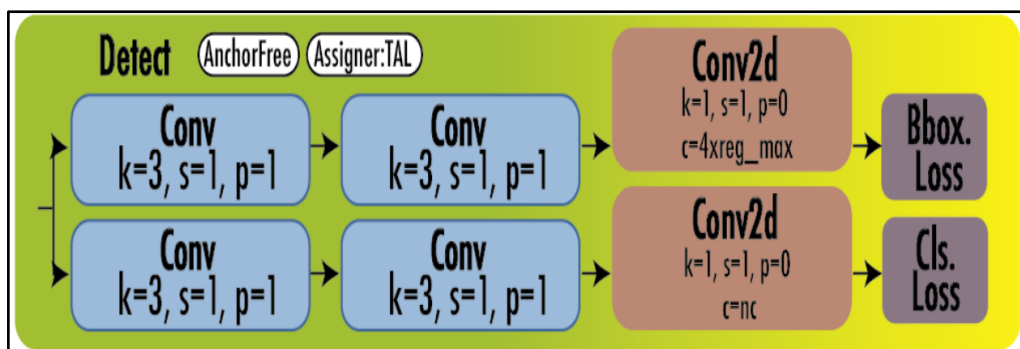


Gambar 4. 1 Kurva Metrik Evaluasi

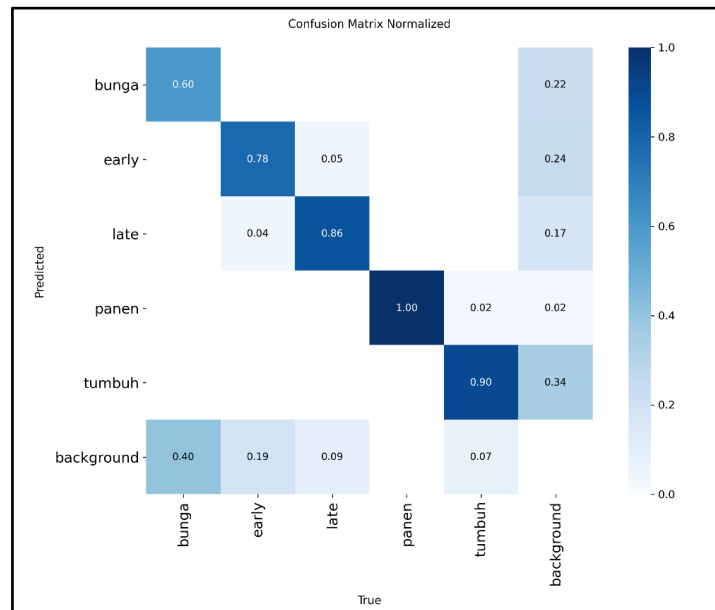
Pada Gambar 4.1 ditampilkan kurva *Precision*, *Recall*, *mAP@0.5*, *mAP@0.5:0.95*, dan *Loss* model *YOLOv8* selama proses pelatihan dengan konfigurasi terbaik yaitu *epochs* = 150, *batch size* = 16, dan *optimizer* = *Adam*. Kurva menunjukkan bahwa nilai metrik evaluasi cenderung meningkat secara stabil sejak *epoch* awal dan mulai mencapai konvergensi pada sekitar *epoch* ke-100, di mana kurva *Precision*, *Recall*, dan *mAP@0.5* mulai melandai dan tidak menunjukkan peningkatan signifikan hingga *epoch* ke-150, dengan nilai *mAP@0.5* akhir sebesar 0.8588. Sementara itu, *training loss* dan *validation loss* (*box loss*, *cls loss*, *dfl loss*) keduanya menunjukkan tren penurunan yang konsisten tanpa gap yang signifikan antara keduanya. Pola kurva ini mengindikasikan bahwa model

telah berhasil mempelajari pola visual fase pertumbuhan tanaman melon tanpa mengalami *overfitting* yang signifikan.

Ketiga jenis *loss* yang ditampilkan pada Gambar 4.1, yaitu *box loss*, *cls loss*, dan *dfl loss*, merupakan hasil proses langsung dari struktur *decoupled head* pada arsitektur YOLOv8 yang telah dijelaskan pada Sub-bab 3.4.1. *Box loss* dan *dfl* (*Distribution Focal Loss*) dihasilkan oleh cabang regresi yang bertugas memprediksi posisi dan ukuran *bounding box* secara langsung tanpa *anchor box*, sedangkan *cls loss* dihasilkan oleh cabang klasifikasi yang bertugas menentukan kelas fase pertumbuhan secara independen. Tren penurunan yang konsisten pada ketiga *loss* tersebut, baik pada data *training* maupun *validation*, mengindikasikan bahwa kedua cabang pada *head* bekerja secara seimbang tanpa salah satu cabang mendominasi proses pembelajaran, sehingga model dapat mempelajari posisi dan kelas objek secara bersamaan dengan baik. Cabang regresi dan klasifikasi pada bagian head yang menghasilkan nilai *loss* dapat dilihat pada Gambar 4.2 berikut.



Gambar 4. 2 Cabang Regresi dan Cabang Klasifikasi pada Bagian *Head* di Arsitektur YOLOv8

Gambar 4. 3 *Confusion Matrix*

Pada Gambar 4.3 ditampilkan *confusion matrix* ternormalisasi model *YOLOv8* pada konfigurasi parameter terbaik. Hasil menunjukkan bahwa sebagian besar kelas fase pertumbuhan terdeteksi dengan tepat, yang terlihat dari nilai diagonal matrik yang dominan. Kelas panen dan tumbuh menunjukkan tingkat deteksi tertinggi dengan nilai masing-masing 1.00 dan 0.90, yang mengindikasikan bahwa karakteristik visual kedua fase tersebut paling mudah dibedakan oleh model. Sementara itu, kelas bunga dan *early* memiliki potensi kesalahan klasifikasi yang lebih tinggi dengan nilai deteksi masing-masing 0.60 dan 0.78, di mana sebagian besar kesalahannya berupa misklasifikasi ke kelas *background*, bunga salah terdeteksi sebagai *background* sebesar 0.40, dan *early* sebesar 0.19. Hal ini kemungkinan disebabkan oleh kemiripan karakteristik visual kedua fase tersebut dengan kondisi latar belakang pada lingkungan CCTV yang digunakan.

Perbedaan tingkat akurasi antar kelas ini dapat dijelaskan melalui mekanisme deteksi multi-skala pada bagian *neck* arsitektur *YOLOv8*. Objek bunga

melon berukuran relatif kecil dan tipis dibandingkan objek panen dan tumbuh yang berupa buah berukuran besar dengan warna mencolok, sehingga fitur bunga lebih banyak ditangkap pada skala deteksi resolusi tinggi (P3) yang rentan terhadap *noise* visual dan kemiripan tekstur dengan latar belakang daun. Sebaliknya, objek panen dan tumbuh lebih dominan terdeteksi pada skala P4 dan P5 yang berfokus pada objek berukuran sedang hingga besar, sehingga fitur warna dan bentuknya lebih mudah dibedakan oleh modul C2f dan SPPF pada *backbone*. Hal ini menjelaskan mengapa kesalahan klasifikasi pada kelas bunga dan *early* lebih banyak terjadi berupa misklasifikasi ke kelas background, sebagaimana ditunjukkan pada Gambar 4.3 sebelumnya.

Berdasarkan grafik kurva metrik, *confusion matrix*, serta *kurva loss* yang telah ditampilkan, konfigurasi parameter terbaik yang diperoleh secara sistematis melalui pendekatan pengujian dua tahap ini terbukti menghasilkan model yang *konvergen* dan mampu membedakan setiap fase pertumbuhan tanaman melon dengan baik. Konfigurasi tersebut selanjutnya digunakan sebagai model akhir yang diintegrasikan ke dalam sistem visualisasi *digital model* berbasis deteksi YOLOv8 untuk visualisasi fase pertumbuhan tanaman melon berdasarkan citra kamera CCTV yang diunggah pengguna.

4.3 Hasil Pengujian Deteksi Model YOLOv8

Subbab ini menyajikan hasil pengujian deteksi model YOLOv8 dalam mengenali fase pertumbuhan tanaman melon berdasarkan citra kamera CCTV yang diunggah pengguna ke dalam sistem *Unity*. Pengujian dilakukan dengan mengunggah citra uji ke dalam sistem, kemudian sistem memproses citra melalui

alur *preprocessing letterbox*, inferensi model *ONNX* via *Unity Sentis*, hingga penentuan fase, kemudian fase yang dihasilkan sistem dibandingkan dengan fase asli untuk menentukan apakah hasil deteksi termasuk benar atau salah. Fase yang diuji meliputi *early vegetatif*, *late vegetatif*, pembungaan, pertumbuhan buah, dan panen. Hasil pengujian disajikan pada Tabel 4.4 berikut.

Tabel 4. 4 Hasil Pengujian Deteksi Model *YOLOv8*

No	Citra Uji	Fase Asli	Fase Terdeteksi	Status
1		<i>Early vegetatif</i>	<i>Early vegetatif</i>	Benar
2		<i>Early vegetatif</i>	<i>Early vegetatif</i>	Benar
3		<i>Late vegetatif</i>	<i>Late vegetatif</i>	Benar
4		<i>Late vegetatif</i>	<i>Late vegetatif</i>	Benar
5		Pembungaan	Pembungaan	Benar
6		Pembungaan	Pembungaan	Benar

No	Citra Uji	Fase Asli	Fase Terdeteksi	Status
7		Pertumbuhan Buah	Pertumbuhan Buah	Benar
8		Pertumbuhan Buah	Pertumbuhan Buah	Benar
9		Panen	Panen	Benar
10		Panen	Pertumbuhan Buah	Salah

Berdasarkan hasil pengujian pada Tabel 4.4, model *YOLOv8* menunjukkan kemampuan dalam mendeteksi fase pertumbuhan tanaman melon sesuai dengan citra uji yang diberikan. Pada setiap citra, sistem menghasilkan sejumlah deteksi objek yang kemudian diakumulasikan per kelas sebagai dasar penentuan fase akhir, jumlah akumulasi ini ditampilkan sebagai total deteksi pada *panel* hasil deteksi di antarmuka *Unity*.

Kesalahan deteksi terjadi pada citra uji nomor 10, di mana jika dilihat secara kasat mata tidak terdapat perbedaan kondisi tanaman apabila dibandingkan dengan citra uji nomor 9. Kesalahan deteksi ini disebabkan oleh waktu pengambilan gambar pada citra uji nomor 10 yang dilakukan pada siang hari (pukul 13:05 WIB), di mana kondisi pencahayaan yang masuk ke dalam *greenhouse* lebih terang

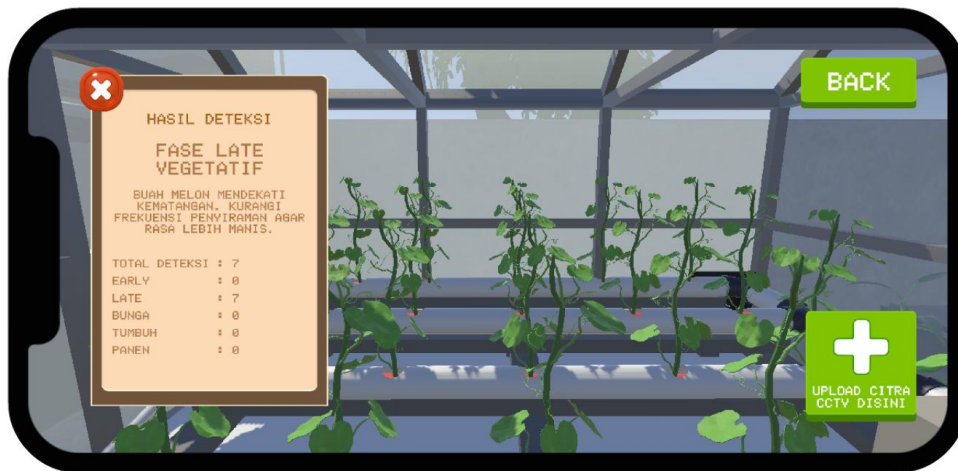
sehingga dapat mengubah tampilan warna buah melon menjadi sedikit kehijauan saat pengambilan gambar berlangsung. Berbeda dengan pengambilan gambar pada citra uji nomor 9 yang dilakukan pada pagi hari (pukul 08:05 WIB), di mana pencahayaan yang masuk ke dalam *greenhouse* tidak terlalu terang. Hal ini membuktikan bahwa kondisi pencahayaan dapat memengaruhi hasil deteksi sistem.

Contoh hasil deteksi secara visual disajikan pada Gambar 4.4 hingga Gambar 4.8 berikut.



Gambar 4. 4 Contoh Visual Fase *Early Vegetatif* Citra Uji Nomor 1

Pada Gambar 4.4 terlihat bahwa sistem berhasil mendeteksi fase *early vegetatif* untuk kedua citra uji yang ditunjukkan melalui *panel* hasil deteksi pada antarmuka *Unity*. Contoh citra uji nomor 1 memunculkan *panel* yang menampilkan total sebesar 11 deteksi dengan seluruh deteksi mengarah pada kelas *early*, sehingga sistem memutuskan fase *early vegetatif* dan menampilkan *prefab EarlyPrefab* sebagai representasi visual dalam bentuk *digital model*. Kondisi visual tanaman pada fase ini dicirikan oleh daun yang masih berukuran kecil dan belum berkembang sempurna.



Gambar 4. 5 Contoh Visual Fase *Late Vegetatif* Citra Uji Nomor 3

Pada Gambar 4.5 terlihat bahwa sistem berhasil mendeteksi fase *late vegetatif* untuk kedua citra uji yang ditunjukkan melalui *panel* hasil deteksi pada antarmuka *Unity*. Contoh citra uji nomor 3 memunculkan *panel* yang menampilkan total sebesar 7 deteksi dengan seluruh deteksi mengarah pada kelas *late*, sehingga sistem memutuskan fase *late vegetatif* dan menampilkan *prefab LatePrefab* sebagai representasi visual dalam bentuk *digital model*. Kondisi visual fase ini dicirikan dengan daun yang telah berbentuk menjari sempurna dan lebih lebat dibandingkan fase *early vegetatif*.



Gambar 4. 6 Contoh Visual Fase Pembungaan Citra Uji Nomor 5

Pada Gambar 4.6 terlihat bahwa sistem berhasil mendeteksi fase pembungaan untuk kedua citra uji yang ditunjukkan melalui *panel* hasil deteksi pada antarmuka *Unity*. Contoh citra uji nomor 5 memunculkan *panel* yang menampilkan total sebesar 11 deteksi dengan deteksi dominan pada kelas bunga yang memenuhi ambang minimum tetapi deteksi kelas panen tidak memenuhi ambang *minimum*, sehingga sistem memutuskan fase pembungaan dan menampilkan *prefab BungaPrefab* sebagai representasi visual dalam bentuk *digital model*. Kondisi visual fase ini dicirikan dengan kemunculan bunga berwarna kuning pada tanaman melon.



Gambar 4. 7 Contoh Visual Fase Petumbuhan Buah Citra Uji Nomor 7

Pada Gambar 4.7 terlihat bahwa sistem berhasil mendeteksi fase pertumbuhan buah untuk kedua citra uji yang ditunjukkan melalui *panel* hasil deteksi pada antarmuka *Unity*. Contoh citra uji nomor 7 memunculkan *panel* yang menampilkan total sebesar 8 deteksi dengan deteksi dominan pada kelas tumbuh yang memenuhi ambang minimum tetapi kelas panen tidak memenuhi ambang minimum, sehingga sistem memutuskan fase pertumbuhan buah dan menampilkan

prefab TumbuhPrefab sebagai representasi visual dalam bentuk *digital model*. Kondisi visual fase ini dicirikan dengan kemunculan buah melon yang masih berwarna hijau dan berukuran kecil.



Gambar 4. 8 Contoh Visual Fase Panen Citra Uji Nomor 9

Pada Gambar 4.8 terlihat bahwa sistem berhasil mengidentifikasi fase panen, ditunjukkan panel hasil deteksi pada antarmuka Unity. Contoh citra uji nomor 9 memunculkan panel yang menampilkan total 13 deteksi dengan deteksi kelas panen yang memenuhi ambang minimum, sehingga sistem memutuskan fase panen dan memunculkan prefabs *PanenPrefabs* sebagai representasi visual dalam bentuk digital model. Kondisi visual fase ini dicirikan dengan warna buah melon yang kuning keemasan sesuai karakteristik varietas golden melon.

Rekapitulasi hasil pengujian deteksi model *YOLOv8* dapat disajikan pada Tabel 4.5 berikut.

Tabel 4. 5 Rekapitulasi Hasil Pengujian Deteksi Model *YOLOv8*

Keterangan	Jumlah
Jumlah citra uji	10
Deteksi benar	9
Deteksi salah	1
Akurasi pengujian	90%

Tingkat keberhasilan pengujian dihitung menggunakan Persamaan 4. 1 persamaan berikut:

$$Akurasi = \frac{Jumlah\ Deteksi\ Benar}{Jumlah\ Seluruh\ Citra\ Uji} \times 100\% \quad (4. 1)$$

Berdasarkan perhitungan tersebut, diperoleh nilai akurasi sebesar 90%, yang menunjukkan bahwa model *YOLOv8* mampu mendeteksi fase pertumbuhan tanaman melon dengan tepat pada 9 dari 10 citra uji yang digunakan.

4.4 Hasil Pengujian Komponen Visualisasi *Digital Model* (Penentuan Fase Untuk Visualisasi Fase Pertumbuhan Tanaman Melon)

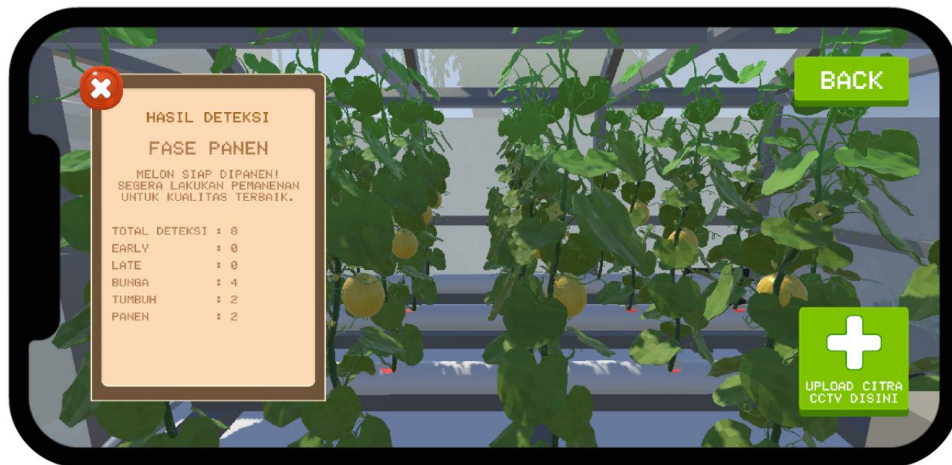
Subbab ini menyajikan hasil pengujian komponen visualisasi *digital model* pada bagian penentuan fase untuk visualisasi pertumbuhan tanaman melon berdasarkan mekanisme penentuan fase yang telah dirancang. Pengujian tidak dilakukan menggunakan citra CCTV secara langsung, melainkan melalui simulasi *input* nilai deteksi per kelas secara manual menggunakan *script debug* yang dikembangkan khusus untuk keperluan pengujian ini di lingkungan *Unity*. Pendekatan ini dipilih agar setiap mekanisme dapat diuji dengan kombinasi nilai deteksi yang dapat dikontrol sepenuhnya, sehingga kebenaran logika mekanisme penentuan fase dapat diverifikasi secara sistematis tanpa bergantung pada variabilitas hasil deteksi model. Kombinasi nilai deteksi pada setiap kelas, yaitu *early vegetatif*, *late vegetatif*, pembungaan, pertumbuhan buah, dan panen, ditentukan sesuai skenario yang telah dirancang pada Bab 3, kemudian fase yang dihasilkan sistem dibandingkan dengan fase yang diharapkan. Mekanisme yang diuji meliputi *minimum count*, *fallback majority vote*, dan *tiebreaker* berdasarkan

urutan kematangan biologis. Selain penentuan fase, sistem juga diamati untuk memastikan bahwa *prefab 3D* yang ditampilkan pada *Unity* telah sesuai dengan fase akhir yang diputuskan. Hasil pengujian disajikan pada Tabel 4.6 berikut.

Tabel 4. 6 Hasil Pengujian Komponen Visualiasi *Digital Model* (Penentuan Fase untuk Visualisasi Fase Pertumbuhan Tanaman Melon)

No	Skenario	<i>early</i>	<i>late</i>	<i>bunga</i>	<i>tumbuh</i>	<i>panen</i>	Fase/ <i>Prefab</i> Diharapkan	Fase/ <i>Prefab</i> Dihasilkan	Status
Minimum Count (jumlah minimal)									
1	Satu fase memenuhi <i>minimum count</i>	0	0	0	0	2	Panen (<i>PanenPrefab</i>)	Panen (<i>PanenPrefab</i>)	Berhasil
2	Dua fase memenuhi <i>minimum count</i> , prioritas kematangan biologis	0	0	0	2	2	Panen (<i>PanenPrefab</i>)	Panen (<i>PanenPrefab</i>)	Berhasil
3	Tiga fase memenuhi <i>minimum count</i> , prioritas kematangan biologis	0	0	4	2	2	Panen (<i>PanenPrefab</i>)	Panen (<i>PanenPrefab</i>)	Berhasil
Fallback Majority Vote (Berdasarkan jumlah deteksi ketika tidak memenuhi mekanisme <i>minimum count</i>)									
4	Tidak ada yang memenuhi <i>minimum count</i> , <i>spawn</i> dengan jumlah deteksi terbanyak	0	0	2	1	0	Pembungaan (<i>BungaPrefab</i>)	Pembungaan (<i>BungaPrefab</i>)	Berhasil
Skenario Seri (Berdasarkan urutan kematangan biologis)									
5	Seri dua fase, prioritas kematangan biologis	0	0	1	1	0	Pertumbuhan buah (<i>TumbuhPrefab</i>)	Pertumbuhan buah (<i>TumbuhPrefab</i>)	Berhasil
6	Tidak ada deteksi sama sekali	0	0	0	0	0	<i>Unknown</i> (tidak muncul <i>prefab</i>)	<i>Unknown</i> (tidak muncul <i>prefab</i>)	Berhasil

Untuk memperjelas hasil pengujian pada setiap mekanisme, berikut ditampilkan tangkapan layar sistem yang menunjukkan kesesuaian antara fase yang dihasilkan dengan fase yang diharapkan pada masing-masing skenario.



Gambar 4. 9 Skenario *Minimum Count* (Skenario 1,2,3)

Pada Gambar 4.9 ditampilkan hasil pengujian mekanisme *minimum count* yang mencakup skenario 1, 2, dan 3. Ketiga skenario menunjukkan kondisi di mana satu hingga tiga kelas memenuhi ambang *minimum* secara bersamaan yaitu, kelas bunga sebesar 4 deteksi, kelas tumbuh sebesar 2 deteksi, dan kelas panen sebesar 2 deteksi. Sistem memprioritaskan fase dengan tingkat kematangan biologis tertinggi yaitu fase panen, sehingga *prefab PanenPrefab* ditampilkan pada *Unity* sesuai dengan fase yang diharapkan. Hasil ini membuktikan bahwa mekanisme *minimum count* berjalan sesuai urutan prioritas biologis yang telah dirancang.



Gambar 4. 10 Skenario *Fallback Majority Vote* (Skenario 4)

Pada Gambar 4.10 ditampilkan hasil pengujian mekanisme *fallback majority vote* pada skenario 4, yaitu kondisi ketika tidak ada kelas yang memenuhi *minimum count*. Sistem menerima *input* deteksi dengan jumlah terbanyak pada kelas bunga sebesar 2 deteksi, sehingga sistem memilih fase pembungaan sebagai keputusan akhir. *Prefab BungaPrefab* ditampilkan pada *Unity* sesuai dengan fase yang diharapkan. Hasil ini membuktikan bahwa mekanisme *fallback* berjalan sebagai cadangan yang efektif ketika mekanisme *minimum* tidak terpenuhi.



Gambar 4. 11 Skenario Seri dan *Unknown* (Skenario 5 dan 6)

Pada Gambar 4.11 ditampilkan hasil pengujian skenario seri yang mencakup dua kondisi, yaitu seri dua fase dan kondisi tanpa deteksi sama sekali. Pada skenario 5, sistem menerima *input* dengan jumlah deteksi yang sama antara fase bunga dan fase tumbuh masing-masing sebesar 1 deteksi, sehingga sistem menerapkan *tiebreaker* berdasarkan urutan kematangan biologis dan memilih fase pertumbuhan buah sebagai keputusan akhir. *Prefab Tumbuh* *Prefab* ditampilkan pada *Unity* sesuai dengan fase yang diharapkan. Pada skenario 6, sistem tidak menerima deteksi pada kelas manapun sehingga menghasilkan status *Unknown* dan tidak menampilkan *prefab* apapun, yang juga sesuai dengan perilaku yang diharapkan.

Rekapitulasi hasil pengujian komponen visualisasi *digital model* disajikan pada Tabel 4.7 berikut.

Tabel 4. 7 Rekapitulasi Hasil Pengujian Komponen Visualisai *Digital Model*

Keterangan	Jumlah
Jumlah skenario uji	6
Skenario berhasil	6
Skenario tidak berhasil	0
Tingkat keberhasilan	100%

Tingkat keberhasilan pengujian dihitung menggunakan persamaan 4.2 berikut.

$$\text{Tingkat keberhasilan} = \frac{\text{Jumlah Skenario Berhasil}}{\text{Jumlah Seluruh Skenario Uji}} \times 100\% \quad (4. 2)$$

Berdasarkan perhitungan tersebut, diperoleh tingkat keberhasilan sebesar 100%, yang menunjukkan bahwa mekanisme penentuan fase dalam sistem visualisasi *digital model* telah berjalan sesuai dengan seluruh skenario yang dirancang.

4.5 Evaluasi User

Subbab ini menyajikan hasil evaluasi *user* terhadap aplikasi visualisasi fase pertumbuhan tanaman melon berbasis *mobile* menggunakan metode *User Version of the Mobile Application Rating Scale* (uMARS). Evaluasi dilakukan oleh 35 responden yang mencoba aplikasi secara langsung, mulai dari mengunggah citra kamera CCTV tanaman melon melalui antarmuka aplikasi, mengamati proses deteksi fase menggunakan model *YOLOv8*, hingga melihat visualisasi *digital model* pada antarmuka *Unity*, kemudian mengisi kuesioner uMARS yang terdiri dari 20 butir pernyataan dengan skala *Likert* 1-5. Aspek yang dievaluasi meliputi *engagement*, *functionality*, *aesthetics*, *information quality*, dan *subjective quality*.

4.5.1 Hasil Kuisisioner uMARS

Skor uMARS masing-masing responden dihitung berdasarkan rata-rata jawaban per subskala, kemudian dirata-ratakan ke dalam skor uMARS total. Sebagai contoh, apabila Responden 1 memberikan jawaban dengan total skor pada masing-masing subskala adalah $S_Eng = 4,60$, $S_Fun = 4,00$, $S_Aes = 4,33$, $S_Inf = 4,75$, dan $S_Sub = 3,50$, maka skor uMARS Responden 1 adalah sebesar 4,24. Rekapitulasi skor uMARS seluruh responden disajikan pada Tabel 4.8 berikut.

Tabel 4. 8 Rekapitulasi Skor Rata-Rata per Subskala tiap Responden

Responden	<i>S_Eng</i>	<i>S_Fun</i>	<i>S_Aes</i>	<i>S_Inf</i>	<i>S_Sub</i>	<i>uMARS</i>
R1	4,60	4,00	4,33	4,75	3,50	4,24
R2	4,40	4,00	5,00	4,50	4,50	4,48
R3	4,80	4,00	4,33	4,50	4,50	4,43
R4	4,00	4,25	4,00	4,00	4,25	4,10
R5	4,00	3,25	2,33	3,75	3,75	3,42
R6	3,80	4,25	4,00	4,00	3,50	3,91
R7	4,00	4,25	3,00	3,50	3,50	3,65
R8	5,00	5,00	5,00	5,00	5,00	5,00
R9	4,40	3,75	4,00	3,75	3,75	3,93
R10	4,20	3,75	4,67	4,00	4,00	4,12
R11	3,00	3,25	5,00	4,00	4,25	3,90
R12	4,00	4,00	5,00	4,25	3,50	4,15

Responden	<i>S_Eng</i>	<i>S_Fun</i>	<i>S_Aes</i>	<i>S_Inf</i>	<i>S_Sub</i>	<i>uMARS</i>
R13	4,80	3,25	4,67	4,75	4,50	4,39
R14	3,80	3,75	4,00	3,50	4,00	3,81
R15	4,80	4,75	5,00	5,00	4,25	4,76
R16	3,80	3,25	4,33	4,25	3,25	3,78
R17	4,20	5,00	5,00	5,00	3,50	4,54
R18	5,00	5,00	4,25	5,00	4,00	4,65
R19	3,20	3,25	4,00	3,75	2,00	3,24
R20	3,40	3,50	4,00	4,50	2,75	3,63
R21	3,60	5,00	4,33	4,75	3,25	4,19
R22	3,60	4,00	3,67	3,75	4,25	3,85
R23	4,20	4,75	4,67	5,00	4,75	4,67
R24	4,20	5,00	3,67	5,00	3,00	4,17
R25	4,40	5,00	5,00	5,00	4,25	4,73
R26	3,60	4,25	4,00	3,75	3,50	3,82
R27	5,00	4,50	4,33	4,75	4,00	4,52
R28	5,00	4,25	5,00	4,50	4,75	4,70
R29	4,60	4,50	5,00	4,50	3,75	4,47
R30	3,80	4,50	4,00	5,00	3,25	4,11
R31	4,00	4,00	4,00	4,00	4,00	4,00
R32	4,00	4,00	4,33	2,75	4,00	3,82
R33	4,20	4,00	4,00	4,00	4,00	4,04
R34	3,60	4,00	4,00	4,00	3,75	3,87
R35	4,20	4,75	4,00	4,00	4,75	4,34
Rata-Rata	4,15	4,17	4,28	4,30	3,87	4,16

Nilai rata-rata skor uMARS keseluruhan responden dihitung menggunakan Persamaan 4.3 berikut.

$$\bar{X} = \frac{\sum uMARS_i}{n} \quad (4.3)$$

Berdasarkan perhitungan tersebut, diperoleh rata-rata skor uMARS sistem sebesar 4,16. Keterangan pernyataan (item 1-20) dan rekapitulasi skor skala likert 1-5 dari 20 item pernyataan keseluruhan responden terlampir pada Lampiran 1 dan Lampiran 2.

4.5.2 Interpretasi Skor uMARS

Nilai rata-rata skor uMARS yang diperoleh diinterpretasikan mengacu pada kategori penilaian uMARS yang dikemukakan oleh (Stoyanov et al., 2016), sebagaimana disajikan pada Tabel 4.9 berikut.

Tabel 4. 9 Interpretasi Skor uMARS

Rentang Skor	Kategori	Pertanyaan
1,00 – 1,99	Sangat Buruk	Aplikasi tidak memadai dan perlu perombakan menyeluruh
2,00 – 2,99	Buruk	Aplikasi memiliki banyak kekurangan yang harus diperbaiki
3,00 – 3,99	Cukup / Dapat Diterima	Aplikasi memenuhi standar minimal yang dapat diterima
4,00 – 4,49	Baik	Aplikasi berkualitas baik dan layak digunakan
4,50 – 5,00	Sangat Baik	Aplikasi berkualitas sangat tinggi dan sangat direkomendasikan

Berdasarkan rata-rata skor uMARS sebesar 4,16. aplikasi yang dikembangkan termasuk dalam kategori Baik, yang menunjukkan bahwa aplikasi berkualitas baik dan layak digunakan dari sisi *engagement*, *functionality*, *aesthetics*, *information quality*, dan *subjective quality* pengguna.

4.6 Pembahasan

Subbab ini membahas hasil pengujian sistem secara keseluruhan yang mencakup performa *tuning parameter* model *YOLOv8*, ketepatan deteksi fase pertumbuhan, keberhasilan logika penentuan fase pada komponen visualisasi *digital model*, serta hasil evaluasi pengguna menggunakan metode uMARS. Pembahasan dilakukan untuk menafsirkan hasil yang telah diperoleh pada subbab sebelumnya, sehingga dapat diketahui sejauh mana sistem yang dikembangkan telah memenuhi tujuan penelitian, yaitu memvisualisasikan perubahan fase pertumbuhan tanaman melon dalam bentuk *digital model* berdasarkan hasil deteksi *YOLOv8* menggunakan *Unity*.

Berdasarkan hasil pengujian *tuning parameter* pada Subbab 4.2, konfigurasi terbaik diperoleh pada *epochs* = 150, *batch size* = 16, dan *optimizer* = *Adam*, dengan nilai *mAP@0.5* sebesar 0.8588 dan *mAP@0.5:0.95* sebesar 0.5510. Hasil ini

menunjukkan bahwa penentuan *hyperparameter* berpengaruh cukup signifikan terhadap kemampuan model dalam mendeteksi fase pertumbuhan tanaman melon. Peningkatan jumlah *epochs* dari 50 hingga 150 secara umum berdampak positif terhadap performa model, yang terlihat dari tren kenaikan nilai $mAP@0.5$ dan *F1-Score* seiring bertambahnya *epochs*, tanpa disertai indikasi *overfitting* yang signifikan sebagaimana ditunjukkan oleh kurva *training loss* dan *validation loss* yang *konvergen*. Dari sisi *batch size*, nilai 16 menghasilkan $mAP@0.5$ lebih tinggi dibandingkan batch size 32 pada *epochs* yang sama (0.8771 dengan 0.8699 pada skenario 6), yang mengindikasikan bahwa frekuensi pembaruan bobot yang lebih tinggi pada *batch size* kecil memberikan sinyal *gradien* yang lebih beragam sehingga model lebih baik dalam menggeneralisasi pola visual *dataset*. Dari sisi *optimizer*, *Adam* terpilih sebagai yang terbaik karena mekanisme *adaptive learning rate* nya efektif menyesuaikan bobot secara individual untuk setiap parameter, sehingga model lebih cepat dan stabil dalam mencapai konvergensi pada *dataset* fase pertumbuhan tanaman melon yang digunakan.

Efektivitas *optimizer Adam* ini juga didukung oleh kedalaman arsitektur YOLOv8 yang terdiri atas banyak lapisan konvolusi pada *backbone* CSPDarknet53 dan *neck* PAN-FPN, di mana mekanisme *adaptive learning rate* pada *Adam* memungkinkan setiap lapisan menyesuaikan kecepatan pembelajarannya secara individual, sehingga proses konvergensi pada arsitektur yang dalam seperti ini dapat berlangsung lebih stabil dibandingkan *optimizer* dengan *learning rate* tetap seperti SGD.

Berdasarkan hasil pengujian deteksi pada Subbab 4.3, model *YOLOv8* memperoleh akurasi sebesar 90% dari 10 citra uji, dengan 9 citra terdeteksi benar dan 1 citra terdeteksi salah. Fase *early* vegetatif, *late* vegetatif, pembungaan, dan pertumbuhan buah seluruhnya terdeteksi dengan benar, yang menunjukkan bahwa model mampu mengenali karakteristik visual masing-masing fase tersebut secara konsisten. Kesalahan deteksi terjadi pada salah satu citra fase panen yang terklasifikasi sebagai fase pertumbuhan buah, yang kemungkinan disebabkan oleh kondisi pencahayaan yang memengaruhi tampilan warna buah sehingga kemiripan visual antara kedua fase tersebut meningkat dan model tidak mampu membedakannya secara optimal pada citra tersebut.

Berdasarkan hasil pengujian komponen visualiasi *digital model* pada Subbab 4.4, sistem memperoleh tingkat keberhasilan sebesar 100% dengan seluruh 6 skenario uji menghasilkan keputusan fase dan *spawn prefab* yang sesuai dengan yang diharapkan. Mekanisme *minimum count* terbukti mampu memprioritaskan fase dengan tingkat kematangan biologis tertinggi secara konsisten, sebagaimana ditunjukkan pada skenario 1 hingga 3 di mana satu hingga tiga kelas memenuhi ambang *minimum* secara bersamaan dan sistem tetap memilih fase panen sebagai keputusan akhir. Mekanisme *fallback majority vote* berfungsi efektif sebagai cadangan ketika tidak ada kelas yang memenuhi mekanisme *minimum count*, dengan memilih fase berdasarkan jumlah deteksi terbanyak sebagaimana dibuktikan pada skenario 4. Mekanisme *tiebreaker* berbasis urutan kematangan biologis berhasil menyelesaikan kondisi seri antar fase sebagaimana dibuktikan pada skenario 5, serta sistem menangani kondisi tidak terduga berupa tidak adanya

deteksi sama sekali dengan mengembalikan status *Unknown* tanpa menampilkan *prefab* apapun pada skenario 6, yang menunjukkan bahwa sistem mampu merespons berbagai situasi dengan baik. Hasil ini menunjukkan bahwa logika mekanisme penentuan fase yang dirancang efektif dalam menerjemahkan *output* deteksi menjadi keputusan fase yang stabil dan sesuai konteks pertumbuhan tanaman melon.

Berdasarkan hasil evaluasi pada Subbab 4.5, diperoleh rata-rata skor *uMARS* sebesar 4,16 yang menempatkan aplikasi pada kategori Baik berdasarkan interpretasi (Stoyanov et al., 2016). Hasil ini menunjukkan bahwa aplikasi tidak hanya berhasil secara teknis, tetapi juga mampu memberikan pengalaman penggunaan yang baik dari sisi seluruh subskala yang dievaluasi. Subskala *information quality* memperoleh nilai tertinggi sebesar 4,30, yang mengindikasikan bahwa informasi fase pertumbuhan yang disajikan dinilai akurat, jelas, dan relevan bagi pengguna. Subskala *aesthetics* memperoleh nilai 4,28, yang menunjukkan bahwa tampilan visual aplikasi dinilai menarik dan konsisten oleh pengguna. Sementara itu, subskala *subjective quality* memperoleh nilai terendah sebesar 3,87, meskipun masih berada dalam kategori baik, hal ini dapat menjadi masukan spesifik untuk meningkatkan kesan keseluruhan aplikasi dari perspektif pengguna pada pengembangan selanjutnya.

Secara keseluruhan, seluruh komponen sistem yang diuji menunjukkan hasil yang sesuai dengan tujuan penelitian. Model *YOLOv8* dengan konfigurasi terbaik mencapai nilai *mAP@0.5* sebesar 0,8588 dengan akurasi deteksi fase pertumbuhan tanaman melon sebesar 90%, logika mekanisme penentuan fase pada komponen

visualiasi *digital model* berhasil menghasilkan keputusan fase yang tepat pada seluruh skenario uji dengan tingkat keberhasilan 100%, dan evaluasi pengguna melalui metode *uMARS* menghasilkan rata-rata skor 4,16 yang termasuk dalam kategori Baik. Ketiga hasil pengujian ini membuktikan bahwa sistem visualisasi *digital model* berbasis deteksi YOLOv8 yang menggunakan citra kamera CCTV mampu memvisualisasikan perubahan fase pertumbuhan tanaman melon secara dinamis, interaktif, dan informatif.

4.7 Integrasi Islam

4.7.1 *Muamalah Ma Allah*

Penelitian ini menghasilkan sistem visualisasi *digital model* pertumbuhan tanaman melon berbasis deteksi YOLOv8 yang telah diuji secara menyeluruh dari sisi performa deteksi, keberhasilan komponen visualisasi *digital model*, serta pengalaman pengguna, dengan seluruh hasil pengujian menunjukkan capaian yang baik. Capaian tersebut tidak terlepas dari proses penelitian yang dilakukan secara teliti, bertahap, dan sistematis. dimulai dari *tuning hyperparameter*, perancangan logika penentuan fase, hingga evaluasi pengguna. Kesungguhan dalam setiap tahapan tersebut dilandasi oleh kesadaran bahwa setiap amal dan pekerjaan manusia senantiasa berada dalam pengawasan Allah *Subhanahu Wa Ta'ala*. Kesadaran inilah yang menjadi inti dari muamalah ma Allah, yakni menjaga hubungan dengan Allah melalui niat yang lurus, sikap ihsan dalam bekerja, dan rasa syukur atas segala kemampuan yang dianugerahkan-Nya, sebagaimana ditegaskan dalam firman Allah *Subhanahu Wa Ta'ala* pada Surat An-Naml ayat 88:

وَتَرَى الْجِبَالَ تَحْسِبُهَا جَamِدَةً وَهِيَ تَمُرُّ مَرَّ السَّحَابِ صُنِعَ اللَّهُ الَّذِي اتَّقَنَ كُلَّ شَيْءٍ إِنَّهُ خَبِيرٌ بِمَا تَفْعَلُونَ

“Engkau akan melihat gunung-gunung yang engkau kira tetap di tempatnya, padahal ia berjalan seperti jalannya awan. (Demikianlah) penciptaan Allah menjadikan segala sesuatu dengan sempurna. Sesungguhnya Dia Mahateliti terhadap apa yang kamu kerjakan.” (QS. An-Naml: 88).

Menurut Tafsir Al-Wajiz karya Syaikh Prof. Dr. Wahbah Az-Zuhaili, penggalan akhir ayat ini menegaskan bahwa Allah Maha Mengetahui segala apapun yang dikerjakan manusia, baik maupun buruk, dan Dia akan memberikan balasan atas setiap perbuatan tersebut. Sifat *al-khabīr* yang disandang Allah bermakna Maha Mengetahui terhadap segala sesuatu, baik yang tampak maupun yang tidak tampak (Az-zuhaili, 2016). Penegasan ini mengandung makna yang sangat dalam dalam konteks muamalah ma Allah, bahwa tidak ada satu pun ikhtiar manusia yang luput dari pengetahuan dan penilaian-Nya. Oleh karena itu, bekerja dengan sungguh-sungguh dan penuh kejujuran bukan semata karena tuntutan duniawi, melainkan merupakan bentuk pertanggungjawaban seorang hamba kepada Tuhannya inilah hakikat muamalah ma Allah yang sesungguhnya.

Dalam konteks penelitian ini, kesadaran tersebut tercermin dari proses *tuning* yang dilakukan secara bertahap dan jujur untuk menemukan konfigurasi terbaik, logika mekanisme penentuan fase yang dirancang secara berlapis untuk menangani berbagai kondisi deteksi, serta evaluasi yang dilakukan secara komprehensif dari sisi teknis maupun pengguna. Seluruh ikhtiar tersebut diniatkan sebagai bentuk ibadah dan rasa syukur atas akal dan ilmu yang dikaruniakan Allah *Subhanahu Wa Ta’ala*, dengan keyakinan penuh bahwa Allah senantiasa

mengetahui setiap proses yang dikerjakan, dan bahwa hasil akhirnya hanyalah bagian dari pertolongan dan izin-Nya semata.

4.7.2 *Muamalah Ma'annas*

Selain membangun hubungan dengan Allah melalui niat dan kesungguhan dalam berkarya, penelitian ini juga memiliki dimensi sosial yang erat kaitannya dengan *muamalah ma'annas*, yakni hubungan manusia dengan sesama manusia. Sistem visualiasi *digital model* fase pertumbuhan tanaman melon berbasis deteksi YOLOv8 yang dihasilkan dalam penelitian ini tidak hanya bernilai teknis semata, tetapi dirancang untuk memberikan manfaat nyata bagi para petani melon, praktisi pertanian, dan masyarakat luas yang bergantung pada sektor hortikultura. Dorongan untuk menghadirkan karya ilmiah yang bermanfaat bagi sesama ini sejalan dengan perintah Allah *Subhanahu Wa Ta'ala* dalam penggalan ayat 99 Surat Al-An'am:

انظُرُوا إِلَى ثَمَرِهِ إِذَا أَثْمَرَ وَيَنْعِهِ إِنَّ فِي ذَٰلِكُمْ لَآيَاتٍ لِّقَوْمٍ يُؤْمِنُونَ

“Perhatikanlah buahnya pada waktu berbuah dan menjadi masak. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda (kekuasaan Allah) bagi kaum yang beriman..” (QS. Al-An'am: 99).

Menurut *Tafsir Al-Wajiz* karya Syaikh Prof. Dr. Wahbah az-Zuhaili, ayat ini mengajak manusia untuk memperhatikan buah ketika tumbuhan berbuah serta perkembangannya sesuai dengan bentuknya, karena sesungguhnya dalam hal yang demikian itu terdapat dalil-dalil atas kesempurnaan kuasa sang *Khaliq* bagi kaum yang mengimani keberadaan dan kuasa Allah (Az-zuhaili, 2016). Perintah untuk memperhatikan dan mengamati proses pertumbuhan tanaman dalam ayat ini bukan sekadar anjuran kesadaran individu, melainkan mengandung implikasi sosial yang

luas. hasil pengamatan tersebut hendaknya dikelola, dikembangkan menjadi ilmu yang bermanfaat, dan disebarluaskan demi kemaslahatan sesama manusia.

Dalam konteks inilah penelitian ini menemukan relevansinya sebagai wujud muamalah ma'annas. Sistem yang dikembangkan mampu mengenali dan memvisualisasikan fase-fase pertumbuhan tanaman melon secara otomatis melalui model YOLOv8 dan komponen visualiasi *digital model*, sehingga informasi pertumbuhan yang sebelumnya hanya dapat diketahui oleh petani berpengalaman kini dapat diakses secara lebih mudah, akurat, dan efisien oleh siapapun. Dengan demikian, penelitian ini berkontribusi pada upaya pemerataan akses terhadap teknologi pertanian modern sekaligus menjadi bentuk nyata dari semangat berbagi ilmu dan kepedulian terhadap sesama, yang pada akhirnya berpotensi meningkatkan produktivitas dan kesejahteraan para petani melon sebagai cerminan nilai Islam bahwa sebaik-baik manusia adalah yang paling bermanfaat bagi orang lain.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil membangun sistem visualisasi perubahan fase pertumbuhan tanaman melon berbasis Digital Model yang mengintegrasikan deteksi objek YOLOv8 dengan representasi tiga dimensi menggunakan Unity, di mana seluruh fase pertumbuhan tanaman dapat divisualisasikan secara dinamis melalui mekanisme penentuan fase dan prefab tiga dimensi yang berjalan sesuai rancangan. Model deteksi YOLOv8 yang dikembangkan, dengan konfigurasi terbaik diperoleh melalui pengujian tuning parameter dua tahap, menunjukkan kinerja yang baik dalam mengenali fase pertumbuhan tanaman melon, dengan tingkat akurasi pengujian yang tergolong tinggi meskipun masih ditemukan sedikit kesalahan deteksi pada beberapa citra uji.

Mekanisme penentuan fase yang menggabungkan pendekatan *minimum count*, *fallback majority vote*, dan *tiebreaker* berbasis urutan kematangan biologis terbukti efektif dalam menerjemahkan output deteksi menjadi keputusan fase yang stabil pada seluruh skenario uji yang dilakukan. Sementara itu, evaluasi pengguna menggunakan instrumen uMARS menunjukkan bahwa sistem ini diterima secara positif oleh pengguna dan dikategorikan pada tingkat Baik berdasarkan interpretasi (Stoyanov et al., 2016), yang menunjukkan bahwa sistem tidak hanya berhasil secara teknis tetapi juga mampu memberikan pengalaman penggunaan yang baik pada *platform mobile*. Secara keseluruhan, sistem visualisasi digital model berbasis

deteksi YOLOv8 menggunakan Unity yang dikembangkan terbukti mampu memvisualisasikan perubahan fase pertumbuhan tanaman melon secara dinamis, interaktif, dan informatif berdasarkan citra kamera CCTV yang diunggah pengguna ke dalam sistem.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan serta berbagai temuan selama proses pengujian, terdapat beberapa hal yang dapat dijadikan masukan untuk pengembangan dan penyempurnaan sistem ke depan. Saran-saran ini ditujukan untuk meningkatkan performa, akurasi, serta fungsionalitas sistem visualisasi *digital model* berbasis deteksi YOLOv8 menggunakan *Unity* dalam menghadapi kondisi yang lebih beragam.

1. Pengembangan Jumlah dan Variasi *Dataset* yang Lebih Lengkap

Dataset yang digunakan masih terbatas pada satu greenhouse dengan kondisi pencahayaan dan sudut pengambilan yang relatif seragam. Untuk penelitian selanjutnya, disarankan memperluas dataset dengan variasi kondisi pencahayaan, sudut kamera, dan varietas melon yang berbeda guna meningkatkan kemampuan generalisasi model, khususnya dalam membedakan fase yang memiliki kemiripan visual seperti fase pertumbuhan buah dan panen.

2. Implementasi Koneksi Data Otomatis

Sistem yang dikembangkan saat ini masih mengandalkan unggahan citra manual oleh pengguna. Untuk penelitian selanjutnya, disarankan mengimplementasikan koneksi data otomatis antara kamera dan sistem melalui server, sehingga pembaruan representasi visual dapat berlangsung secara

otomatis tanpa interaksi pengguna dan sistem dapat berkembang menuju kategori sistem *monitoring real-time*.

3. Penambahan Analisis Data, dan Representasi Visual

Untuk penelitian selanjutnya, disarankan menambahkan fitur analisis data pertumbuhan seperti estimasi ukuran dan berat buah berbasis deteksi citra, serta meningkatkan representasi visual dengan visualisasi per tanaman secara individual dan animasi transisi antarpase yang lebih realistis, sehingga sistem dapat berfungsi sebagai sarana pemantauan yang lebih akurat dan informatif terhadap kondisi nyata setiap tanaman.

DAFTAR PUSTAKA

- Aini, Q., Firmansyah, R., & Arif, Y. M. (2024). *Applying convolutional neural network and Nadam optimization in flower classification*. 13(4), 2865–2877. <https://doi.org/10.11591/eei.v13i4.6203>
- Arif, Y. M., & Harini, S. (2021). An Automatic Scenario Control In Serious Game To Visualize Tourism Destinations Recommendation. *IEEE Access*, PP, 1. <https://doi.org/10.1109/ACCESS.2021.3091425>
- Az-zuhaili, W. (2016). *Tafsir AL-Wajiz Jilid 2 Bagian 1 (I)*. Lajnah Pentashihan Mushaf Al-Qur'an.
- Boahen, J. O. (2024). Advancements in Precision Agriculture: Integrating Computer Vision for Intelligent Soil and Crop Monitoring in the Era of Artificial Intelligence. In *Interantional Journal of Scientific Research in Engineering and Management* (Vol. 08, Issue 03, pp. 1–5). <https://doi.org/10.55041/ijrsrem29725>
- Buyuksalih, I., Bayburt, S., Buyuksalih, G., Baskaraca, A. P., Karim, H., Rahman, A. A., S, B. I. M. A., Estate, R., & Bahru, J. (2017). *3d Modelling And Visualization Based On The Unity Game Engine – IV*(October), 14–15.
- Jupiter, F., Negara, E. S., Kunang, Y. N., & Herdiansyah, M. I. (2023). *Implementasi Algoritma CNN dan YOLO untuk Mendeteksi Jenis Kendaraan pada Jalan Raya*. 14(2), 110–117.
- Keil, J., Edler, D., Schmitt, T., & Dickmann, F. (2021). Creating Immersive Virtual Environments Based on Open Geospatial Data and Game Engines. *KN - Journal of Cartography and Geographic Information*, 71(1), 53–65. <https://doi.org/10.1007/s42489-020-00069-6>
- Kim, J. G., Shin, S. H., & Chung, S. (2025). *growth stage classification of basil in multi-plant environments*. November, 1–13. <https://doi.org/10.3389/fpls.2025.1707985>
- Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W., Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W., Kritzinger, W., Henjes, J., & Gmbh, G. (2018). ScienceDirect Digital Digital Twin Twin in in manufacturing : manufacturing : A A categorical review Digital Twin in review classification. *IFAC-PapersOnLine*, 51(11), 1016–1022. <https://doi.org/10.1016/j.ifacol.2018.08.474>
- Moing, A., William Allwood, J., Aharoni, A., Baker, J., Beale, M. H., Ben-Dor, S., Biais, B., Brigante, F., Burger, Y., Deborde, C., Erban, A., Faigenboim, A., Gur, A., Goodacre, R., Hansen, T. H., Jacob, D., Katzir, N., Kopka, J., Lewinsohn, E., ... Schaffer, A. A. (2020). Comparative metabolomics and molecular phylogenetics of melon (*Cucumis melo*, cucurbitaceae)

- biodiversity. *Metabolites*, 10(3). <https://doi.org/10.3390/metabo10030121>
- Murtaza, S., Al, R., Adiba, H., Hossain, T., Saha, A. K., & Rahman, R. (2023). *Development of Interactive Data Visualization System in Three-Dimensional Immersive Space*. 14(10).
- Oktarina, Y., Wijaya, R., Dewi, T., & Risma, P. (2025). Aplikasi CNN untuk Analisis Visual Pertumbuhan Tanaman Bitter Melon dalam Sistem Akuaponik. *Jurnal Rekayasa Elektro Sriwijaya*, 6(2), 76–84. <https://doi.org/10.36706/jres.v6i2.152>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (n.d.). *You Only Look Once: Unified, Real-Time Object Detection*.
- Rif'an, M., Widyasunu, P., Widarawati, R., & Ummami, N. R. (2024). Pengaruh perbedaan nutrisi fosfor dan media tanam terhadap pertumbuhan tanaman melon (*Cucumis melo* L.) hidroponik sistem irigasi tetes. *Jurnal AGRO*, 11(1), 172–186. <https://doi.org/10.15575/31140>
- Roslan, M. I., & Ibrahim, Z. (2023). Fruit Detection and Recognition Using Faster R- CNN with FPN30 Pre-trained Network. *2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, 2023, 1–6. <https://doi.org/10.1109/ICRAIE59459.2023.10468169>
- Santoso, I., Alvin, Y., & Soetrisno, A. (2021). *Implementasi Deep Learning Menggunakan Metode Implementasi Deep Learning Menggunakan Metode Low Vision*. January 2023. <https://doi.org/10.14710/transient.v10i3.502-509>
- Schneider, F., Swiatek, J., & Jelali, M. (2024). *Detection of Growth Stages of Chilli Plants in a Hydroponic Grower Using Machine Vision and YOLOv8 Deep Learning Algorithms*.
- Stoyanov, S. R., Psych, M., Hides, L., Kavanagh, D. J., & Wilson, H. (2016). *Development and Validation of the User Version of the Mobile Application Rating Scale (uMARS) Corresponding Author : Related Article : 4*, 1–5. <https://doi.org/10.2196/mhealth.5849>
- Suri, S., & Nirwana AN, A. (2022). Konstruksi Metode Tafsir Ijmali: Kajian Terhadap Kitab At-Tafsir Al-Muyassar Karya ‘Aidh Al-Qarni. *AL QUDS: Jurnal Studi Alquran Dan Hadis*, 6(3), 1073–1090. <https://doi.org/10.29240/alquds.v6i3.4313>
- Terven, J. R., & Cordova-esparza, D. M. (2024). *A C OMPREHENSIVE R EVIEW OF YOLO A RCHITECTURES IN C OMPUTER V ISION : F ROM YOLO V 1 TO YOLO V 8 AND*. 1–36.
- Wei, M., Huang, Y., Mo, C., Wang, H., Zeng, Q., Yang, W., Chen, J., Zhang, X., & Kong, Q. (2023). Telomere-to-telomere genome assembly of melon (*Cucumis melo* L. var. *inodorus*) provides a high-quality reference for meta-QTL analysis of important traits. *Horticulture Research*, 10(10). <https://doi.org/10.1093/hr/uhad189>

LAMPIRAN

Lampiran 1 Keterangan Pernyataan (item 1-20):

No.	Keterangan Pernyataan	No.	Keterangan Pernyataan
Q1	Aplikasi ini menyenangkan dan menghibur untuk digunakan	Q11	Desain layar aplikasi ini berkualitas tinggi
Q2	Aplikasi ini menarik dan membuat saya ingin menggunakannya	Q12	Aplikasi ini memiliki desain visual yang konsisten
Q3	Aplikasi ini dapat disesuaikan dengan kebutuhan saya	Q13	Informasi dalam aplikasi ini ditulis dengan baik dan relevan dengan tujuan aplikasi
Q4	Aplikasi ini memungkinkan saya melakukan hal-hal sesuai dengan yang saya inginkan	Q14	Informasi yang terdapat dalam aplikasi ini akurat dan benar
Q5	Saya kemungkinan akan berbagi informasi dari aplikasi ini kepada orang lain	Q15	Aplikasi ini berasal dari sumber yang dapat dipercaya
Q6	Aplikasi ini berjalan dengan baik dan tanpa masalah teknis	Q16	Informasi dalam aplikasi ini ditampilkan dengan jelas
Q7	Aplikasi ini mudah dipelajari cara penggunaannya	Q17	Saya berpikir akan sering menggunakan aplikasi ini dalam 12 bulan ke depan
Q8	Saya dapat menjadi terampil menggunakan aplikasi ini dengan cepat	Q18	Saya akan merekomendasikan aplikasi ini kepada orang yang membutuhkannya
Q9	Aplikasi ini memiliki semua fitur yang Anda harapkan ada	Q19	Saya bersedia membayar untuk menggunakan aplikasi ini
Q10	Tampilan antarmuka aplikasi ini menarik secara visual	Q20	Penilaian bintang keseluruhan terhadap aplikasi ini

Lampiran 2 Rekapitulasi Skor 35 Responden Skala Likert 1-5:

No.	Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Total	Rata-rata
1	Q1	5	5	4	5	4	4	4	5	3	4	4	5	5	4	5	5	3	4	3	4	85	4.25
2	Q2	5	5	3	4	5	5	4	4	3	5	5	5	5	4	4	5	4	5	5	4	89	4.45
3	Q3	5	5	5	4	5	4	5	4	3	5	4	4	4	4	5	5	4	5	4	5	89	4.45
4	Q4	4	3	5	4	4	3	5	4	5	4	3	5	3	5	4	4	3	4	5	5	82	4.10
5	Q5	4	4	5	4	3	3	4	3	3	2	2	3	4	5	3	3	5	5	1	4	70	3.50
6	Q6	4	4	3	3	5	5	4	4	4	4	4	4	4	4	4	4	3	4	3	4	78	3.90
7	Q7	4	5	3	4	4	4	5	5	3	3	3	3	4	4	2	4	4	4	2	4	74	3.70
8	Q8	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	100	5.00
9	Q9	4	5	4	4	5	5	4	4	2	4	4	4	4	3	3	5	4	4	3	4	79	3.95
10	Q10	4	5	3	5	4	5	5	3	2	5	4	5	5	5	3	3	2	5	4	5	82	4.10
11	Q11	4	2	3	3	3	2	3	3	5	5	5	5	4	4	4	4	3	5	5	4	76	3.80
12	Q12	4	4	4	5	3	3	4	5	4	5	5	5	5	4	4	4	3	4	3	4	82	4.10
13	Q13	4	5	5	5	5	4	3	3	3	5	4	5	5	5	5	4	3	5	5	5	88	4.40
14	Q14	4	4	4	3	4	4	4	4	3	4	4	4	4	3	3	4	4	4	4	4	76	3.80
15	Q15	5	5	5	5	4	5	4	5	5	5	5	5	5	5	5	5	4	4	5	4	95	4.75
16	Q16	4	3	4	4	4	2	5	3	3	4	5	4	4	4	4	5	2	4	3	4	75	3.75
17	Q17	4	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4	2	4	90	4.50
18	Q18	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	3	5	3	5	96	4.80
19	Q19	2	4	4	3	3	2	4	4	3	4	4	4	4	4	3	4	1	3	1	3	64	3.20
20	Q20	4	4	3	3	3	3	4	3	4	4	4	4	4	5	4	5	2	3	2	4	72	3.60
21	Q21	3	3	4	5	3	5	5	5	5	3	5	5	5	5	5	4	3	3	2	5	83	4.15
22	Q22	3	3	3	4	5	5	3	4	4	3	4	4	4	3	4	4	4	4	4	5	77	3.85

No.	Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Total	Rata-rata
23	Q23	4	4	4	5	4	5	4	5	5	5	5	4	5	5	5	5	4	5	5	5	93	4.65
24	Q24	3	5	5	3	5	5	5	5	5	5	3	3	5	5	5	5	1	5	1	5	84	4.20
25	Q25	5	5	3	4	5	5	5	5	5	5	5	5	5	5	5	5	3	5	4	5	94	4.70
26	Q26	4	4	2	4	4	4	5	5	3	4	4	4	4	3	4	4	3	4	3	4	76	3.80
27	Q27	5	5	5	5	5	4	4	5	5	4	5	4	5	5	5	4	4	4	3	5	91	4.55
28	Q28	5	5	5	5	5	3	4	5	5	5	5	5	4	5	4	5	5	5	4	5	94	4.70
29	Q29	5	5	4	4	5	4	5	5	4	5	5	5	5	4	4	5	2	5	3	5	89	4.45
30	Q30	4	4	4	4	3	5	4	4	5	4	4	4	5	5	5	5	3	3	3	4	82	4.10
31	Q31	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	80	4.00
32	Q32	4	5	4	3	4	4	3	4	5	5	4	4	2	3	2	4	4	3	4	5	76	3.80
33	Q33	5	5	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	81	4.05
34	Q34	4	4	3	3	4	4	4	4	4	4	4	4	4	4	4	4	3	5	3	4	77	3.85
35	Q35	4	4	4	4	5	4	5	5	5	4	4	4	4	3	5	4	5	5	4	5	87	4.35