



# Universitas Nusantara PGRI Kediri

## UPT. PERPUSTAKAAN, PUBLIKASI DAN INOVASI

Alamat: Kampus 1, Jl. KH. Ahmad Dahlan No.76 Kota Kediri 64112  
Telp. (0354) 771576, (0354) 771503, (0354) 771495, Fax. (0354) 771576  
Website: <http://ppi.unpkediri.ac.id/> Email: [perpustakaan@unpkediri.ac.id](mailto:perpustakaan@unpkediri.ac.id)

### SURAT KETERANGAN BEBAS SIMILARITY

Ketua UPT Perpustakaan, Publikasi dan Inovasi Universitas Nusantara PGRI Kediri menerangkan bahwa mahasiswa dengan identitas berikut:

Nama Mahasiswa : Muhamad Fajar Sayuto  
NPM : 2213020045  
Program Studi : S1-Teknik Informatika

Judul Karya Ilmiah:

"IMPLEMENTASI SINGLE SHOT DETECTOR DAN MOBILENETV2 PADA SISTEM SORTASI TOMAT OTOMATIS DALAM EKOSISTEM IOT"

Dinyatakan sudah memenuhi syarat batas maksimal 30% *similarity* sesuai dengan ketentuan yang berlaku pada setiap subbab naskah Laporan **Tugas Akhir/Skripsi/Tesis** yang disusun.  
Demikian Surat Keterangan ini kami berikan untuk dapat dipergunakan seperlunya.

Kediri, 24 Juni 2026  
Ka UPT PPI,



Dr. Abdul Aziz Hunaifi, M.A



# Turnitin LLCC

## 2213020045\_Muhamad Fajar Sayuto

 TI-12

---

### Document Details

**Submission ID**

trn:oid:::2945:395328182

**Submission Date**

Jun 24, 2026, 2:13 PM GMT+7

**Download Date**

Jun 24, 2026, 2:16 PM GMT+7

**File Name**

2213020045\_Muhamad Fajar Sayuto.docx

**File Size**

3.1 MB

62 Pages

11,273 Words

74,581 Characters

# 12% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Filtered from the Report

- Bibliography

## Top Sources

- 7%  Internet sources
- 2%  Publications
- 10%  Submitted works (Student Papers)

## Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## Top Sources

- 7% Internet sources
- 2% Publications
- 10% Submitted works (Student Papers)

## Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

|    |                |                                                                 |     |
|----|----------------|-----------------------------------------------------------------|-----|
| 1  | Internet       | repository.unpkediri.ac.id                                      | <1% |
| 2  | Student papers | Institut Bisnis dan Teknologi Indonesia (INSTIKI) on 2025-05-30 | <1% |
| 3  | Student papers | Perpustakaan on 2025-08-06                                      | <1% |
| 4  | Student papers | UPN Veteran Yogyakarta on 2026-04-30                            | <1% |
| 5  | Internet       | id.123dok.com                                                   | <1% |
| 6  | Internet       | repository.umsu.ac.id                                           | <1% |
| 7  | Student papers | Universitas Tarumanagara on 2025-12-19                          | <1% |
| 8  | Student papers | Universitas Pertamina on 2026-01-19                             | <1% |
| 9  | Student papers | Universitas Tarumanagara on 2025-12-23                          | <1% |
| 10 | Internet       | e-jurnal.lppmunsera.org                                         | <1% |
| 11 | Internet       | jurnal.stkipppgritulungagung.ac.id                              | <1% |

|    |                |                                                                 |     |
|----|----------------|-----------------------------------------------------------------|-----|
| 12 | Internet       | www.unboxparadigm.com                                           | <1% |
| 13 | Student papers | UPN Veteran Yogyakarta on 2026-05-29                            | <1% |
| 14 | Internet       | www.scribd.com                                                  | <1% |
| 15 | Student papers | Universitas Muhammadiyah Sukabumi on 2025-07-16                 | <1% |
| 16 | Internet       | ejournal.uniks.ac.id                                            | <1% |
| 17 | Internet       | vdocuments.net                                                  | <1% |
| 18 | Internet       | www.jpiti.journals.id                                           | <1% |
| 19 | Internet       | www.pewarta88.com                                               | <1% |
| 20 | Student papers | Admin Turnitin UIIS on 2025-07-31                               | <1% |
| 21 | Student papers | STT PLN on 2024-08-06                                           | <1% |
| 22 | Student papers | National Institute Of Technology, Tiruchirappalli on 2023-09-17 | <1% |
| 23 | Internet       | digilib.uinsa.ac.id                                             | <1% |
| 24 | Internet       | ejournal.stmikelahma.ac.id                                      | <1% |
| 25 | Student papers | Universitas Gunadarma on 2025-09-25                             | <1% |

|    |                |                                                                                  |     |
|----|----------------|----------------------------------------------------------------------------------|-----|
| 26 | Student papers | Universitas Islam Riau on 2026-05-21                                             | <1% |
| 27 | Student papers | Universitas Muhammadiyah Purwokerto on 2025-07-10                                | <1% |
| 28 | Internet       | repo.undiksha.ac.id                                                              | <1% |
| 29 | Internet       | repository.ulb.ac.id                                                             | <1% |
| 30 | Internet       | doaj.org                                                                         | <1% |
| 31 | Internet       | repo.itera.ac.id                                                                 | <1% |
| 32 | Student papers | UPN Veteran Yogyakarta on 2026-05-05                                             | <1% |
| 33 | Student papers | Universitas Jember on 2025-12-08                                                 | <1% |
| 34 | Internet       | ejournal.uniramalang.ac.id                                                       | <1% |
| 35 | Internet       | eprints.utdi.ac.id                                                               | <1% |
| 36 | Publication    | Amanda Rizky Putri Wibowo, Savana Putra Aditama, Fathia Bilqish Wulandari, Mu... | <1% |
| 37 | Student papers | Telkom University on 2025-07-02                                                  | <1% |
| 38 | Student papers | UNIVERSITAS BUDI LUHUR on 2026-02-21                                             | <1% |
| 39 | Student papers | Universitas Muslim Indonesia on 2024-12-09                                       | <1% |

|    |                |                                                          |     |
|----|----------------|----------------------------------------------------------|-----|
| 40 | Student papers | Universitas Pertamina on 2025-07-21                      | <1% |
| 41 | Internet       | informatika.unpkediri.ac.id                              | <1% |
| 42 | Internet       | repository.uniramalang.ac.id                             | <1% |
| 43 | Internet       | text-id.123dok.com                                       | <1% |
| 44 | Internet       | vdokumen.com                                             | <1% |
| 45 | Student papers | Universitas Bangka Belitung on 2026-06-15                | <1% |
| 46 | Student papers | Universitas Dian Nuswantoro on 2015-06-28                | <1% |
| 47 | Student papers | Universitas Katolik Widya Mandala Surabaya on 2025-12-19 | <1% |
| 48 | Student papers | Universitas Muhammadiyah Purwokerto on 2025-08-15        | <1% |
| 49 | Student papers | Universitas Sumatera Utara on 2025-05-23                 | <1% |
| 50 | Internet       | api2017a001.wordpress.com                                | <1% |
| 51 | Student papers | poltekssn on 2025-07-09                                  | <1% |
| 52 | Internet       | repository.umy.ac.id                                     | <1% |
| 53 | Student papers | Fakultas Teknik on 2025-07-30                            | <1% |

|    |                |                                                    |     |
|----|----------------|----------------------------------------------------|-----|
| 54 | Student papers | Hankuk University of Foreign Studies on 2024-12-19 | <1% |
| 55 | Student papers | Institut Teknologi Sumatera on 2026-05-18          | <1% |
| 56 | Student papers | UM Surabaya on 2025-07-14                          | <1% |
| 57 | Student papers | UNIVERSITAS BUDI LUHUR on 2026-01-27               | <1% |
| 58 | Student papers | Universitas Brawijaya on 2019-07-04                | <1% |
| 59 | Student papers | Universitas Khairun on 2026-06-09                  | <1% |
| 60 | Student papers | Universitas Putera Batam on 2024-01-27             | <1% |
| 61 | Student papers | Universitas Tarumanagara on 2025-12-19             | <1% |
| 62 | Internet       | ojs.acad-pub.com                                   | <1% |
| 63 | Internet       | repository.unissula.ac.id                          | <1% |
| 64 | Student papers | Transylvania University on 2024-07-25              | <1% |
| 65 | Student papers | Universitas Muhammadiyah Purwokerto on 2025-05-27  | <1% |
| 66 | Student papers | Universitas Tarumanagara on 2025-12-18             | <1% |
| 67 | Internet       | docshare.tips                                      | <1% |

|    |                |                                                                                    |     |
|----|----------------|------------------------------------------------------------------------------------|-----|
| 68 | Internet       | ejournal.unib.ac.id                                                                | <1% |
| 69 | Internet       | etheses.uin-malang.ac.id                                                           | <1% |
| 70 | Internet       | github.com                                                                         | <1% |
| 71 | Internet       | ojs.stmikdharmapalariau.ac.id                                                      | <1% |
| 72 | Internet       | repository.amikom.ac.id                                                            | <1% |
| 73 | Internet       | repository.ub.ac.id                                                                | <1% |
| 74 | Student papers | Bellevue Public School on 2022-03-11                                               | <1% |
| 75 | Student papers | Binus University International on 2019-06-28                                       | <1% |
| 76 | Student papers | Institut Bisnis dan Teknologi Indonesia (INSTIKI) on 2025-05-30                    | <1% |
| 77 | Student papers | STT PLN on 2024-08-08                                                              | <1% |
| 78 | Publication    | Sofyh Zacksavira, Cindy Anggraeni Br Sebayang, Jonathan Imanuel, Grestia Br Sib... | <1% |
| 79 | Student papers | Sriwijaya University on 2019-07-01                                                 | <1% |
| 80 | Student papers | Sriwijaya University on 2020-03-05                                                 | <1% |
| 81 | Student papers | UNIVERSITAS BUDI LUHUR on 2025-07-21                                               | <1% |

|    |                |                                                  |     |
|----|----------------|--------------------------------------------------|-----|
| 82 | Student papers | Udayana University on 2024-12-02                 | <1% |
| 83 | Student papers | Universitas Hayam Wuruk Perbanas on 2026-06-11   | <1% |
| 84 | Student papers | Universitas Islam Riau on 2026-04-30             | <1% |
| 85 | Student papers | Universitas Khairun on 2026-05-11                | <1% |
| 86 | Student papers | Universitas Muhammadiyah Palembang on 2026-06-10 | <1% |
| 87 | Student papers | Universitas Muhammadiyah Sukabumi on 2026-01-13  | <1% |
| 88 | Student papers | Universitas Muhammadiyah Sukabumi on 2026-01-19  | <1% |
| 89 | Student papers | Universitas Sumatera Utara on 2026-06-04         | <1% |
| 90 | Internet       | doku.pub                                         | <1% |
| 91 | Internet       | eprints.uns.ac.id                                | <1% |
| 92 | Internet       | journal.trunojoyo.ac.id                          | <1% |
| 93 | Internet       | repo.unikadelasalle.ac.id                        | <1% |
| 94 | Internet       | widuri.raharja.info                              | <1% |
| 95 | Internet       | www.medcom.id                                    | <1% |

|     |                |                                                                            |     |
|-----|----------------|----------------------------------------------------------------------------|-----|
| 96  | Student papers | Udayana University on 2021-01-19                                           | <1% |
| 97  | Student papers | Universitas Brawijaya on 2017-07-29                                        | <1% |
| 98  | Student papers | Universitas Islam Riau on 2026-04-28                                       | <1% |
| 99  | Student papers | Universitas Negeri Surabaya The State University of Surabaya on 2025-10-20 | <1% |
| 100 | Student papers | Universitas Tarumanagara on 2025-12-12                                     | <1% |
| 101 | Student papers | Perpustakaan on 2025-08-06                                                 | <1% |
| 102 | Student papers | UIN Syarif Hidayatullah Jakarta on 2023-06-05                              | <1% |
| 103 | Student papers | UPN Veteran Yogyakarta on 2026-05-04                                       | <1% |
| 104 | Student papers | Universitas Maritim Raja Ali Haji on 2025-04-21                            | <1% |
| 105 | Student papers | Universitas Muslim Indonesia on 2025-07-16                                 | <1% |
| 106 | Internet       | jurnal.unprimdn.ac.id                                                      | <1% |

## BAB I

### PENDAHULUAN

#### A. Latar Belakang

Pertanian memiliki peranan penting dalam menjaga ketahanan pangan global (Diana Syahputri dkk., 2023). Di era saat ini, pertanian modern seperti hortikultura menjadi sektor penting, terutama pada bagian buah tomat (Septiadi & Mundiya, 2021). Pada tahun 2023 produksi buah tomat mencapai 1,2 juta ton (BPS 2023). Dengan adanya kondisi tersebut, perlu diadakannya peningkatan efisiensi pasokan pangan dan penerapan standar mutu pada tahap setelah panen agar tetap menjaga kualitas dan meningkatkan daya saing produk (Meilina Sari & Ansiska, 2024).

Tahapan setelah panen, terutama waktu pengelompokan kualitas (*Grading*) merupakan proses yang penting karena memengaruhi nilai jual tomat (Hasriani & Arwati, 2022). Hingga saat ini proses *grading* masih banyak dilakukan secara manual oleh pekerja (S. Santosa dkk., 2024). Proses sortir secara manual memiliki kekurangan, seperti ketidakkonsistenan hasil yang disebabkan oleh subjektivitas penilaian dan faktor kelelahan para pekerja (Sihombing, 2024). Hal ini menyebabkan produk yang beredar di pasaran memiliki kualitas yang tidak sama (Andina, 2025).

Sistem sortir otomatis menjadi solusi yang paling relevan untuk digunakan di era sekarang. Model *CNN* terbukti mampu mengekstraksi fitur kompleks dari citra buah dengan akurasi tinggi (Rahma Della & Darnita, 2024). Selain itu, pemakaian metode deteksi objek seperti *YOLOv8* mampu mencapai akurasi 96,6% (Isdi & Harmadi, 2024). Algoritma *YOLOv9c* memiliki akurasi dan efisiensi lebih baik dari versi sebelumnya (Fatahna dkk., 2025). Meskipun akurasinya tinggi, komputasi *YOLOv9c* cukup tinggi sehingga kurang optimal untuk sistem *IoT* berbasis konveyor (Prasetyo dkk., 2024).

Menanggapi terkait kekurangan pada penelitian sebelumnya, kombinasi antara *Single Shot Detector (SSD)* dan *MobileNetV2* diusulkan sebagai solusi

yang cukup efektif (Isdi & Harmadi, 2024). *SSD* merupakan metode deteksi objek dengan komputasi lebih ringan dan memiliki akurasi tinggi (Rahma Della & Darnita, 2024). Sedangkan *MobileNetV2* menjadi *backbone* yang ringan agar proses komputasi menjadi lebih cepat tanpa mengorbankan akurasi (Alfaruq dkk., 2023).

Dengan menggabungkan *computer vision* dan *deep learning* melalui *SSD* dan *MobileNetV2*, sistem dapat bekerja dengan cepat dan memiliki akurasi yang tinggi. Selain itu pendekatan ini juga membantu proses penerapan pertanian cerdas (*Smart Agriculture*) di Indonesia. Dengan demikian, penelitian ini mampu membantu peningkatan daya saing produk hortikultura nasional di pasar global (Isra Humaira B dkk., 2021).

## B. Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan, berikut ini beberapa permasalahan yang dapat diidentifikasi:

1. Proses penyortiran masih banyak dilakukan secara manual, hingga menyebabkan hasil sortir mengalami ketidakkonsistenan karena pengaruh subjektivitas penilaian dari tenaga kerja.
2. Teknologi otomatisasi yang telah dikembangkan sebelumnya masih banyak mengalami kendala, terutama dalam menjaga keseimbangan antara kecepatan deteksi dan akurasi yang tinggi pada penerapan secara *real-time*.
3. Belum adanya implementasi sistem sortir secara otomatis dengan mengombinasikan *SSS-MobileNetV2* pada buah tomat berbasis *IoT* dengan conveyor Pemilah dengan akurasi tinggi.

## C. Rumusan Masalah

Berdasarkan identifikasi masalah yang telah dipaparkan, berikut ini beberapa rumusan masalah pada penelitian ini:

1. Bagaimana Merancang sebuah sistem penyortiran otomatis untuk klasifikasi kematangan buah tomat yang efektif dalam segi kecepatan

komputasi dengan akurasi yang tinggi menggunakan kombinasi antara *SSD* dan *MobileNetV2* yang akan diterapkan pada *IoT* konveyor Pemilah secara *real-time*?

2. Bagaimana kombinasi *SSD-MobileNetV2* mampu mencapai keseimbangan antara kecepatan deteksi dan akurasi yang tinggi dalam pengolahan citra buah tomat?
3. Bagaimana mengimplementasikan sistem sortir buah tomat otomatis dengan akurasi tinggi berbasis *SSD-MobileNetV2* yang diintegrasikan pada konveyor Pemilah secara *real-time*?

#### D. Batasan Masalah

Agar penelitian lebih terarah dan fokus, terdapat beberapa batasan yang ditetapkan, diantaranya:

1. Objek penelitian: Penelitian ini berfokus pada buah tomat lokal yang diberi label menjadi 2 kelas, yaitu *fresh* (matang, bagus dan tanpa cacat) dan *rotten* (matang atau mentah dengan cacat) .
2. Arsitektur model: penelitian ini menggunakan kombinasi antara deteksi objek dan klasifikasi objek, menggunakan *Single Shot Detection (SSD)* dengan arsitektur *MobileNetV2*.
3. Perangkat keras: untuk *microcontroller* menggunakan ESP-32, dengan aktuator servo serta menggunakan konveyor Pemilah. Kamera yang digunakan *webcam eksternal* EYD dengan resolusi 1080P. *Background* foto yang digunakan untuk mengambil gambar kertas A4 warna putih. *Device* yang digunakan minimal memiliki spesifikasi *Ryzen 5 pro 3500u* dengan GPU *Radeon vega 8*. Ram 16 GB dengan SSD 512 GB.
4. Bahasa pemrograman yang digunakan *python* versi 3.10 dengan *library streamlit, PyTorch, OpenCV* sebagai penunjang penelitian.
5. Sistem hanya menghasilkan perhitungan jumlah buah tomat dan pengelompokan kualitas berdasarkan kriteria yang telah ditentukan peneliti.

6. Menggunakan metrik konvolusional untuk evaluasi, seperti akurasi, presisi, recall dan F1-score.
7. Proses *modelling* dilakukan menggunakan *tools google collab*, sedangkan untuk visualisasi menggunakan *VScode*.
8. Implementasi sistem dilakukan dalam bentuk simulasi prototipe pada lingkungan terkontrol.

#### E. Tujuan Penelitian

Dengan rumusan masalah yang telah dipaparkan, ditetapkan beberapa tujuan utama penelitian ini sebagai berikut:

1. Merancang sebuah sistem sortir otomatis untuk klasifikasi kematangan buah tomat yang memiliki kecepatan komputasi dan akurasi tinggi menggunakan arsitektur *SSD-MobileNetV2*.
2. Menguji penggunaan model *SSD-MobileNetV2* dengan *convolutional matrix* untuk mengetahui kelayakan sistem dalam segi kecepatan deteksi dan akurasi tinggi yang akan diimplementasikan pada industri secara *real-time*.
3. Mengimplementasikan sebuah sistem sortir otomatis untuk klasifikasi kematangan buah tomat dengan model *SSD-MobileNetV2* pada konveyor Pemilah berbasis *IoT* secara *real-time*.

#### F. Manfaat Penelitian

Adapun penelitian ini memberikan keuntungan serta kontribusi yang baik bagi berbagai pihak, seperti:

1. Bagi industri pertanian, diharapkan bisa memberikan kemudahan dalam melakukan penyortiran buah tomat, sehingga hasil yang didapatkan akan konsisten serta hemat biaya dan waktu.
2. Bagi akademisi, penelitian ini bisa dijadikan sebagai referensi untuk mengembangkan penelitian tentang pengolahan citra, terutama buah tomat.

## BAB II

### LANDASAN TEORI

#### A. Teori dan Penelitian Terdahulu

##### 1. Landasan Teori

###### a. Buah Tomat (*Solanum lycopersicum*)

Tomat merupakan bagian komoditas hortikultura yang strategis di Indonesia. Tomat cukup mudah untuk dibudidayakan, selain itu kandungan gizi yang terdapat didalamnya bisa dibilang cukup tinggi, seperti *likopen*, vitamin C dan *beta-karoten* yang memiliki manfaat baik bagi Kesehatan tubuh (Ahmed dkk., 2023).

Kualitas buah tomat dipengaruhi oleh beberapa hal, seperti kondisi lingkungan dan Teknik budidaya yang benar. Kondisi suhu, intensitas Cahaya, maupun pemberian nutrisi menjadi faktor utama (Susanti, 2024). Penggunaan pupuk  $FeSO_4$  dengan konsentrasi 50 mg/L memberikan peningkatan kualitas pada buah tomat dan membuat hasil produksi meningkat (Sitompul dkk., 2023).

###### b. Penilaian Mutu dan *Grading* Buah Tomat



Gambar 2. 1 Foto Buah Tomat

Penentuan mutu buah tomat dilakukan dengan metode penelitian sistematis berdasarkan beberapa faktor, seperti Tingkat kematangan, ukuran dan warna kulit. Selain itu, kualitas buah tomat dipengaruhi oleh lingkungan tumbuh dan perlakuan pascapanen (Sumiasih dkk., 2023). Proses *grading* dilakukan dengan pengamatan terhadap karakteristik warna pada buah tomat dengan mempertimbangkan tekstur permukaan buah (Humaira dkk., 2021). Penentuan mutu dan

*grading* berperan penting dalam menjaga konsistensi kualitas buah tomat yang beredar di pasaran. Di era pertanian modern ini proses *grading* bisa dilakukan dengan teknologi digital seperti pengolahan citra (Nuraisyah dkk., 2023).

### c. *Preprocessing*

Langkah pertama dalam pemrosesan gambar disebut *preprocessing* yang bertujuan untuk memodifikasi data *input* agar sesuai dengan persyaratan model *deep learning*. *Preprocessing* membantu model mengekstrak fitur dengan efisiensi maksimal, meningkatkan konsistensi data, dan mempercepat perhitungan (Siino dkk., 2024). Pada studi ini menggunakan tahapan *preprocessing* normalisasi dan *resize* sebagai berikut:

#### 1) Pemotongan Citra (*Crop ROI*)

*Crop ROI (Region of Interest)* berfungsi untuk memotong citra *input* sesuai dengan ukuran pada *bounding box*, sehingga citra *input* akan menjadi lebih fokus pada tomat ketika nanti dilakukan *resize* (Hwang dkk., 2022).

#### 2) *Resize*

*Resize* dilakukan untuk menyesuaikan ukuran gambar dengan standar dari *inputan* MobileNetV2 yaitu 224 X 224. Penetapan resolusi ini mengacu pada arsitektur asli MobileNetV2 yang dipublikasikan oleh (Sandler dkk., 2019) dalam prosiding *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Untuk menjaga informasi warna, prosedur ini dilakukan tanpa mengubah jumlah saluran warna pada gambar. Faktor skala (*scaling factor*) yang ditentukan dengan membagi dimensi gambar target dengan dimensi gambar asli digunakan untuk mengubah ukuran gambar (Das dkk., 2021). Rumus *resize* dinyatakan sebagai berikut:

$$sf = \frac{P_{tujuan}}{P_{awal}} \dots\dots\dots (2.1)$$

Koordinat *piksel* setelah *resize* dihitung menggunakan persamaan berikut:

$$\begin{aligned} x^i &= x \times sf \\ y^i &= y \times sf \end{aligned} \quad \text{.....} \quad (2.2)$$

Keterangan:

- a)  $sf$  : *scaling factor resize*
- b)  $P_{tujuan}$ : ukuran piksel citra tujuan
- c)  $P_{awal}$  : ukuran piksel citra awal
- d)  $x, y$  : koordinat piksel pada citra awal
- e)  $x^i, y^i$  : koordinat piksel pada citra hasil *resize*

### 3) Konversi Citra Digital (to RGB)

Data citra digital yang ditangkap oleh sensor (*webcam*) direpresentasikan secara komputasional sebagai matriks tiga dimensi, di mana  $C$  (*channels*) dalam implementasi menggunakan pustaka OpenCV, kanal warna pada dimensi ketiga disusun dalam urutan *BGR* (*Blue-Green-Red*), yang bertolak belakang dengan arsitektur *MobileNetV2* yang mensyaratkan input *RGB* (*Red-Green-Blue*). Dengan ketidaksesuaian ini perlu dilakukan konversi warna untuk mencegah interpretasi fitur (Massie dkk., 2024).

### 4) Normalisasi

Normalisasi ialah proses pengurangan rentang nilai piksel ke skala yang lebih stabil dan lebih kecil. Tujuan tahap ini adalah untuk menghindari nilai numerik yang terlalu besar, yang dapat menghambat proses pelatihan dan mengganggu stabilitas pembaruan bobot jaringan saraf. Pada penelitian ini dilakukan normalisasi piksel rentang asli  $[0.255]$  ke rentang  $[0.1]$  (Hutagaol dkk., 2025). Proses normalisasi dilakukan menggunakan rumus berikut:

$$I_{norm} = \frac{I_{pixel}}{255} \dots\dots\dots (2.3)$$

Keterangan:

- a)  $I_{norm}$  : nilai *piksel* setelah normalisasi
- b)  $I_{pixel}$  : nilai *piksel* sebelum normalisasi
- c) 255 : nilai maksimum intensitas *piksel* citra RGB

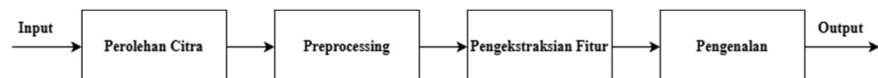
#### d. Klasifikasi Citra

Klasifikasi citra ialah metode pengelompokan gambar ke dalam beberapa kelas atau kategori. Tahapan-tahapan selama klasifikasi ialah *input* gambar, *preprocessing* gambar, membagi data menjadi dua bagian, yaitu data latih dan uji, kemudian dilakukan evaluasi untuk mengukur seberapa baik model bekerja (Ambarwari dkk., 2023).

Biasanya proses klasifikasi diawali dengan ekstraksi fitur dari citra *input*, setiap komponen seperti warna, pola dan tekstur diubah ke numerik agar bisa diolah model. Fitur-fitur tadi dikelompokkan ke berbagai kelas atau kategori sesuai karakteristiknya (Alzahrani, 2025).

#### e. Computer Vision

Pengolahan citra merupakan sebuah tahapan awal dari *computer vision* yang merupakan bagian dari kecerdasan buatan (*AI*). *Computer vision* membuat mesin bisa memahami informasi visual, seperti citra gambar atau video. Cara kerja teknologi ini seperti penglihatan manusia dalam mengenali pola, warna, bentuk, dan tekstur (Siwilopo & Marcos, 2023).

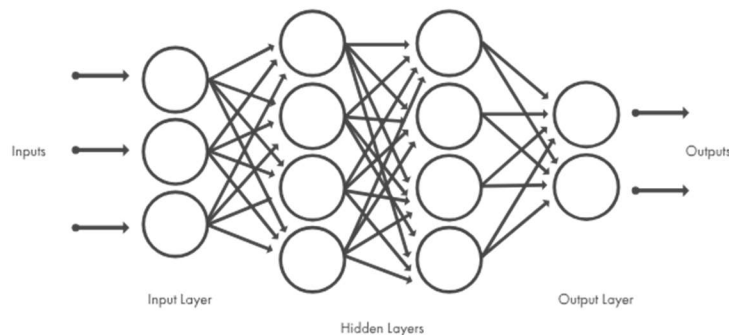


**Gambar 2. 2 Tahapan Computer Vision**

Citra digital dapat diartikan sebagai fungsi dua dimensi yang mencerminkan distribusi intensitas cahaya pada bidang koordinat (x, y), di mana setiap koordinat menyimpan nilai kecerahan dan piksel tertentu (Humaira dkk., 2021). Penerapan *computer vision* melalui

beberapa proses, seperti dengan melakukan *image acquisition* menggunakan kamera RGB, setelah itu masuk ke tahap *image processing* untuk memperbaiki kualitas gambar, kemudian proses *image enhancement* untuk mempertajam kualitas gambar, dan setelah itu masuk proses *image segmentation* untuk memisahkan antara objek dengan latar belakang seperti diagram pada gambar 2.2 (Robianto dkk., 2021).

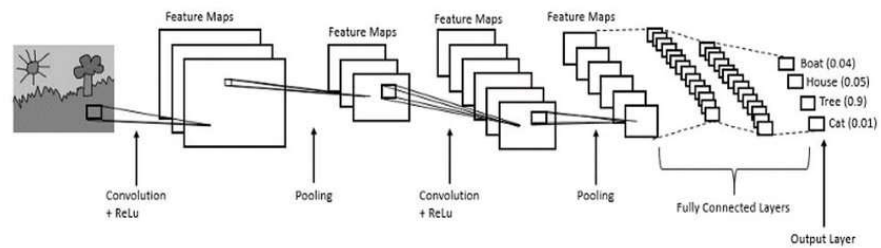
f. **Deep Learning dan CNN**



**Gambar 2.3 Deep Learning**

*Deep Learning* ialah bagian dari *Machine Learning* yang menekankan pada penggunaan jaringan saraf (*Neural Network*) yang memiliki banyak lapisan (*deep*) untuk merepresentasikan data seperti pada gambar 2.3. *Deep Learning* telah menjadi metode unggulan dalam pengolahan citra seperti klasifikasi, deteksi dan segmentasi objek, karena kemampuannya yang mampu belajar secara langsung dari data yang kompleks (Alzubaidi dkk., 2021).

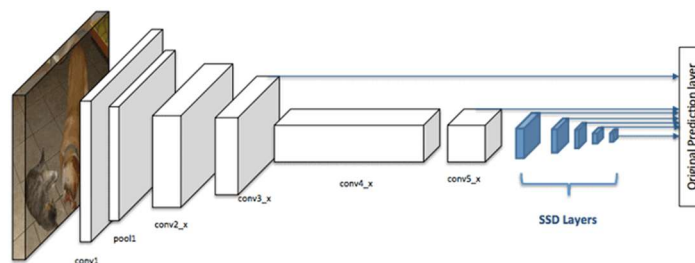
Salah satu arsitektur dari *Deep Learning* ialah *Convolutional Neural Network (CNN)*. Di mana *CNN* dirancang khusus untuk Teknik pengolahan citra dengan menggunakan mekanisme ekstraksi fitur seperti sudut, tepi dan tekstur. Di mana komponen-komponen tersebut merupakan faktor yang sangat dibutuhkan untuk proses klasifikasi (Purwono dkk., 2022).



**Gambar 2. 4 Arsitektur CNN**

Pada Umumnya *CNN* memiliki beberapa struktur seperti gambar 2.4, untuk strukturnya terdiri dari lapisan konvolusi (*convolutional layer*) untuk melihat bagian kecil citra (memerlukan filter/kernel), lapisan *pooling* (*downsampling*) berguna untuk menyesuaikan ukuran dimensi tanpa mengorbankan fitur penting, dan lapisan *fully-connected* yang menggabungkan fitur menjadi *output*. Oleh karena itu, *CNN* memungkinkan proses ekstraksi fitur dan klasifikasi citra secara efisien dan otomatis (Purwono dkk., 2022).

g. **Single Shot Detector (SSD)**



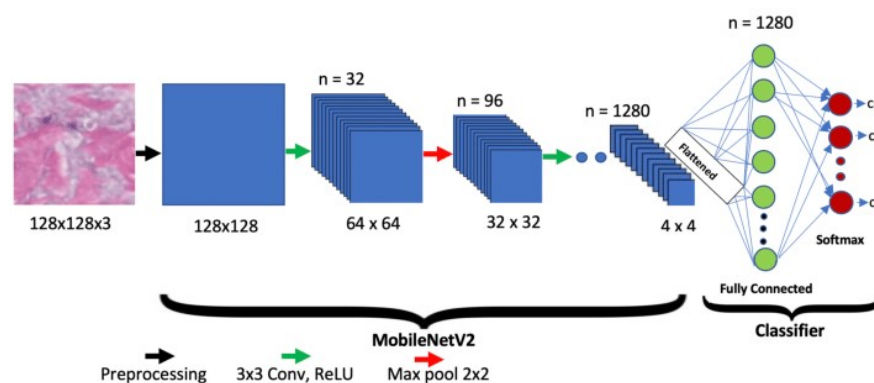
**Gambar 2. 5 Arsitektur SSD**

*SSD* adalah model deteksi objek *one-stage* yang memungkinkan pengenalan objek disertai penentuan Lokasi objek yang cepat dan efisien dalam satu jaringan (*forward-pass*) seperti pada gambar 2.5. Penerapan *SSD* dilakukan dengan menggabungkan klasifikasi objek dan prediksi kotak batas (*bounding boxes*) ke dalam satu jaringan konvolusi. Studi ini menggunakan beberapa *feature map* dengan skala yang berbeda untuk mendeteksi objek dari ukuran besar hingga kecil, serta kotak *default* (*default boxes*) dengan rasio aspek yang beragam

untuk mengidentifikasi bentuk objek yang berbeda (Nurchayyo & Iqbal, 2022).

Dikarenakan proses deteksi dilakukan langsung dalam satu tahap, *SSD* memiliki kecepatan lebih unggul dan cocok untuk aplikasi *real-time*. Namun, Ketika menangani objek yang kecil atau berskala sangat berbeda dalam satu citra dapat menyebabkan kehilangan detail objek kecil, hal ini karena skala fitur yang menurun lewat jaringan konvolusi (Guo dkk., 2021).

#### h. *MobileNetV2*



**Gambar 2. 6** Arsitektur *MobileNetV2*

MobileNetV2 ialah arsitektur *Convolutional Neural Network* (CNN) yang dirancang untuk mengekstrak fitur visual secara efisien dari komputer. Model ini dikembangkan oleh Google sebagai penerus generasi pertama yaitu *MobileNet*, pengembangan ini dilakukan untuk mengurangi kebutuhan komputasi dan memori sehingga dapat diimplementasikan pada perangkat dengan konsumsi daya rendah, seperti *Internet of Things*. Hal ini terlihat dari penggunaan *depthwise separable convolution*, sebuah teknik yang membagi proses konvolusi menjadi dua tahap yaitu *depthwise convolution* dan *pointwise convolution*. Pendekatan ini bertujuan untuk menyesuaikan jumlah parameter dan mempercepat proses komputasi tanpa secara signifikan menurunkan akurasi (R. R. Santosa dkk., 2025). Rumus *depthwise*

*convolution* (Persamaan 2.4), *pointwise convolution* (Persamaan 2.5) dan aktivasi *ReLU6* (Persamaan 2.6) sebagai berikut.

$$Y_{dw}(i, j, c) = \sum_{m=1}^k \sum_{n=1}^k X(i+n, j+n, c) \cdot K_{dw}(m, n, c) \quad \dots \quad (2.4)$$

Keterangan:

- a)  $Y_{dw}(i, j, c)$  : *output depthwise convolution*
- b)  $X(i, j, c)$  : nilai fitur *input* pada posisi  $(i, j)$  dan Channel  $c$
- c)  $K_{dw}(m, n, c)$  : *kernel depthwise* pada channel  $c$
- d)  $k$  : ukuran *kernel*
- e)  $c$  : indeks Channel

*Pointwise convolution* menggunakan kernel  $1 \times 1$  untuk menghasilkan fitur yang lebih representatif.

$$Y_{pw}(i, j, f) = \sum_{c=1}^C Y_{dw}(i, j, c) \cdot K_{pw}(c, f) \quad \dots \quad (2.5)$$

Keterangan:

- a)  $Y_{pw}(i, j, f)$  : *output pointwise convolution*
- b)  $Y_{dw}(i, j, c)$  : *output depthwise convolution*
- c)  $K_{pw}(c, f)$  : *kernel pointwise*
- d)  $C$  : jumlah Channel *input*
- e)  $f$  : indeks fitur *output*

*MobileNetV2* membatasi nilai aktivasi maksimum 6, hal ini bertujuan untuk menjaga presisi numerik ketika dijalankan pada perangkat dengan komputasi rendah (Parsaulian dkk., 2024).

$$X_{act} = \min(\max(0, x), 6) \quad \dots \quad (2.6)$$

Keterangan:

- a)  $x$  : nilai *input* hasil konvolusi
- b)  $X_{act}$  : nilai output setelah aktivasi (rentang 0 hingga 6)

Selain itu, *MobileNetV2* memperkenalkan dua inovasi penting: *linear bottleneck* dan *inverted residuals*. Ketika ukuran kanal input dan output sama, konsep *inverted residuals* memungkinkan jaringan memperkuat informasi penting melalui koneksi *shortcut*, sehingga gradien menjadi lebih stabil selama pelatihan. Di sisi lain, *linear bottleneck* digunakan untuk menentukan latensi informasi *non-linear* sebagai hasil dari transformasi *fitting* yang kompleks. Dibandingkan dengan arsitektur *CNN* tradisional yang memiliki parameter besar, kombinasi kedua komponen ini membuat *MobileNetV2* lebih efisien dalam mengekstraksi fitur resolusi tinggi dengan kecepatan (Yong dkk., 2023).

Fitur yang dihasilkan oleh *MobileNetV2* masih berupa *feature map* dengan dimensi spasial setelah melewati serangkaian blok konvolusi dan *inverted residuals*. Teknik *Global Average Pooling (GAP)* digunakan untuk mengompres informasi spasial ini menjadi representasi vektor yang kompak (Dogan, 2023). Rumus *GAP* dituliskan pada persamaan 2.7.

$$GAP_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W F_e(i, j) \dots\dots\dots (2.7)$$

Keterangan:

- a)  $GAP_c$  : Nilai rata-rata fitur pada Channel ke- $c$
- b)  $H, W$  : Tinggi dan lebar spasial dari *feature map*
- c)  $F_e(i, j)$  : Nilai piksel fitur pada posisi  $(i, j)$  di Channel ke- $c$

*Fully Connected Layer*, yang juga dikenal sebagai *Dense Layer*, menerima vektor fitur yang dihasilkan oleh *GAP*. Dengan memetakan fitur yang diekstraksi ke jumlah kelas target (*fresh/busuk*), lapisan ini bisa melakukan klasifikasi tingkat tinggi (Ameh Joseph dkk., 2022). Rumus *GAP* dituliskan sebagai berikut.

$$Z = W^T \cdot X + b \quad \dots\dots\dots (2.8)$$

**Keterangan:**

- a)  $Z$  : Vektor *output (logits)* dengan dimensi sejumlah kelas
- b)  $X$  : Vektor *input* (hasil dari *Global Average Pooling*)
- c)  $W$  : Matriks bobot yang menghubungkan neuron *input* ke neuron *output*
- d)  $b$  : Vektor bias

Setelah mendapat skor mentah (*logits*), dilanjutkan dengan klasifikasi dengan *softmax*. Vektor keluaran dinormalisasi oleh fungsi *Softmax* sehingga jumlah setiap elemennya sama dengan 1. Hal ini memungkinkan sistem untuk menentukan tingkat kepercayaan masing-masing kelas (Mehra dkk., 2023). Untuk rumus *softmax* dituliskan sebagai berikut.

$$P_i = \frac{e^{Z_i}}{\sum_{j=1}^K e^{Z_j}} \quad \dots\dots\dots (2.9)$$

**Keterangan:**

- a)  $P_i$  : Probabilitas prediksi untuk kelas ke- $i$
- b)  $Z_i$  : Skor (*logits*) kelas ke- $i$  dari *Dense Layer*
- c)  $K$  : Jumlah total kelas target
- d)  $e$  : Bilangan *Euler* (konstanta matematika  $\approx 2.718$ )

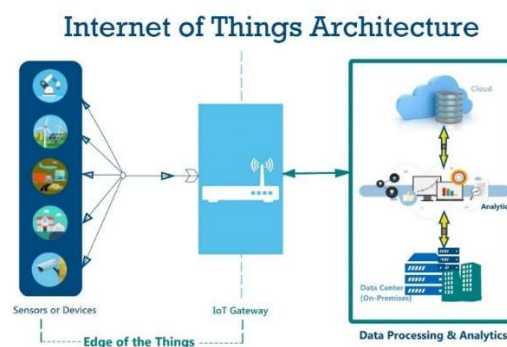
Untuk meminimalisir kesalahan klasifikasi di mana objek busuk dinyatakan *fresh* (*False Negative*) dilakukan penerapan *Confidence Thresholding* dengan nilai 80%, jika hasil klasifikasi *fresh* lebih dari nilai yang sudah ditentukan maka sistem akan menyatakan *fresh*, namun jika kurang atau sama dengan nilai *Threshold* maka akan dinyatakan busuk (Pérez-Roncal dkk., 2022).

$$C = \begin{cases} \text{Fresh,} & \text{Jika } P_{\text{fresh}} > \theta \\ \text{Busuk,} & \text{Jika } P_{\text{busuk}} \leq \theta \end{cases} \quad \dots\dots\dots (2.10)$$

Keterangan:

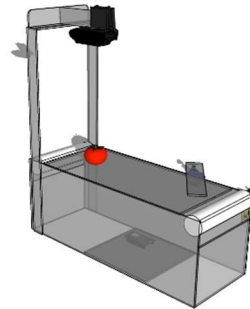
- a)  $C$  : Hasil Akhir
- b)  $P_{fresh}$  : Nilai probabilitas *fresh*
- c)  $P_{busuk}$  : Nilai probabilitas busuk
- d)  $\theta$  : Nilai *Threshold* (80%)

i. ***Internet of Things dan Conveyor otomatis***



**Gambar 2. 7 Arsitektur *IoT***

Sistem konveyor otomatis yang terintegrasi dengan teknologi *Internet of Things (IoT)* gambar 2.7 memungkinkan pemeliharaan *real-time* dan peningkatan proses, serta otomatisasi operasional yang sebelumnya bergantung kepada tenaga manusia. Dalam aplikasi pertanian, konveyor otomatis dapat memindahkan buah dari satu titik ke titik lain, sementara sensor dan modul *IoT* dapat memantau secara terus-menerus kondisi seperti jumlah buah, kecepatan, lingkungan sekitar, atau status antrean buah (Zain dkk., 2024). Sebagai contoh, penelitian berjudul "Rancang Bangun Sistem Pemantauan Berbasis *IoT* pada Sistem Konveyor Multi untuk Pemindahan Benda" (2024) menunjukkan bahwa sistem konveyor Multi yang dilengkapi dengan sensor dan mikrokontroler ESP32 memungkinkan operasi efisien dengan waktu respons rata-rata 449 ms, sehingga mengurangi potensi *bottleneck* di titik pemindahan (Erdani dkk., 2024).



**Gambar 2. 8 Prototipe Konveyor**

Penggunaan konveyor otomatis untuk penyortiran buah juga semakin didukung oleh teknologi seperti visi mesin (*machine vision*) dan berbagai sensor lainnya. Sebagai contoh, penelitian yang mengembangkan sistem konveyor otomatis dengan fitur warna, ukuran, dan bentuk buah melalui pemrosesan citra menunjukkan bahwa penyortiran dan pengklasifikasian dapat dilakukan secara otomatis dan lebih akurat dari pada dengan metode manual (Jangam dkk., 2023).

#### j. Evaluasi Kinerja Sistem

Evaluasi merupakan tahapan yang krusial karena dapat mengukur sejauh mana model dapat bekerja dengan akurat, sehingga hasil yang diberikan dapat dipertanggungjawabkan. *Confusion matrix* menjadi model evaluasi yang paling sering digunakan karena mampu mengevaluasi seluruh hasil prediksi terhadap data uji, baik yang benar maupun salah dengan detail (Kurniati, 2024).

*Confusion matrix* adalah jenis tabel yang menunjukkan seberapa akurat dan berguna prediksi untuk setiap kelas dalam model klasifikasi. Empat komponen utama dari matriks ini adalah *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. Dengan memahami semua komponen ini, peneliti dapat menentukan metrik evaluasi lainnya seperti akurasi, presisi, *recall*, dan F1-score dengan baik (Hardiyanto & Akbar, 2025).

### 1) Akurasi

Akurasi adalah metrik dasar yang digunakan untuk menghitung persentase prediksi benar dari total jumlah prediksi. Meskipun sering digunakan, akurasi saja tidak dapat secara akurat menilai kinerja model, terutama ketika data set tidak konsisten (Kurniati, 2024).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots\dots\dots (2.11)$$

### 2) Presisi

Presisi menghitung seberapa banyak prediksi positif yang benar. Nilai yang tinggi menunjukkan bahwa model lebih sering memberikan prediksi benar (*True Positive*) dari pada prediksi salah (*false positive*) (Hardiyanto & Akbar, 2025; Kurniati, 2024).

$$Precision = \frac{TP}{TP + FP} \dots\dots\dots (2.12)$$

### 3) Recall

*Recall* menghitung seberapa baik model dapat mengidentifikasi semua data positif yang diprediksi benar (*True Positive*) dalam data set. *Recall* yang tinggi menunjukkan bahwa model dapat mendeteksi data positif dengan baik (Hardiyanto & Akbar, 2025).

$$Recall = \frac{TP}{TP + FN} \dots\dots\dots (2.13)$$

### 4) F1-Score

F1-score adalah rata-rata harmonik antara *precision* dan *recall* yang memberikan gambaran yang jelas tentang kinerja model, terutama ketika terdapat ketidakseimbangan kelas dalam data set (Kurniati, 2024).

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \dots\dots\dots (2.14)$$

## 2. Kajian Pustaka

Berlandaskan tinjauan literatur untuk mengidentifikasi kesenjangan penelitian, beberapa artikel terdahulu yang relevan dipilih sebagai acuan pengembangan model deteksi kualitas tomat berbasis *computer vision*. Dalam jurnal berjudul "Deteksi Objek Manusia pada Citra Menggunakan *Single Shot Detector (SSD)* Berbasis *Edge Computing*," oleh Iqbal dkk. (2024) mengembangkan sistem deteksi objek manusia menggunakan *SSD* pada perangkat *edge ESP32-CAM* dan *MobileNetV2* sebagai klasifikasi citra. Hasil menunjukkan akurasi *mAP* mencapai 87,51% pada siang hari dan waktu respons rata-rata 2,843 detik. Hasil yang didapat kurang maksimal karena *SSD* kurang optimal untuk penggunaan Multi objek. Oleh karena itu, penelitian ini mengusulkan model *SSD-MobileNetV2* untuk deteksi buah tomat. Objek akan dideteksi secara individu sehingga bisa maksimal menggunakan mode *SSD*. Selain itu tipe citra yang digunakan juga memiliki perbedaan, di mana penelitian ini menggunakan latar belakang putih polos pada citra untuk memudahkan proses deteksi.

Penelitian Sun et al. (2025) dengan judul "*ToRLNet: A Lightweight Deep Learning Model for Tomato Detection and Quality Assessment Across Ripeness Stages*" mengembangkan model *ToRLNet* berbasis *YOLOv12n* untuk mendeteksi ambang batas kematangan tomat dalam kondisi lapangan yang kompleks. Hasil penelitian menunjukkan kinerja tinggi dengan *F1-score* sebesar 88,49% dan *mAP50* sebesar 91,76%. Dikarenakan komputasi dari *YOLOv12n* cukup tinggi, maka kurang sesuai jika diterapkan pada konveyor Pemilah. Oleh karena itu, Penelitian ini mengusulkan penggunaan *SSD-MobileNetV2* sebagai model utama. Dengan kombinasi metode tersebut sistem dapat lebih efisien dan akurat dengan daya komputasi yang cukup ringan.

Selanjutnya Pamungkas & Amrulloh (2025) melalui artikel berjudul “Analisis Hasil Klasifikasi Penyakit Daun Bawang Merah Menggunakan CNN Arsitektur *Xception*” memanfaatkan arsitektur *CNN Xception* untuk mengklasifikasikan penyakit pada daun bawang. Data set yang digunakan gambar daun bawang yang telah diproses dan di *augmentasi*, lalu menerapkan model *Xception* dengan *transfer learning*. Hasil akurasi mencapai 99,71% dan validasi 97,37%. Penelitian ini berfokus pada buah tomat dengan menggunakan arsitektur *MobileNetV2*. Selain itu, penelitian ini terintegrasi dengan *IoT* berbasis konveyor Pemilah yang dilengkapi dengan sistem deteksi menggunakan *SSD*.

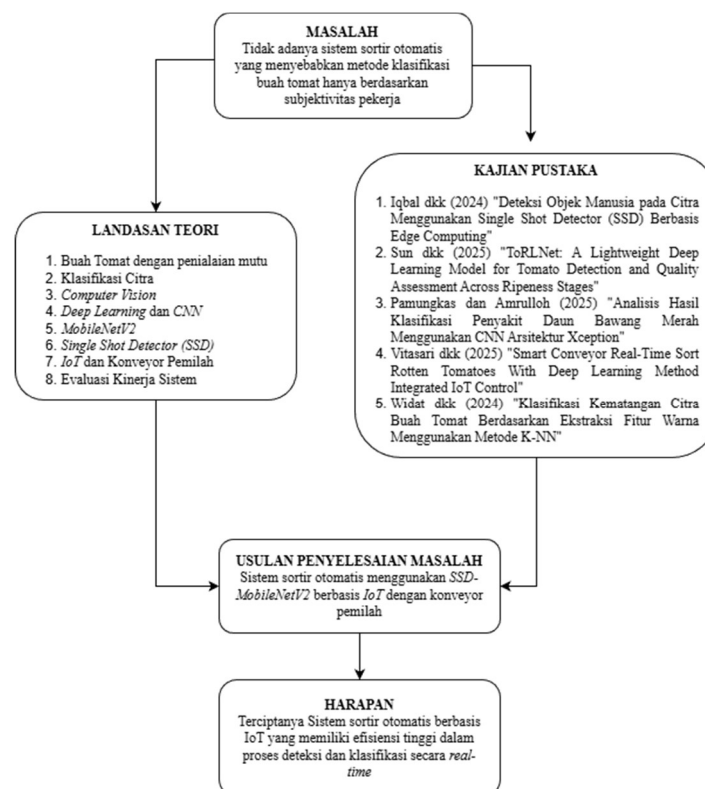
Adapun penelitian oleh Vitasari et al. (2025) yang berjudul “*Smart Conveyor Real-Time Sort Rotten Tomatoes With Deep Learning Method Integrated IoT Control*” mengembangkan sistem konveyor pintar untuk penyortiran tomat berdasarkan kategori kualitas menggunakan model *SSD MobileNetV2*. Sistem ini didukung dengan integrasi *IoT* melalui *Raspberry Pi* dan motor *servo* untuk penggerak konveyor, dengan akurasi deteksi mencapai 94%. Penelitian ini memiliki kedekatan dengan studi peneliti terutama dari aspek metode tersebut menggunakan *SSD MobileNetV2* dan konveyor. Perbedaan dengan penelitian ini terletak pada penggunaan *mikrokontroler*, di mana peneliti menggunakan *ESP32*. Selain itu, terdapat beberapa *scenario* yang ditetapkan dalam proses pengujian serta penggunaan metode evaluasi menggunakan *matrix* evaluasi yang digunakan untuk mengukur tingkat akurasi menggunakan *confusion matrix*. Perbedaan lainnya terdapat pada pemilihan data set, di mana penelitian ini menggunakan data set primer yang diberi label langsung oleh petani.

Selain itu Widat et al. (2024) dalam artikelnya yang berjudul “Klasifikasi Kematangan Citra Buah Tomat Berdasarkan Ekstraksi Fitur Warna Menggunakan Metode K-NN” melakukan penelitian untuk menentukan ambang batas kematangan tomat berdasarkan analisis warna citra. Metode yang digunakan adalah K-Nearest Neighbor (K-NN) dengan

ekstraksi RGB dan HSV, diterapkan pada 150 data tomat. Hasil penelitian menunjukkan bahwa nilai  $k = 5$  memberikan kinerja terbaik dengan akurasi 98,33%, ini berarti 59 dari 60 citra berhasil diklasifikasikan dengan benar. Perbedaan dengan penelitian ini yaitu pemilihan model, dimana penelitian ini menggunakan kombinasi *SSD-MobileNetV2* untuk melakukan proses deteksi dan klasifikasi, selain itu, penelitian ini menghasilkan prototipe konveyor pemilah dimana pada penelitian sebelumnya tidak melakukan hal tersebut.

## B. Kerangka Berpikir

Gambar 2.9 Merepresentasikan seluruh tahapan bagaimana penelitian ini berjalan, mulai dari identifikasi masalah, teori yang digunakan, penelitian terdahulu, Hingga solusi yang ditawarkan untuk permasalahan yang dihadapi, serta penulis juga menyertakan harapan dari penelitian ini.



Gambar 2. 9 Kerangka Berpikir

Gambar 2.9 Menjelaskan bahwa penelitian ini merupakan bentuk jawaban dari permasalahan terhadap ketiadaannya sistem sortir otomatis yang menyebabkan penilaian bergantung pada subjektivitas pekerja, solusi yang ditawarkan adalah sistem sortir otomatis berbasis *IoT* dengan menggunakan model *SSD-MobileNetV2*.

### 1. Data Input

Penelitian ini menggunakan beberapa jenis *inputan* data, jenis ini dibedakan berdasarkan kegunaannya masing-masing, untuk jenis-jenisnya ialah:

#### a. Data set Buah Tomat

Data set diperoleh menggunakan kamera *Handphone* yang diambil secara mandiri dan divalidasi oleh pakar (Petani Tomat), data set ini akan dilatih dengan model *deep learning* agar bisa mengklasifikasikan kualitas buah tomat ke dalam kategori *fresh* (warna cerah dan tekstur utuh) serta kategori busuk (cacat).

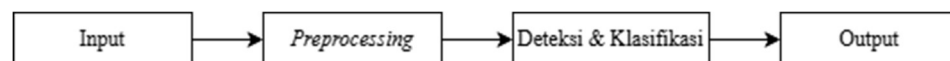
#### b. Fitur yang Diekstraksi Citra

Fitur yang diekstraksi meliputi warna (oren cerah hingga hijau), tekstur kulit (Mulus hingga cacat berlubang), serta bentuk dan ukuran buah tomat.

#### c. Data Pelatihan Model

Melakukan pelatihan model *SSD-MobileNetV2* untuk klasifikasi buah tomat di mana data latih berasal dari gambar-gambar buah tomat yang telah diberi label.

### 2. Gambaran Proses



**Gambar 2. 10 Gambaran Proses**

Berikut ini merupakan tahapan untuk proses mulai dari *input* sampai *output* sesuai dengan gambar 2.10 diantaranya adalah:

- a. Pengambilan *input* citra menggunakan kamera beresolusi 1080P yang diletakkan pada konveyor yang dilengkapi dengan pencahayaan lampu mini *LED* agar lebih seragam dalam pengambilan gambar.
- b. *Preprocessing* dilakukan untuk meningkatkan kualitas gambar, baik dari segi ukuran maupun *contras* sehingga dapat memudahkan model dalam melakukan klasifikasi.
- c. Proses deteksi menggunakan metode *Single Shot Detector (SSD)* dengan *backbone MobileNetV2* dengan hasil klasifikasi *fresh* dan busuk.
- d. Hasil akan dipilah sesuai dengan kategorinya menggunakan *servo* dan konveyor berjalan.

### 3. Data Output

Hasil yang didapatkan dari sistem sortir otomatis menggunakan model *SSD-MobileNetV2* yang diterapkan pada *prototype* konveyor Pemilah berbasis *IoT* adalah buah tomat yang disimpan ke dalam tempat penyimpanan sesuai dengan kategorinya, di mana jika tomat *fresh* *servo* akan bergerak dan mengarahkan ke wadah *fresh*, sedangkan jika tomat diklasifikasi busuk maka *servo* tidak bergerak dan membiarkan tomat jatuh di ujung konveyor ke dalam wadah dengan kategori busuk.

### BAB III

## METODE PENELITIAN

### A. Desain Penelitian

Desain penelitian digunakan untuk memberikan arah dan struktur yang jelas dalam pelaksanaan penelitian, sehingga setiap langkah dapat diselesaikan secara sistematis dan sesuai dengan tujuan yang telah ditentukan. Untuk komponen-komponen desain penelitian terdiri dari:

#### 1. Jenis Penelitian (*Research Type*)

Penelitian ini menggunakan metodologi Penelitian Pengembangan (*Development Research*) karena bertujuan untuk mengembangkan produk baru berupa sistem klasifikasi kematangan buah tomat berbasis *SSD-MobileNetV2* yang disimulasikan dalam lingkungan konveyor pintar. Desain penelitian ini dipilih karena fokus penelitian tidak hanya pada analisis, tetapi juga pada penciptaan, pengembangan, dan pengujian model deteksi yang dapat bekerja secara otomatis.

Sistem yang sedang dikembangkan dimaksudkan untuk memberikan solusi bagi masalah klasifikasi tomat yang saat ini masih banyak ditangani secara manual dan subjektif. Diharapkan penggunaan *SSD-MobileNetV2* akan memfasilitasi proses klasifikasi yang lebih efisien, konsisten, dan akurat, serta mendorong otomatisasi dalam industri pertanian melalui simulasi sistem konveyor.

#### 2. Variabel Penelitian

##### a. Variabel *Independent* (Bebas / *Prediktor*)

Variabel independen dalam penelitian ini meliputi penggunaan algoritma *SSD-MobileNetV2* sebagai model utama untuk mendeteksi dan mengklasifikasikan tingkat kematangan buah tomat. Variabel ini juga memengaruhi faktor teknis seperti resolusi citra, parameter pelatihan (ukuran *batch*, *epoch*, dan *learning rate*), serta jumlah data yang digunakan, karena semua faktor ini berpotensi memengaruhi kinerja model.

### b. Variabel *Dependent* (Terikat / *Respons*)

Variabel dependen yang diukur adalah kinerja sistem klasifikasi, yang dapat diamati melalui akurasi klasifikasi dan waktu deteksi. Sementara waktu deteksi menunjukkan jumlah waktu yang dibutuhkan sistem untuk menyelesaikan proses deteksi dan klasifikasi pada satu citra atau dalam alur simulasi konveyor pintar, akurasi klasifikasi menunjukkan persentase keberhasilan model dalam mengklasifikasikan tomat ke kategori kematangan yang sesuai. Kedua indikator ini digunakan untuk menentukan efektivitas model *SSD-MobileNetV2* dalam sistem klasifikasi otomatis berdasarkan simulasi.

### c. Variabel Kontrol (Pengendali)

Dalam penelitian ini, variabel kontrol mengukur berbagai kondisi untuk memastikan hasil yang konsisten. Spesifikasi perangkat keras, seperti *CPU*, *GPU*, dan *RAM*, tidak berubah selama pelatihan dan pengujian. Lingkungan pemrograman juga ditingkatkan dengan penggunaan *Python* dan *library* khusus seperti *TensorFlow* dan *OpenCV* yang digunakan secara konsisten sepanjang fase penelitian. Selain itu, data set citra tomat yang digunakan untuk analisis dan pengujian tidak diubah untuk mencegah variabilitas data, dan proses simulasi konveyor pintar yang berfungsi sebagai lingkungan pengujian dilakukan secara terstandarisasi untuk memastikan konsistensi hasil evaluasi.

## 3. Metode Pengumpulan Data

Pengumpulan data untuk penelitian ini dilakukan dengan metode dokumentasi langsung terhadap objek studi, yaitu tomat. Gambar tomat diperoleh menggunakan kamera digital dengan resolusi minimal 5 MP dengan sensor *RGB* untuk memastikan kualitas gambar yang jelas dan representatif. Pengambilan citra dilakukan dalam berbagai kondisi, termasuk perbedaan tingkat kematangan, ukuran, tekstur, dan warna kulit tomat, sehingga data set yang diperoleh dapat secara akurat mencerminkan kondisi tomat di lapangan. Data yang dikumpulkan kemudian digunakan

sebagai data pelatihan dan pengujian untuk model *SSD-MobileNetV2* guna melakukan deteksi dan klasifikasi objektif tingkat kematangan tomat. Selain itu, data ini juga digunakan untuk mengevaluasi kinerja sistem menggunakan metrik seperti akurasi, presisi, dan waktu deteksi dalam simulasi konveyor pintar yang digunakan dalam penelitian ini.

## B. Instrumen Penelitian

Penelitian ini melibatkan beberapa instrumen seperti perangkat keras, perangkat lunak, data set serta instrumen analisis yang setiap bagian memiliki peranan penting dalam proses pengumpulan data, pemrosesan data, pelatihan model dan implementasi pada konveyor Pemilah, serta membantu proses evaluasi kinerja sistem. Bagian tersebut meliputi:

### 1. Perangkat Keras (*Hardware*)

- a. Spesifikasi komputer yang digunakan dalam penelitian ini minimal memiliki *CPU Ryzen 5 Pro 3500u* dengan *GPU Radeon Vega 8* dan menggunakan *RAM 16 GB* serta *SSD 512 GB*.
- b. Penelitian ini menggunakan kamera *Smartphone* minimal *5 MP* sebagai pengambilan foto data set dan *webcam EYD 1080P* sebagai deteksi objek pada *IoT*.
- c. Pada penerapan prototipe *IoT* penelitian ini menggunakan *belt conveyor*, motor *DC*, akrilik transparan sebagai *frame*, *ESP-32* sebagai kontroler dan mini *led* berwarna putih sebagai pencahayaan pada *webcam*.

### 2. Perangkat Lunak (*Software*)

- a. Sistem operasi yang digunakan pada penelitian ini adalah *windows 11 pro* dengan versi 25H2.
- b. Bahasa pemrograman yang digunakan *python 3.10* yang digunakan untuk pemrosesan citra, pelatihan model, implementasi algoritma sampai proses evaluasi.
- c. *Library* atau *framework* yang digunakan meliputi *Tensorflow* untuk melatih dan mengimplementasikan model *SSD-MobileNetV2*,

*OpenCV* untuk pemrosesan citra serta *numpy*, *pandas* dan *seaborn* untuk pengolahan data dan visualisasi data.

- d. *IDE* atau *tools* yang digunakan antara lain *Google Colab*, *Jupyter Notebook* dan *VScode* untuk menulis kode program.

### 3. Data set

Data set buah tomat didapat secara langsung menggunakan kamera *Smartphone* dengan berbagai kondisi seperti tingkat kematangan, tekstur kulit, warna kulit dan ukuran. Format data set citra yang dihasilkan dari kamera meliputi *JPG*, *JPEG* dan *PNG* yang kemudian dibagi menjadi data *training* dan data *testing*.

### 4. Analisis Hasil

Pengujian non-fungsional menggunakan *confusion Matrix* agar dapat melihat tingkat kesalahan setiap kelas, selain itu pengukuran akurasi, *recall*, *presisi* dan *F1-Score* diperlukan untuk mengetahui tingkat keberhasilan model dalam melakukan klasifikasi.

## C. Tempat dan Jadwal Penelitian

Penelitian ini dilakukan di Desa Gebangkerep Kecamatan Baron Kabupaten Nganjuk. Pemilihan lokasi ini dinilai sesuai karena memiliki pasokan tomat yang melimpah sehingga memudahkan dalam proses pengumpulan data set. Rincian kegiatan yang dilakukan selama proses penelitian tertera pada tabel 3.1 di bawah ini.

**Tabel 3. 1 Jadwal Penelitian**

| No. | Uraian Kegiatan             | Bulan Ke-1 |   |   |   | Bulan Ke-2 |   |   |   | Bulan Ke-3 |   |   |   | Bulan Ke-4 |   |   |   | Bulan Ke-5 |   |   |   | Bulan Ke-6 |   |   |   |
|-----|-----------------------------|------------|---|---|---|------------|---|---|---|------------|---|---|---|------------|---|---|---|------------|---|---|---|------------|---|---|---|
|     |                             | 1          | 2 | 3 | 4 | 1          | 2 | 3 | 4 | 1          | 2 | 3 | 4 | 1          | 2 | 3 | 4 | 1          | 2 | 3 | 4 | 1          | 2 | 3 | 4 |
| 1.  | Studi Literatur             |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |
| 2.  | Pengumpulan Data            |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |
| 3.  | Analisis Sistem             |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |
| 4.  | Desain Sistem               |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |
| 5.  | Implementasi Model          |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |
| 6.  | Uji Coba dan Analisis hasil |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |
| 7.  | Penyusunan Laporan          |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |            |   |   |   |

## D. Objek Penelitian / Subjek Penelitian

### 1. Analisis Kebutuhan Sistem

Untuk memfasilitasi pengembangan sistem klasifikasi tomat berbasis *computer vision*, standarisasi persyaratan fungsional dan non-fungsional serta spesifikasi data yang telah ditetapkan menjadi bagian penting. Fitur dan persyaratan kinerja yang diperlukan untuk menjamin efektivitas operasional sistem telah dijabarkan secara rinci dalam deskripsi ini. Untuk mencapai akurasi deteksi terbaik, jumlah dan variasi data gambar juga telah ditentukan sesuai dengan prinsip-prinsip *Deep Learning*. Akibatnya, standar persyaratan ini berfungsi sebagai dasar yang andal dan konsisten untuk mencapai tujuan penelitian.

### 2. Objek Penelitian

Objek penelitian ini ialah sistem deteksi dan klasifikasi tomat yang dikembangkan menggunakan algoritma *SSD MobileNetV2*. Menggunakan gambar yang diambil secara *real-time* oleh *webcam* yang dipasang di jalur konveyor, sistem ini dirancang untuk mengenali dan mengklasifikasikan tomat. Konveyor penyortiran bertugas memindahkan tomat agar setiap objek dapat terdeteksi dengan benar, sementara sensor digunakan untuk memastikan tomat berada dalam bidang pandang kamera sebelum proses inferensi. Tujuan studi ini ialah untuk mengevaluasi kemampuan model dalam mengidentifikasi dan mengklasifikasikan tomat. Model ini menggunakan data set foto tomat dengan kondisi sebenarnya dalam berbagai model, termasuk pencahayaan, sudut, ukuran, dan warna, untuk memastikan sistem beroperasi sebaik mungkin dalam lingkungan operasional konveyor penyortiran.

### 3. Subjek Penelitian

Data gambar tomat dari pemasok tomat menjadi fokus penelitian ini. Untuk meningkatkan kualitas data selama proses penyortiran, foto tomat diproses langsung menggunakan kamera *smartphone*. Sistem deteksi dan klasifikasi berbasis *SSD-MobileNetV2* yang terintegrasi ke dalam konveyor penyortiran menggunakan data tersebut sebagai alat utama untuk

analisis, evaluasi, dan penilaian kinerja. Hal ini memungkinkan model untuk menilai karakteristik tomat secara akurat dalam berbagai kondisi operasional.

## **E. Prosedur Penelitian**

### **1. Studi Literatur**

Sumber utama yang digunakan dalam proses identifikasi awal studi ini ialah buku, artikel, dan jurnal yang berkaitan dengan *deep learning*, *computer vision*, serta teknologi sensor dan kamera konveyor. Tujuan tinjauan literatur ini ialah untuk memahami secara mendalam teori dasar algoritma *SSD-MobileNetV2*, konsep deteksi dan klasifikasi objek, serta penerapan *computer vision* dalam sistem otomatisasi pertanian. Untuk mengidentifikasi kemajuan metodologis, kelemahan, dan area inovasi, penelitian sebelumnya tentang pemilahan buah menggunakan teknologi berbasis gambar juga ditinjau. Pada tahap ini, para peneliti berhasil mengembangkan kerangka teoritis yang kokoh dan mengidentifikasi cara-cara di mana teknologi dapat meningkatkan efisiensi proses deteksi dan klasifikasi tomat berbasis konveyor.

### **2. Pengumpulan Data**

Sebagai representasi kondisi nyata di lapangan, proses pengumpulan data dilakukan di pemasok tomat di Desa Gebangkerep, Kecamatan Baron, Kabupaten Nganjuk. Sebuah kamera dengan resolusi minimum 5 MP digunakan untuk mengumpulkan data gambar tomat dalam berbagai kondisi pencahayaan. Foto-foto ini diambil untuk mengumpulkan data visual yang objektif dan beragam guna analisis, pelatihan, dan evaluasi model deteksi dan klasifikasi berbasis *SSD MobileNetV2*.

### **3. Analisis Sistem**

Untuk menentukan teknik deteksi dan klasifikasi yang paling efektif, dilakukan analisis sistem berdasarkan tinjauan literatur dan persyaratan operasional sistem konveyor Pemilah. Sistem ini dapat mengidentifikasi tomat dengan presisi saat tomat bergerak di sepanjang konveyor berkat kemampuan deteksi objek yang cepat dan efisien dari algoritma *SSD-*

*MobileNetV2*. Analisis ini mempertimbangkan berbagai faktor, seperti variasi kematangan tomat, ukuran, warna permukaan, dan pencahayaan di area pemindaian, untuk memastikan sistem beroperasi secara stabil dan menghasilkan hasil deteksi terbaik.

#### 4. Desain Sistem

Desain sistem dalam studi ini menjelaskan komponen dan struktur alur kerja yang diperlukan untuk membangun sistem deteksi dan klasifikasi tomat berbasis *SSD MobileNetV2* pada konveyor Pemilah. Tahap desain ini mencakup pembuatan alur kerja pemrosesan gambar, identifikasi mekanisme pengambilan gambar menggunakan *webcam* pada jalur konveyor, serta penentuan persyaratan sistem sebelum melanjutkan ke model deteksi. Desain sistem juga menjelaskan bagaimana data dikumpulkan, diproses, dan dikirimkan ke modul klasifikasi agar setiap komponen memiliki peran yang jelas. Fase pengujian dan implementasi pada bagian berikutnya didasarkan pada tinjauan mendalam desain sistem ini.

#### 5. Implementasi Model

Model ini diimplementasikan dengan mengintegrasikan algoritma *SSD MobileNetV2* menggunakan bahasa pemrograman *Python* yang telah dikembangkan sebelumnya. Model yang telah dilatih, berfungsi sebagai komponen utama untuk deteksi dan klasifikasi tomat, kini terhubung dengan modul pengambilan gambar dari *webcam* pada konveyor Pemilah. Seluruh alur pemrosesan, mulai dari penerimaan gambar hingga *Preprocessing* dan inferensi model, diimplementasikan dalam satu sistem untuk menjamin operasi otomatis dan konsisten. Desain arsitektur sistem diimplementasikan sebagai aplikasi yang mendukung proses penyortiran tomat berbasis *computer vision* dan berfungsi sesuai dengan spesifikasi teknis selama fase implementasi ini.

#### 6. Uji Coba dan Analisis Hasil

Pengujian sistem dilakukan untuk mengevaluasi kelayakan logika dan fungsional setiap komponen yang diimplementasikan. Fokus utama

pengujian adalah kemampuan sistem untuk mengambil gambar dari *webcam*, memprosesnya menggunakan model *SSD-MobileNetV2*, dan menampilkan hasil deteksi dan klasifikasi secara andal pada konveyor Pemilah. Untuk memastikan sistem berfungsi sesuai yang diharapkan, merespons kondisi operasional dunia nyata, dan dapat mengidentifikasi objek tomat berdasarkan kategori yang ditentukan, setiap fungsi diuji.

Selain pengujian fungsional sistem, analisis *confusion matrix* hasil klasifikasi digunakan untuk mengevaluasi kinerja model. *confusion matrix* digunakan untuk menghitung metrik evaluasi seperti akurasi, presisi, *recall*, dan F1-score guna menilai kualitas prediksi model secara lebih rinci. Analisis ini memberikan gambaran tentang kemampuan model untuk membedakan kelas tomat, mengidentifikasi kesalahan klasifikasi, dan menyoroti area yang masih memerlukan perbaikan.

*Debugging* dilakukan untuk menemukan dan memperbaiki kesalahan atau ketidakteraturan dalam alur pemrosesan yang teridentifikasi selama uji coba. Proses ini melibatkan pemeriksaan kode program, verifikasi alur data, dan validasi ulang fungsi pendukung untuk memastikan sistem dapat beroperasi dengan optimal sebelum digunakan dalam pengujian tambahan.

## 7. Penyusunan Laporan

Laporan ini disusun dengan merangkum setiap fase penelitian. Hasil analisis efektivitas model *MobileNetV2 SSD* dalam mengenali dan mengklasifikasikan tomat secara otomatis di konveyor Pemilah disajikan dalam laporan ini, beserta penilaian kinerja sistem. Laporan ini merangkum temuan penelitian utama dan menyajikan kesimpulan mengenai kontribusi sistem dalam meningkatkan efisiensi Pemilahan tomat serta potensinya dalam mendukung modernisasi teknologi pertanian. Oleh karena itu, laporan ini dapat dijadikan acuan bagi peneliti masa depan yang ingin meneliti penerapan *computer vision* dalam otomatisasi penyortiran produk pertanian.

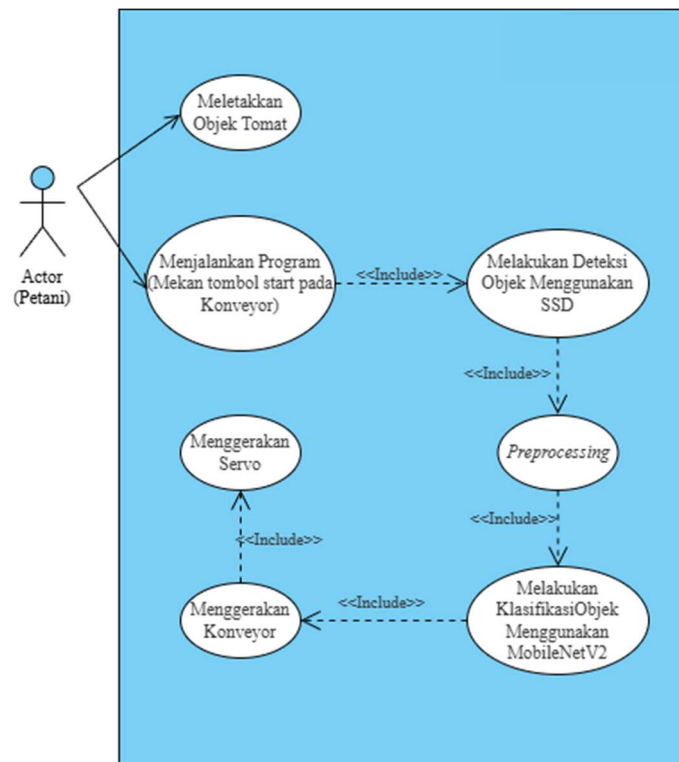
## F. Teknik Analisis Data

### 1. Desain Sistem

#### a. Arsitektur Sistem (*System Architecture*)

Penelitian ini menggunakan pendekatan *Object-Oriented Analysis and Design (OOAD)*, karena paradigma yang digunakan pada sistem ini diimplementasikan dalam bahasa pemrograman *Python*. Penggunaan *OOAD* relevan untuk modularisasi komponen sistem seperti deteksi objek, *preprocessing* citra, ekstraksi fitur hingga klasifikasi.

##### 1) *Use Case Diagram*



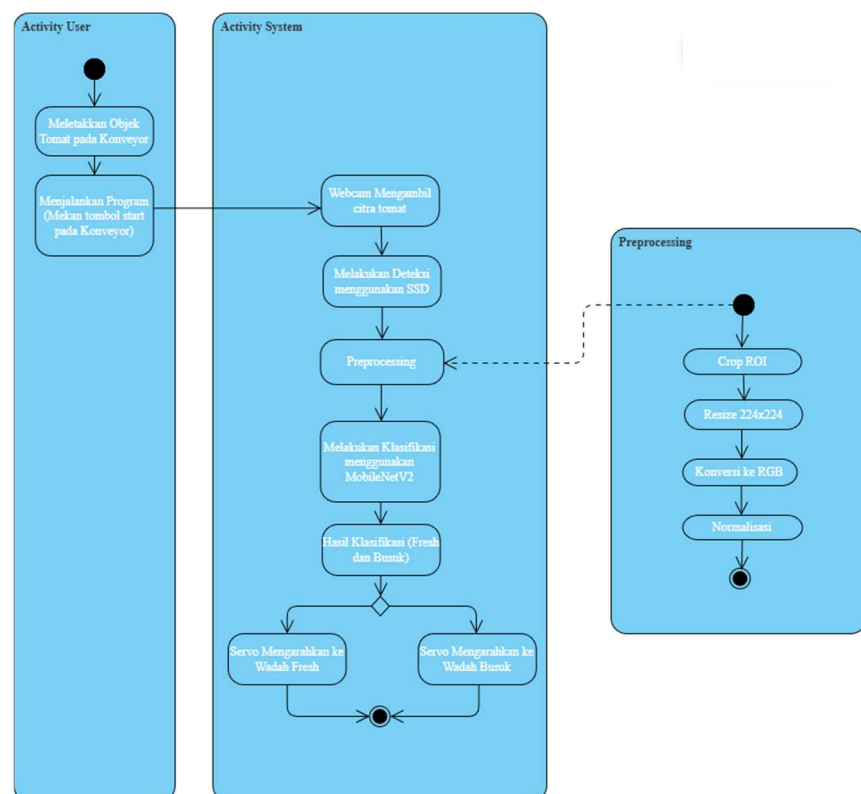
**Gambar 3. 1 Use Case Diagram**

*Use Case diagram* pada Gambar 3.1 menunjukkan bagaimana sistem deteksi dan klasifikasi tomat berbasis *IoT* yang menggunakan konveyor. Petani berinteraksi dengan petani sebagai aktor. Petani meletakkan tomat di atas konveyor dan menekan tombol mulai untuk memulai program. Setelah

program berjalan, sistem mengambil alih proses secara otomatis. Sistem menggunakan *webcam* dengan model *SSD* untuk mengambil foto tomat, yang kemudian diproses menggunakan *preprocessing*.

Setelah *preprocessing*, tomat diklasifikasikan menggunakan *MobileNetV2* untuk mengidentifikasi kategori tomat. Objek tomat dipindahkan di atas konveyor untuk menyelesaikan proses penyortiran setelah sistem menggerakkan *aktuator servo* sesuai dengan kategori berdasarkan hasil klasifikasi. Seluruh prosedur dijalankan secara otomatis dan berurutan. Hal ini memungkinkan sistem untuk menyortir tomat dengan lebih akurat dan efisien.

## 2) Activity Diagram



**Gambar 3. 2 Activity Diagram**

*Activity diagram* pada Gambar 3.2 menunjukkan cara sistem deteksi dan klasifikasi tomat berbasis *IoT* menggunakan

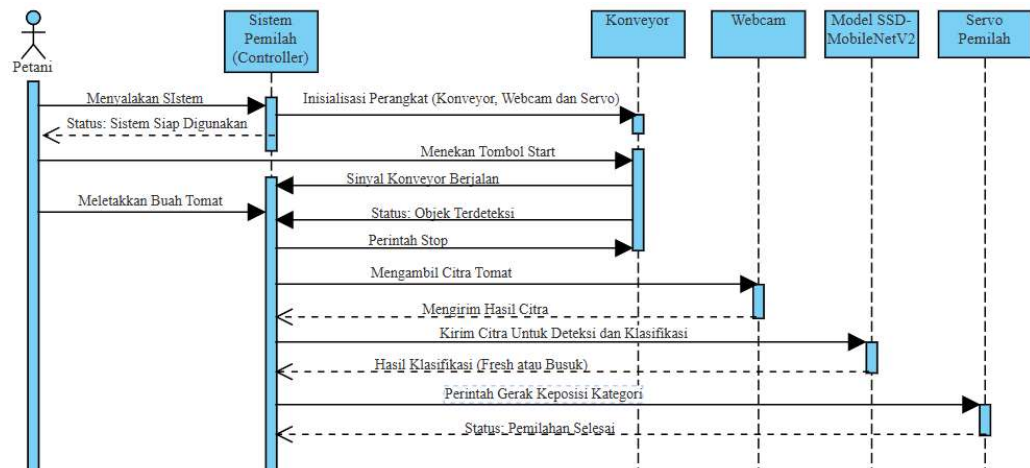
konveyor Pemilah. Dengan meletakkan tomat di atas konveyor dan menekan tombol mulai untuk menjalankan program, aktor (petani) memulai proses. Setelah perintah mulai diberikan, sistem akan menjalankan seluruh alur kerja secara otomatis, tanpa memerlukan masukan lebih lanjut dari aktor.

Sistem kemudian mengaktifkan *webcam* untuk mengambil gambar tomat di atas konveyor sebagai masukan gambar visual. Untuk mendeteksi keberadaan tomat, pengambilan gambar dilengkapi dengan model *SSD* dengan *output* tomat dengan *bounding box*. Selain itu, untuk memaksimalkan gambar masukan untuk klasifikasi, gambar tersebut akan melalui tahap *preprocessing* seperti *Crop ROI (Region of Interest)* guna memotong citra sesuai titik *bounding box*, setelah itu dilakukan *Resize ke 224X224 piksel*, kemudian melakukan konversi ke RGB untuk memastikan input sesuai dan langkah terakhir adalah melakukan normalisasi terhadap nilai RGB. Kemudian masuk tahap klasifikasi Tomat menggunakan *MobileNetV2* untuk menentukan apakah objek masuk dalam kategori segar atau busuk. Untuk memisahkan tomat ke jalur yang sesuai berdasarkan hasil klasifikasi, sistem akan terlebih dahulu menggerakkan konveyor ke arah *servo*. Sesuai dengan kategori masing-masing, tomat akan disimpan di ruang yang telah ditentukan.

### 3) Sequence Diagram

*Sequence diagram* pada Gambar 3.3 menunjukkan alur komunikasi dan urutan operasi antara aktor (petani) dengan komponen dari sistem penyortiran berbasis *IoT*. Setelah petani menyalakan sistem, *Controller* menginisialisasi konveyor, *webcam*, dan perangkat *servo* hingga sistem siap beroperasi. Ketika petani meletakkan tomat di konveyor dan menekan tombol mulai, proses penyortiran dimulai. Setelah *Controller*

mengirim sinyal untuk menghentikan konveyor hingga objek tomat terdeteksi, sistem menghentikan konveyor untuk memulai tahap inspeksi visual.



**Gambar 3.3 Sequence Diagram**

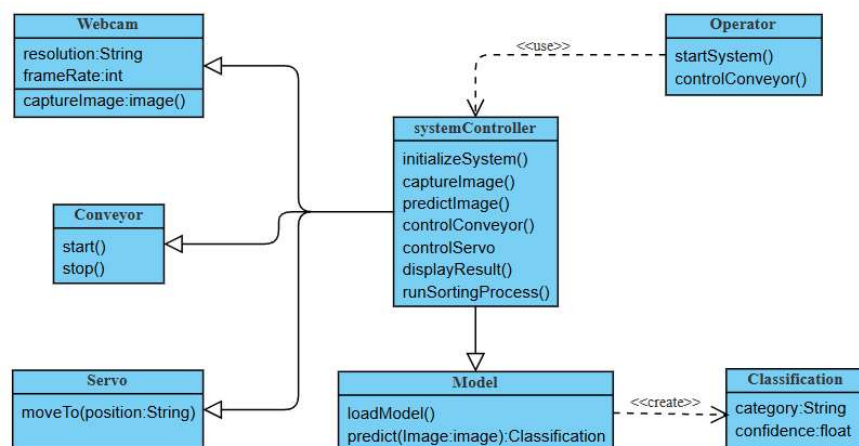
Tahap inspeksi dilakukan ketika *webcam* menangkap gambar tomat dan mengirimkannya ke *controller* untuk diteruskan ke model *SSD* dan *MobileNetV2*. Setelah model menyelesaikan proses deteksi, *controller* menerima hasil klasifikasi (segar atau busuk). *Controller* menggunakan perintah gerakan *servo* untuk mengatur posisi mekanisme penyortiran sesuai dengan kategori kualitas tomat. Setelah penyortiran selesai, sistem kembali ke keadaan siap untuk memproses tomat berikutnya.

#### **b. Desain Database (Database Design)**

*Class Diagram* pada Gambar 3.4 menunjukkan struktur objek dan hubungan antar kelas yang digunakan dalam studi ini. *SystemController* yang merupakan kelas utama sistem, berfungsi sebagai pengendali untuk setiap langkah prosedur, termasuk operasi *aktuator*, prediksi klasifikasi, inisialisasi perangkat, dan pengambilan gambar. *SystemController* berkomunikasi dengan kelas Operator, yang menyediakan perintah untuk memulai sistem

35

dan menjalankan konveyor. Selain itu, *SystemController* berinteraksi dengan kelas *Webcam*, *Conveyor*, dan *Servo* untuk mengawasi fungsi fisik sistem. Sementara *Webcam* menyediakan fungsi *captureImage()* untuk mengambil gambar tomat, *Conveyor* menyediakan fungsi *start()* dan *stop()* untuk memindahkan objek. Fungsi *moveTo()* yang disediakan oleh kelas *Servo* digunakan untuk mengatur mekanisme penyortiran sesuai dengan kategori klasifikasi.

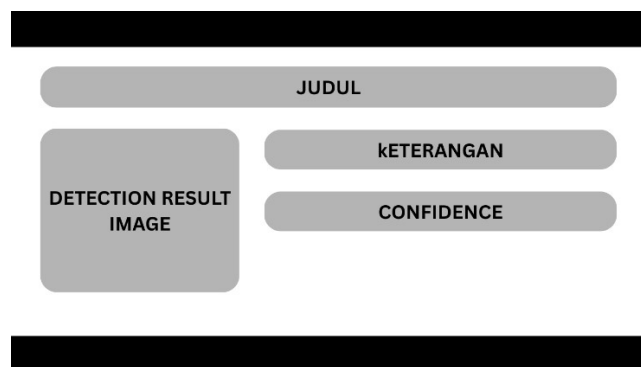


**Gambar 3. 4 Class Diagram**

*SystemController* berkomunikasi dengan kelas *Model*, yang memuat model deteksi dan klasifikasi menggunakan fungsi *loadModel()* dan melakukan inferensi menggunakan fungsi *predict()*. Tingkat kepercayaan (*Confidence*) model diwakili oleh objek *Classification*, yang merupakan hasil dari proses prediksi dan mengandung dua atribut yaitu kategori tomat (segar atau busuk) dan nilai kepercayaan. Fungsi *displayResult()* pada *SystemController* menampilkan objek *Classification* sebelum *servo* menggerakkan konveyor untuk memulai proses penyortiran. Diagram kelas secara jelas menetapkan peran untuk modul kontrol, komponen fisik perangkat *IoT*, dan modul kecerdasan buatan sehingga sistem dapat beroperasi secara otomatis dan terintegrasi.

### c. Desain Antarmuka Pengguna (*UI/UX Design*)

Gambar 3.5 merupakan halaman hasil deteksi di mana halaman ini akan memuat hasil prediksi dari *inputan* konveyor, halaman hasil ini memiliki komponen seperti foto buah setelah dilakukan proses deteksi dengan klasifikasi dengan adanya *bounding box*, setelah itu terdapat hasil klasifikasi berupa *fresh* atau busuk dengan nilai *confidence* antara 0-100 persen.



Gambar 3. 5 Desain Hasil Deteksi

## 2. Simulasi Proses Penyelesaian Masalah

### a. Data *Input*

Gambar tomat yang dikumpulkan secara mandiri menggunakan konveyor Pemilah yang ditujukan sebagai data masukan untuk tahap simulasi. Sebuah *webcam* yang dipasang secara tetap di atas konveyor digunakan untuk memastikan objek selalu berada dalam bidang pandang yang stabil. Tomat diletakkan di atas *belt* konveyor hingga mendekati area deteksi sebagai bagian dari prosedur pengambilan gambar yang sistematis, yang melibatkan pengoperasian konveyor dengan kecepatan tetap. Konveyor akhirnya dihentikan oleh sensor deteksi objek, memungkinkan *webcam* untuk secara otomatis mengambil gambar tomat-tomat tersebut.

Setiap gambar yang dihasilkan mengandung berbagai variasi, termasuk kondisi permukaan, variasi ukuran dan bentuk, hingga

tingkat kematangan (segar atau busuk), selain itu, variasi pencahayaan juga dikendalikan oleh lampu *LED* di area deteksi. Detail warna dan tekstur dapat ditangkap dengan jelas berkat resolusi 1080p dan sensor *RGB* pada *webcam*. Ada sekitar 1.500 gambar dalam data set yang digunakan dan dibagi menjadi dua kelas yaitu tomat segar dan tomat busuk. Setiap kelas memiliki ratusan gambar dalam orientasi yang berbeda. Untuk memastikan sistem dapat mengidentifikasi objek dengan benar dan mengategorikan tingkat kematangan tomat, semua data ini digunakan dalam proses pelatihan model *SSD* dan *MobileNetV2*, validasi kinerja, dan pengujian mekanisme konveyor.

#### b. Proses Pengolahan Data

##### 1) Data Input

Misalkan sistem menerima citra tomat dengan Ukuran asli  $3060 \times 4080 \times 3$  dengan nilai *piksel* pada titik  $R = 235$ ,  $G = 60$ ,  $B = 50$

##### 2) Pemotongan Citra

Setelah proses deteksi citra tomat akan dilakukan pemotongan sesuai dengan titik dan ukuran pada *bounding box*, setelah itu citra akan masuk ke tahap *Resize*.

##### 3) Resize

Standar *input* citra *MobileNetV2* ialah  $224 \times 224 \times 3$ , sehingga perlu dilakukan perhitungan untuk *scaling factor* menggunakan persamaan 2.1:

$$sf = \frac{224}{3060} = 0.073$$
$$sf = \frac{224}{4080} = 0.054$$

Misal pada koordinat (1500,1500) memiliki nilai *piksel* *RGB* 235, 60, 50 setelah dilakukan *resize* nilai dihitung menggunakan persamaan 2.2:

$$1500 \times 0.073 = 110$$

$$1500 \times 0.054 = 81$$

Sehingga setelah proses *rezie* nilai RGB tetap 235, 60, 50 pada koordinat (110,81).

#### 4) Konversi ke *RGB*

Konversi warna ke RGB dilakukan untuk mengantisipasi input yang sesuai dengan yang diharapkan yakni RGB, sehingga tidak terjadi salah representasi fitur warna.

#### 5) Normalisasi

Proses normalisasi dilakukan untuk mengubah rentang nilai piksel dari 0–255 menjadi 0–1 menggunakan persamaan 2.3:

$$R_{norm} = \frac{235}{255} = 0.921$$

$$G_{norm} = \frac{60}{255} = 0.235$$

$$B_{norm} = \frac{50}{255} = 0.196$$

#### 6) Ekstraksi Fitur *MobileNetV2*

##### a) *Depthwise Convolution*

Perhitungan *depthwise convolution* dilakukan menggunakan *kernel*  $3 \times 3$ . Untuk perhitungan kali ini diasumsikan menggunakan *kernel* yang sama untuk ketiga Channel dengan rumus perhitungan menggunakan persamaan 2.4 serta penggunaan aktivasi *ReLU6* setelah menemukan hasil dari *depthwise convolution* dihitung menggunakan persamaan 2.6.

*Kernel*:

$$K_{dw} = \begin{bmatrix} 0 & 0.1 & 0 \\ 0.1 & 0.6 & 0.1 \\ 0 & 0.1 & 0 \end{bmatrix}$$

Perhitungan untuk Channel *Red (R)*:

$$X_R = \begin{matrix} & 0.91 & 0.92 & 0.91 \\ 0.93 & 0.921 & 0.92 & \\ 0.91 & 0.92 & 0.91 & \end{matrix}$$

$$\begin{aligned} Y_{dw\_R} &= (0.91 \cdot 0) + (0.92 \cdot 0.1) + (0.91 \cdot 0) \\ &+ (0.93 \cdot 0.1) + (0.921 \cdot 0.6) \\ &+ (0.92 \cdot 0.1) + (0.91 \cdot 0) + (0.92 \cdot 0.1) \\ &+ (0.91 \cdot 0) = 0.922 \end{aligned}$$

$$R_{act} = \min(\max(0, 0.922), 6) = 0.922$$

Perhitungan untuk Channel *Green* (*G*) :

$$X_G = \begin{matrix} & 0.22 & 0.24 & 0.23 \\ 0.25 & 0.235 & 0.24 & \\ 0.23 & 0.22 & 0.23 & \end{matrix}$$

$$\begin{aligned} Y_{dw\_G} &= (0.22 \cdot 0) + (0.24 \cdot 0.1) + (0.23 \cdot 0) \\ &+ (0.25 \cdot 0.1) + (0.235 \cdot 0.6) \\ &+ (0.24 \cdot 0.1) + (0.23 \cdot 0) + (0.22 \cdot 0.1) \\ &+ (0.23 \cdot 0) = 0.236 \end{aligned}$$

$$G_{act} = \min(\max(0, 0.236), 6) = 0.236$$

Perhitungan untuk Channel *Blue* (*B*):

$$X_B = \begin{matrix} & 0.18 & 0.20 & 0.19 \\ 0.21 & 0.196 & 0.20 & \\ 0.19 & 0.18 & 0.19 & \end{matrix}$$

$$\begin{aligned} Y_{dw\_B} &= (0.18 \cdot 0) + (0.20 \cdot 0.1) + (0.19 \cdot 0) \\ &+ (0.21 \cdot 0.1) + (0.196 \cdot 0.6) \\ &+ (0.20 \cdot 0.1) + (0.19 \cdot 0) + (0.18 \cdot 0.1) \\ &+ (0.19 \cdot 0) = 0.197 \end{aligned}$$

$$B_{act} = \min(\max(0, 0.197), 6) = 0.197$$

#### b) *Pointwise Convolution*

Pada bagian *pointwise convolution* ialah menggabungkan hasil ekstraksi fitur *MobileNetV2* yang telah melewati proses aktivasi *ReLU6* menggunakan *Kernel*  $1 \times 1$ . Nilai bobot *kernel* yang digunakan pada simulasi ini merupakan nilai asumsi representatif model yang telah melalui proses pelatihan (*trained weights*).

Rumus *pointwise convolution* dihitung menggunakan persamaan 2.5 :

$$K_{pw} = [1.0, -0.2, -0.2]$$

$$Y_{pw} = (0.922 \cdot 1.0) + (0.236 \cdot -0.2) + (0.197 \cdot -0.2)$$

$$= 0.835$$

c) *Global Average Pooling* (GAP)

Untuk perhitungan simulasi ini, diasumsikan *pointwise convolution* menghasilkan fitur *spasial* berdimensi  $H=2$  dan  $W=2$ . *GAP* dihitung menggunakan persamaan 2.7.

$$F_c = \begin{bmatrix} 0.835 & 0.820 \\ 0.841 & 0.845 \end{bmatrix}$$

$$GAP = \frac{1}{2 \times 2} (0.835 + 0.820 + 0.841 + 0.845)$$

$$= 0.835$$

d) Dense Layer

Menghitung nilai *logits* untuk 2 kategori, yaitu *fresh* dan busuk menggunakan persamaan 2.8. Di mana bobot  $W_f = 1.5$  dengan  $b = 0.2$ , dan untuk kelas busuk memiliki bobot  $W_b = -1.0$  dengan  $b = 0.2$ .

$$Z_f = (0.835 \cdot 1.5) + 0.2 = 1.453$$

$$Z_b = (0.835 \cdot -1.0) + 0.2 = -0.635$$

7) *Softmax*

Hitung eksponensial:

$$e^{1.453} \approx 4.277$$

$$e^{-0.635} \approx 0.530$$

$$Total = 4.277 + 0.530 = 4.807$$

Probabilitas *Fresh*:

$$P_{fresh} = \frac{4.277}{4.807} = 0.889 \text{ (88,9\%)}$$

Probabilitas Busuk:

$$P_{busuk} = \frac{0.530}{4.807} = 0.110 \text{ (11,0\%)}$$

Sistem akan melakukan pengecekan menggunakan *Confidence Thresholding* pada persamaan 2.10 apakah nilai *confidence*  $P_{fresh} > 80\%$ , jika iya maka sistem akan menyatakan tomat *fresh*, namun jika tidak maka akan dinyatakan busuk. Sehingga didapati hasil akhir klasifikasi Tomat *Fresh* dengan *confidence* 88,9%.

### c. Data output dan evaluasi

#### 1) Nama Kelas Tomat

Luaran utama sistem ini adalah label yang menggunakan hasil klasifikasi model *MobileNetV2* untuk mengklasifikasikan tomat berdasarkan kualitasnya. Luaran dalam studi ini dibagi menjadi dua kelas, yaitu:

- a) Segar (*fresh*): Tomat dalam kategori ini cocok untuk dikonsumsi karena dalam kondisi fisik yang baik, memiliki warna kulit yang cerah, dan tidak memiliki cacat yang terlihat.
- b) Busuk (*rotten*): Tomat dengan perubahan warna abnormal, tekstur lembek, atau adanya cacat pada permukaan..

#### 2) Tingkat Probabilitas

Sistem ini menyediakan nilai probabilitas atau *confidence score* sebagai indikator kuantitatif hasil klasifikasi selain label kelas. Tingkat kepercayaan model *MobileNetV2* terhadap prediksi kategorinya diwakili oleh nilai tersebut.

Fungsi aktivasi *Softmax* digunakan untuk mengekstrak *confidence score* dari lapisan *output*. Nilai *confidence score* berkisar antara 0% hingga 100%, atau 0 dan 1. Misalnya, jika sistem menampilkan label *Fresh* dengan nilai 0,92, model memiliki tingkat kepercayaan 92% bahwa gambar tersebut

adalah tomat *Fresh*. Sisanya 8% mewakili kemungkinan kesalahan atau *noise*.

### 3) *Confusion matrix*

Seperti terlihat pada Tabel 3.2, *confusion matrix* digunakan untuk mengevaluasi kinerja model. Untuk menentukan jumlah prediksi yang akurat (*True Positive* dan *True Negative*) serta kesalahan klasifikasi berupa *False Positive* (tomat busuk terdeteksi sebagai segar) dan *False Negative* (tomat segar terdeteksi sebagai busuk), matriks ini memetakan kelas aktual (baris) ke kelas yang diprediksi (kolom).

**Tabel 3. 2 *Confusion Matrix***

|                     | Prediksi <i>Fresh</i>      | Prediksi Busuk             |
|---------------------|----------------------------|----------------------------|
| Aktual <i>Fresh</i> | <i>TP (True Positive)</i>  | <i>FN (False Negative)</i> |
| Aktual Busuk        | <i>FP (False Positive)</i> | <i>TN (True Negative)</i>  |

Metrik utama yang digunakan untuk mengukur kinerja model berdasarkan komponen matriks diantaranya, ketepatan prediksi secara keseluruhan diukur oleh Akurasi (Persamaan 2.11), tingkat kepercayaan prediksi positif diukur oleh Presisi (Persamaan 2.12), sensitivitas deteksi model diukur oleh *Recall* (Persamaan 2.13), dan nilai presisi dan *recall* diseimbangkan oleh F1-Score (Persamaan 2.14), yang merupakan rata-rata harmonik.

43

## BAB IV

### HASIL DAN PEMBAHASAN

#### A. Hasil Penelitian

Bab ini menyajikan hasil implementasi sistem deteksi tomat otomatis dalam ekosistem IoT yang di mana seperti pada rancangan Bab III. Penelitian ini menggunakan model *hybrid Single Shot Detector (SSD)* sebagai *head detector* dengan *backbone MobileNetV2*, yang berfungsi untuk melakukan klasifikasi tomat dalam kelas segar (*fresh*) dan busuk (*rotten*). Proses pengambilan dataset dilakukan secara mandiri pada petani tomat di desa Gebangkerep kecamatan Baron kabupaten Nganjuk Jawa Timur.

19



**Gambar 4. 1 Pengambilan Dataset**

Pada Gambar 4.1 menunjukkan proses pengambilan dataset yang langsung dipandu oleh ahli (petani). Data yang berhasil dikumpulkan berjumlah 364 citra tomat dengan 163 citra tomat *fresh* dan 201 citra tomat *rotten* dengan berbagai kondisi pengambilan gambar dan pencahayaan guna merepresentasikan dengan kondisi pada lapangan. Proses pengambilan gambar ini dilakukan menggunakan kamera Iphone 11 dengan resolusi 12Mp dan Webcam EYD 1080p.

49

**Tabel 4. 1 Penyajian Sampel Dataset Tomat**

| NO | Sampel Dataset                                                                    | Kelas  | Warna            | Kondisi                             |
|----|-----------------------------------------------------------------------------------|--------|------------------|-------------------------------------|
| 1  |  | Rotten | Hijau Kekuningan | Banyak bercak hitam di permukaannya |
| 2  |  | Fresh  | Merah segar      | Bersih mulus                        |
| 3  |  | Fresh  | Oranye           | Bersih mulus tanpa bercak           |
| 4  |  | Rotten | Kuning kemerahan | Terdapat lubang tanda pembusukan    |
| 5  |  | Rotten | Merah            | Terlalu matang dan busuk            |

Pengumpulan data citra tomat pada Tabel 4.1 ini dilakukan secara terstruktur untuk menangkap variasi visual fisik tomat, di mana klasifikasi secara ketat memisahkan objek ke dalam kelas Fresh dengan ciri kulit bersih mulus berwarna merah segar atau oranye, serta kelas Rotten yang dicirikan oleh kerusakan fisik seperti bercak hitam, lubang pembusukan, atau kondisi terlalu matang. Seluruh sampel gambar tomat ini kemudian melewati proses anotasi data untuk menandai area cacat, noda, atau busuk secara presisi agar algoritma dapat mengekstrak fitur visual secara otomatis.

## 1. Implementasi Desain Sistem

Implementasi desain sistem pada penelitian ini dilakukan dengan menyusun beberapa lembar kerja atau modul yang memiliki fungsi khusus namun saling terkait secara sistematis. Setiap modul ini dibangun secara modular guna memisahkan tugas spesifik, sehingga seluruh komponen dapat saling mendukung keterkaitan penerapan dalam mengklasifikasikan kualitas tomat.

### a. Implementasi Lembar Kerja

Pada lembar kerja atau modul penelitian ini terdiri dari beberapa tahapan:

### 1) Modul Akuisisi Data

Tahap awal dalam implementasi desain sistem diawali dengan proses akuisisi data melalui pengumpulan citra tomat secara langsung dari lapangan guna membangun dataset yang representatif. Data citra yang dihimpun kemudian dikelompokkan secara terstruktur ke dalam dua kategori utama, yakni kelas fresh untuk tomat dengan kondisi kulit bersih mulus serta kelas rotten untuk tomat yang memiliki kecacatan morfologi seperti bercak hitam, lubang pembusukan, atau kondisi lewat matang. Rincian sampel karakteristik dataset citra yang mencakup aspek visual warna dan kondisi fisik objek disajikan secara komparatif pada Tabel 4.1 sebagai landasan bagi model untuk mengenali variasi fitur di lingkungan nyata.

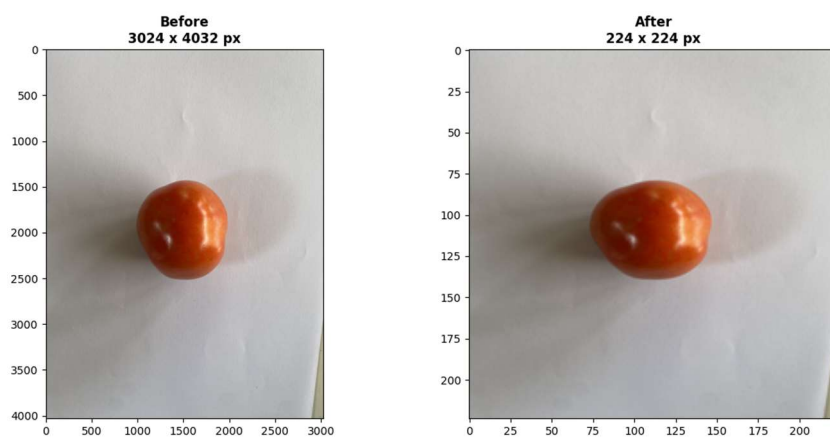


**Gambar 4. 2 Sampel data dengan anotasi**

Setelah dataset citra berhasil diklasifikasikan, langkah selanjutnya adalah melakukan *bounding box* pada objek secara presisi menggunakan perangkat lunak LabelImg. Dalam proses ini, setiap gambar dianotasi dengan membuat kotak pembatas (*bounding box*) yang menandai lokasi spasial piksel tomat dalam bingkai gambar untuk menghasilkan file anotasi berformat .xml. File anotasi tersebut memuat informasi mengenai koordinat dan label kelas yang sangat krusial bagi arsitektur SSD MobileNetV2 dalam mempelajari fitur geometri serta membedakan antara objek target dengan latar belakang selama fase pelatihan model.

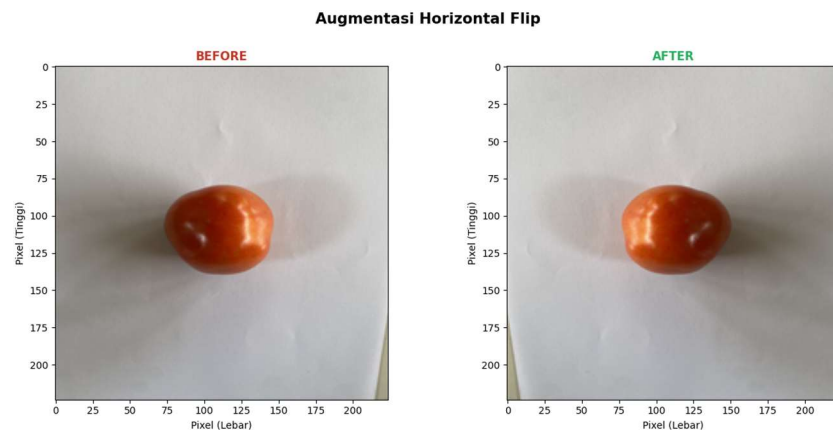
## 2) Modul Pra-pemrosesan (Preprocessing)

Tahap krusial dalam prapemrosesan ini adalah melakukan pengubahan ukuran gambar (resizing) secara dinamis menjadi dimensi 224x224 piksel. Penyeragaman ukuran ini wajib dilakukan agar resolusi spasial dari citra tomat dari lapangan dapat diselaraskan secara presisi dengan dimensi input yang diminta oleh input layer arsitektur MobileNetV2.



**Gambar 4. 3 Resize image 224x224 piksel**

Setelah dimensi gambar diseragamkan, tahapan berikutnya adalah menerapkan teknik augmentasi data untuk memperkaya variasi karakteristik dataset secara artifisial. Proses augmentasi ini mengombinasikan beberapa teknik transformasi citra, meliputi horizontal flip (Seperti pada Gambar 4.4) dan vertical flip untuk membalik posisi geometris objek secara cermin, random brightness contrast untuk menyimulasikan fluktuasi intensitas cahaya di lapangan, gaussian blur untuk memberikan efek keaburan visual, serta rotate untuk memutar sudut kemiringan objek tomat. Penerapan manipulasi visual ini berfungsi memperluas sebaran data latih guna mencegah terjadinya overfitting, sekaligus meningkatkan ketangguhan (robustness) model saat mendeteksi tomat dalam berbagai posisi ekstrem di atas konveyor.



**Gambar 4. 4 Sampel Augmentation (Horizontal Flip)**

Setelah piksel gambar diseragamkan, langkah berikutnya adalah menerapkan fungsi normalisasi piksel dengan cara membagi setiap matriks warna (saluran RGB) dengan nilai 255. Proses pembagian ini berfungsi untuk **mengubah rentang nilai intensitas piksel dari skala asli 0–255 menjadi skala desimal antara 0 hingga 1** guna mempercepat waktu konvergensi model saat fase training dan menyamakan rentang skala kecerahan gambar.

### 3) Modul Ekstraksi Fitur dan Klasifikasi (SSD MobileNetV2)

Arsitektur Deep Learning yang berhasil dibangun di dalam kode program ini mengadopsi model hybrid mutakhir yang mengombinasikan kecepatan dan ketepatan deteksi secara simultan. Pada lapisan fondasi, sistem mengimplementasikan MobileNetV2 sebagai komponen backbone utama yang berperan penting dalam mengekstrak fitur-fitur visual dari citra tomat secara mendalam. Berkat integrasi fungsi aktivasi ReLU6 di dalam strukturnya, backbone ini mampu membatasi nilai aktivasi maksimum hingga angka 6, sehingga sangat stabil saat memproses ekstraksi gradasi warna kematangan serta tekstur permukaan tomat (seperti kondisi mulus pada buah segar maupun

kerutan dan bercak pada buah busuk) meski dijalankan pada perangkat dengan spesifikasi komputasi terbatas.

Untuk mengoptimalkan proses pembelajaran arsitektur tersebut, fase pelatihan dikonfigurasi menggunakan serangkaian hyperparameter spesifik yang disesuaikan dengan karakteristik dataset tomat. Pengondisian ukuran gambar 224x224 piksel diterapkan agar resolusi spasial citra masukan selaras dengan spesifikasi input layer bawaan MobileNetV2, sementara ukuran batch size 8 dipilih untuk menjaga stabilitas alokasi memori (VRAM) komputer lokal selama proses iterasi berlangsung. Pelatihan dieksekusi secara mendalam hingga 120 epoch guna memberikan waktu yang cukup bagi jaringan untuk mencapai titik konvergensi yang matang. Proses pembaruan bobot jaringan dikendalikan oleh nilai learning rate  $5e-5$  yang relatif kecil bersama dengan regulerisasi weight decay  $5e-5$  untuk mencegah terjadinya gradient exploding serta meminimalisir risiko overfitting. Seluruh basis data citra kemudian dibagi secara proporsional ke dalam tiga sub-himpunan terpisah dengan train ratio 0.7 untuk pembaruan bobot model, rasio 0.15 sebagai validator berkala di setiap epoch, serta rasio 0.15 sebagai data uji murni untuk mengukur generalisasi objektif model di akhir pipeline.

Selanjutnya, fitur-fitur visual yang telah diekstraksi oleh MobileNetV2 diumpankan secara langsung menuju lapisan Single Shot Multibox Detector (SSD). Layer SSD ini memegang peran sentral dalam mendeteksi multi-objek secara real-time dengan cara memprediksi skor kelas (fresh atau rotten) sekaligus menghasilkan koordinat kotak pembatas (bounding box) secara simultan dalam satu kali langkah komputasi (single shot). Melalui mekanisme piramida fitur dari SSD, sistem dapat menentukan batas-batas spasial objek tomat di atas konveyor secara akurat,

memastikan tidak ada objek yang terlewatkan, dan siap mengalirkan data koordinat tersebut menuju tahap evaluasi performa model.

#### **b. Keterkaitan Antar Lembar Kerja**

Keterkaitan antar lembar kerja atau modul di dalam sistem pemilah cerdas ini dibangun menggunakan arsitektur terdistribusi berbasis protokol komunikasi Web API. Hubungan fungsional antara seluruh komponen dikendalikan secara sinkron guna menjamin bahwa aliran data citra dari lingkungan fisik dapat diproses secara real-time oleh model kecerdasan buatan, yang kemudian hasilnya dikonversi kembali menjadi aksi mekanis.

Berdasarkan diagram keterkaitan sistem pada Gambar 4.8, proses interaksi diawali ketika Modul Akuisisi Data menangkap bingkai gambar buah tomat di atas jalur konveyor (conveyor) melalui sensor kamera. Bingkai gambar mentah tersebut secara instan dikirimkan menuju server lokal melalui permintaan HTTP (HTTP Request) protokol Web API yang dikelola oleh mikrokontroler. Setibanya data di sisi server Python, Modul Pra-pemrosesan (Preprocessing) langsung mengambil alih untuk mengeksekusi fungsi resizing dinamis ke dimensi 224x224 piksel, melakukan normalisasi nilai matriks warna dengan faktor pembagi 255, serta melakukan penguncian koordinat (clamping).

Setelah kondisi citra memenuhi standar komputasi, data digital tersebut diteruskan secara internal menuju Modul Ekstraksi Fitur dan Klasifikasi (SSD-MobileNetV2). Pada tahapan ini, arsitektur MobileNetV2 mengekstrak karakteristik warna dan struktur geometri permukaan tomat, sementara lapisan SSD secara simultan memprediksi kelas kualitas (fresh atau rotten) serta menentukan batas spasial objek. Hasil akhir inferensi berupa nilai indeks kelas dan skor keyakinan (confidence score) divalidasi keandalannya oleh Modul Evaluasi Internal menggunakan metrik confusion matrix. Struktur

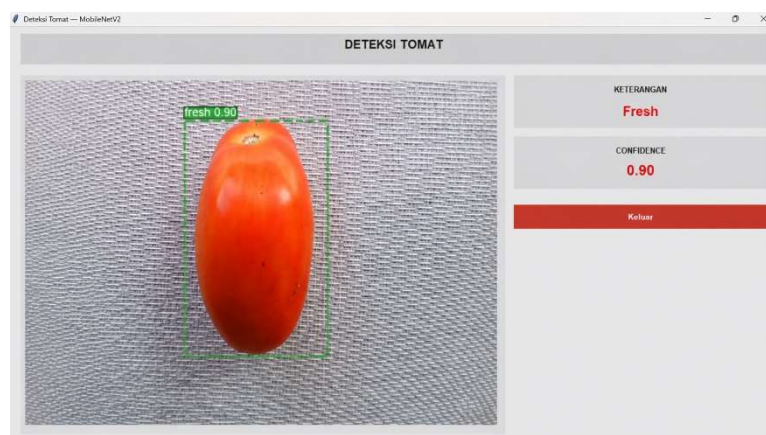
data keputusan final tersebut kemudian dikemas ke dalam format respons Web API menuju edge client (ESP32) untuk mengaktifkan pin digital aktuator motor servo, sehingga buah tomat dapat dipilah secara otomatis ke wadah grade yang sesuai.

## 2. Pengujian Fungsional

Pengujian fungsional dalam penelitian ini menerapkan metode Black Box Testing untuk memvalidasi ketepatan respons sistem terhadap input pengguna. Fokus utama tahapan ini adalah memastikan seluruh fitur sortir berjalan sesuai rencana tanpa meninjau struktur internal kode programnya.

### a. Implementasi Sistem Antarmuka

Gambar 4.5 menampilkan antarmuka utama sistem deteksi tomat berbasis Graphical User Interface (GUI) yang berjalan secara real-time. Panel sebelah kiri menampilkan fitur preview kamera yang berfungsi menampilkan visual objek yang telah terdeteksi pada konveyor, lengkap dengan bounding box putus-putus serta label kelas yang disertai nilai confidence score tepat di atas objek tomat yang terdeteksi. Sementara itu, panel sebelah kanan terbagi menjadi dua bagian fungsional, yaitu kolom Keterangan yang berfungsi mengonfirmasi hasil klasifikasi tomat pada kualitas fresh atau rotten, serta kolom Confidence yang menampilkan nilai akurasi prediksi model terhadap objek tersebut dengan satuan persen.



**Gambar 4. 5 Sistem Antarmuka**

Berdasarkan visualisasi antarmuka sistem pada Gambar 4.5, sistem berhasil mendeteksi objek tomat dan mengklasifikasikannya ke dalam kelas Fresh (Segar). Keputusan klasifikasi tersebut dikonfirmasi secara real-time melalui fitur Keterangan pada panel sebelah kanan, sementara nilai akurasi mencapai 90% pada kolom confidence. Nilai akurasi yang tinggi ini menunjukkan bahwa fitur visual berhasil diekstraksi dengan sangat baik oleh model SSD MobileNetV2. Setelah proses klasifikasi dikonfirmasi, sinyal kontrol dikirim menuju mikrokontroler ESP32 guna menggerakkan motor servo konveyor menuju wadah penampungan yang sesuai.

#### b. Black Box Testing

Pengujian fungsional dengan metode Black Box Testing dilaksanakan terhadap tujuh fungsi utama yang mencakup seluruh alur operasional sistem pemilah. Setiap komponen diuji dengan memberikan kondisi masukan tertentu di lapangan guna mengamati kesesuaian antara respon aktual sistem dengan hasil yang diharapkan.

**Tabel 4. 2 Hasil Pengujian Fungsional**

| Pengujian                    | Langkah Pengujian                                                     | Hasil Yang Diharapkan                                                                      | Keterangan |
|------------------------------|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------|------------|
| Inisialisasi Program         | Program dijalankan.                                                   | Webcam aktif dan server API berjalan.                                                      | Berhasil   |
| Deteksi Tomat                | Tomat diletakkan di konveyor pada area deteksi.                       | Tomat terdeteksi dengan munculnya bounding box dan confidence score pada area objek tomat. | Berhasil   |
| Klasifikasi Kelas Fresh      | Menempatkan Tomat dengan kondisi bagus pada konveyor untuk dideteksi. | Panel “Keterangan” menampilkan hasil “Fresh”.                                              | Berhasil   |
| Klasifikasi Kelas Rotten     | Menempatkan Tomat dengan kondisi busuk pada konveyor untuk dideteksi. | Panel “Keterangan” menampilkan hasil “Rotten”.                                             | Berhasil   |
| Menampilkan Confidence Score | Model melakukan ekstraksi pada                                        | Panel “Confidence” menampilkan nilai                                                       | Berhasil   |

| Pengujian                       | Langkah Pengujian                                               | Hasil Yang Diharapkan                                                | Keterangan |
|---------------------------------|-----------------------------------------------------------------|----------------------------------------------------------------------|------------|
|                                 | karakter visual objek tomat .                                   | akurasi (Confidence Score) setelah hasil deteksi.                    |            |
| Integrasi Aktuator Kelas Fresh  | Sistem mengonfirmasi objek merupakan tomat dengan kelas Fresh.  | Servo bergerak dan mengarahkan tomat ke wadah bagus.                 | Berhasil   |
| Integrasi Aktuator Kelas Rotten | Sistem mengonfirmasi objek merupakan tomat dengan kelas Rotten. | Servo tidak aktif dan membiarkannya lewat menuju wahana tomat busuk. | Berhasil   |
| Tombol Keluar                   | Menekan tombol keluar                                           | Sistem berhenti bekerja dan kamera webcam non-aktif                  | Berhasil   |

Berdasarkan data hasil pengujian fungsional pada tabel 4.2, seluruh komponen utama di dalam sistem terbukti beroperasi dengan baik tanpa ditemukan adanya kegagalan fungsi. Pada fase inisialisasi awal, pengujian menunjukkan bahwa program mampu mengaktifkan webcam sekaligus menjalankan server API untuk mempersiapkan jalur pertukaran data. Ketika objek melintas, tahapan deteksi tomat dan pemetaan batas spasial objek bekerja yang ditandai dengan kemunculan bounding box serta label confidence score tepat pada area objek sasaran. Kemampuan pelokalan objek secara real-time ini mengonfirmasi bahwa penyeragaman dimensi gambar masukan dan fungsionalitas algoritma pelacakan di dalam sistem telah terintegrasi dengan sangat baik.

Selain itu, akurasi model terverifikasi secara konsisten melalui keberhasilan pengujian klasifikasi pada kelas Fresh dan kelas Rotten. Panel antarmuka secara instan memperbarui kolom Keterangan dan memunculkan tingkat akurasi prediksi pada panel Confidence berdasarkan hasil ekstraksi karakteristik visual permukaan kulit buah tomat. Aliran data keputusan digital dari server Python tersebut kemudian berhasil ditransmisikan menuju perangkat keras IoT secara

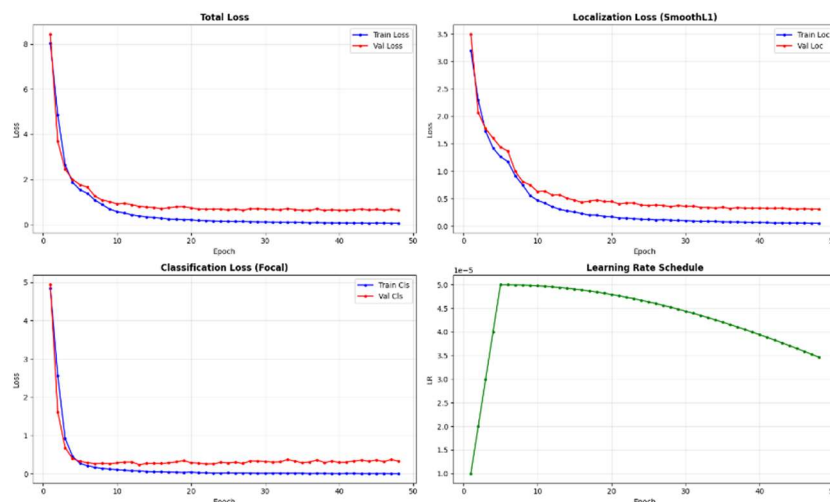
sinkron. Hal ini dibuktikan melalui keberhasilan fungsi mekanis pada tahap akhir, di mana motor servo secara presisi bergerak aktif membuka gerbang pembatas untuk mengarahkan kelas Fresh ke wadah bagus, serta tetap berada pada kondisi tidak aktif guna membiarkan objek meluncur lewat menuju wadah tomat busuk saat mendeteksi kelas Rotten.

### 3. Pengujian Non-Fungsional

Pada Pengujian Non-Fungsional ini terdapat 2 sub bab pengujian, yakni evaluasi kinerja sistem untuk pengujian performa model serta skenario pengujian yang berfokus pada pengujian prototipe pada lingkungan nyata dan dengan berbagai kondisi.

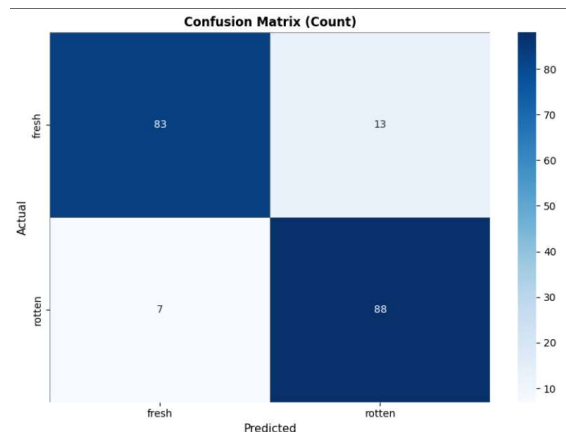
#### a. Evaluasi Kinerja Sistem

Evaluasi kinerja sistem sangat krusial untuk mengukur tingkat konvergensi nilai loss serta kemampuan generalisasi arsitektur SSD MobileNetV2 dalam mengenali fraksi kualitas tomat secara objektif sebelum diintegrasikan dengan sistem mekanis konveyor di lapangan. Nilai penurunan loss dan kestabilan proses training dipaparkan secara berkala melalui visualisasi grafik riwayat pelatihan pada Gambar 4.4.



**Gambar 4. 6 Grafik Training History**

Berdasarkan visualisasi grafik riwayat pelatihan pada Gambar 4.6, dapat dianalisis bahwa model mengalami proses pembelajaran yang sangat sehat dan stabil dari epoch awal hingga mencapai titik akhir. Nilai total loss baik pada data pelatihan (train) maupun data validasi (val) menunjukkan tren penurunan yang tajam secara konvergen tanpa adanya indikasi overfitting atau underfitting, di mana kurva validasi bergerak selaras mendampingi kurva pelatihan. Penurunan nilai loss yang stabil hingga mendekati nilai minimum ini membuktikan bahwa pembatasan fungsi aktivasi ReLU6 pada backbone MobileNetV2 berhasil menjaga kestabilan bobot jaringan dan mempercepat proses konvergensi model. Untuk memetakan akurasi detail sebaran prediksi pada tiap-tiap kelas, visualisasi matriks evaluasi disajikan melalui Confusion Matrix pada Gambar 4.5.



**Gambar 4. 7 Heatmap Confusion Matrix**

Melalui analisis lembar kerja heatmap confusion matrix pada Gambar 4.7, tingkat presisi dan sensitivitas arsitektur hybrid ini dalam memilah kualitas tomat dapat dibuktikan secara kuantitatif. Berdasarkan matriks tersebut, akumulasi nilai prediksi benar (True Positive dan True Negative) mendominasi baris diagonal utama, yang menandakan bahwa model memiliki tingkat akurasi yang sangat tinggi dalam mengklasifikasikan tomat ke dalam kelas fresh maupun kelas rotten. Nilai kesalahan klasifikasi (False Positive dan False Negative) tercatat sangat minim, yang berarti risiko terjadinya salah

pemilahan antara tomat segar yang mulus dengan tomat busuk berbercak telah berhasil ditekan pada tingkat paling rendah dalam lingkungan simulasi komputer.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| fresh        | 0.92      | 0.86   | 0.89     | 96      |
| rotten       | 0.87      | 0.93   | 0.90     | 95      |
| accuracy     |           |        | 0.90     | 191     |
| macro avg    | 0.90      | 0.90   | 0.90     | 191     |
| weighted avg | 0.90      | 0.90   | 0.90     | 191     |

**Gambar 4. 8 Nilai akurasi, presisi, recall dan F1-Score**

Berdasarkan Gambar 4.8 performa internal model menunjukkan tingkat keberhasilan yang sangat tinggi dan seimbang dalam mengklasifikasikan kualitas tomat dengan nilai akurasi total mencapai 90% dari keseluruhan 191 sampel data uji. Pada deteksi objek, kelas fresh unggul dalam ketepatan prediksi dengan nilai precision sebesar 92%, walaupun nilai sensitivitas atau recall-nya berada di angka 86% dari total 96 data support. Sebaliknya, kelas rotten memiliki tingkat sensitivitas penemuan objek yang lebih tinggi dengan nilai recall mencapai 93%, meskipun nilai precision-nya berada di angka 87% dari 95 data support. Keseimbangan performa model otomatis ini dipertegas oleh nilai f1-score yang sangat stabil di angka 89% untuk kelas fresh dan 90% untuk kelas rotten, serta didukung oleh macro avg dan weighted avg yang secara konsisten menunjukkan performa merata di angka 90%, sehingga membuktikan bahwa arsitektur deep learning yang dibangun sangat andal dan siap diintegrasikan pada sistem pemilah fisik di lapangan.

#### **b. Skenario Pengujian Prototipe**

Pengujian eksperimental dilaksanakan secara terstruktur dengan menggunakan 10 sampel objek tomat untuk setiap skenario pengujian yang telah dirancang. Melalui variasi karakteristik fisik pada masing-masing sampel tersebut, sistem diuji pada 9 skenario pengujian hingga mencapai total 90 kali pengujian.

**Tabel 4. 3 Hasil Pengujian Non-Fungsional**

| Kecepatan (RPM) | Kondisi Pencahayaan | Total Pengujian | Benar | Salah | Tidak Terdeteksi | Akurasi (%) | AVG Confidence |
|-----------------|---------------------|-----------------|-------|-------|------------------|-------------|----------------|
| 80              | Cahaya Ruangan      | 10              | 9     | 1     | 0                | 90          | 0,891          |
| 80              | Lampu Led           | 10              | 10    | 0     | 0                | 100         | 0,923          |
| 80              | Tanpa Cahaya        | 10              | 2     | 0     | 8                | 20          | 0,187          |
| 120             | Cahaya Ruangan      | 10              | 9     | 0     | 1                | 90          | 0,876          |
| 120             | Lampu Led           | 10              | 8     | 1     | 1                | 80          | 0,814          |
| 120             | Tanpa Cahaya        | 10              | 1     | 0     | 9                | 10          | 0,093          |
| 240             | Cahaya Ruangan      | 10              | 2     | 1     | 7                | 20          | 0,103          |
| 240             | Lampu Led           | 10              | 3     | 1     | 6                | 30          | 0,315          |
| 240             | Tanpa Cahaya        | 10              | 0     | 0     | 10               | 0           | 0,000          |

Berdasarkan hasil pengujian non-fungsional pada Tabel 4.3, model menunjukkan performa yang bervariasi sesuai dengan kondisi kecepatan conveyor dan pencahayaan yang digunakan. Pada kecepatan 80 RPM, model mampu mendeteksi objek dengan baik pada kondisi cahaya ruangan maupun lampu LED dengan akurasi mencapai 90% hingga 100%. Namun pada kondisi tanpa cahaya, model tidak mampu mendeteksi objek sama sekali dengan akurasi 0% di seluruh variasi kecepatan.

Seiring meningkatnya kecepatan conveyor, performa model mengalami penurunan yang cukup signifikan. Pada kecepatan 120 RPM akurasi berkisar antara 0% hingga 90%, sedangkan pada kecepatan 240 RPM akurasi turun menjadi 0% hingga 70%. Hal ini menunjukkan bahwa kecepatan conveyor dan kondisi pencahayaan merupakan dua faktor utama yang memengaruhi kemampuan model dalam mendeteksi objek secara akurat.

## B. Pembahasan

Analisis komprehensif terhadap prototipe sistem sortir tomat otomatis ini mencakup evaluasi fungsional dan non-fungsional untuk mengukur sejauh

19 mana model *hybrid Single Shot Detector (SSD)* dengan *backbone MobileNetV2* dapat diandalkan dalam mengklasifikasikan kualitas tomat secara otomatis di dalam ekosistem IoT. Pembangunan model ini diawali dengan tahap akuisisi data yang dilakukan secara mandiri di lapangan, yakni di desa Gebangkerep, kecamatan Baron, kabupaten Nganjuk, Jawa Timur, dengan total 364 citra tomat yang terdiri dari 163 citra kelas fresh dan 201 citra kelas rotten. Variasi kondisi pengambilan gambar serta penggunaan dua perangkat berbeda, yaitu kamera iPhone 11 beresolusi 12MP dan webcam EYD 1080p, sengaja diterapkan untuk merepresentasikan kondisi nyata di lapangan sehingga model yang dibangun memiliki kemampuan generalisasi yang lebih baik. 38 Pengelompokan dataset ke dalam kelas fresh dan rotten juga dilakukan secara terstruktur berdasarkan ciri visual yang jelas, seperti warna kulit, ada tidaknya bercak hitam, lubang pembusukan, maupun kondisi terlalu matang, sebagaimana disajikan pada Tabel 4.1.

86 Pada tahap implementasi desain sistem, proses pengolahan data tomat melalui beberapa modul yang saling terintegrasi, mulai dari akuisisi data, anotasi bounding box menggunakan LabelImg, hingga prapemrosesan citra. 104 Tahap prapemrosesan dilakukan dengan menyeragamkan ukuran citra menjadi 224x224 piksel agar sesuai dengan kebutuhan input layer MobileNetV2, kemudian dilanjutkan dengan augmentasi data berupa horizontal flip, vertical flip, random brightness contrast, gaussian blur, dan rotate untuk memperkaya variasi dataset serta mencegah terjadinya overfitting. 24 Normalisasi nilai piksel ke rentang 0–1 turut diterapkan untuk mempercepat konvergensi model selama pelatihan. Pada tahap ekstraksi fitur dan klasifikasi, arsitektur MobileNetV2 yang dilengkapi fungsi aktivasi ReLU6 berperan mengekstrak fitur visual tomat secara stabil, sementara lapisan SSD bertugas memprediksi kelas kualitas tomat sekaligus menentukan koordinat bounding box dalam satu langkah komputasi. 70 Konfigurasi pelatihan dengan batch size 8, learning rate 5e-5, weight decay 5e-5, serta pembagian data dengan rasio 70:15:15 untuk train, validasi, dan uji menunjukkan bahwa proses pelatihan telah dirancang secara cermat untuk menjaga stabilitas pembelajaran model hingga 120 epoch.

67 Dari sisi keterkaitan antar modul, sistem ini dibangun dengan arsitektur terdistribusi berbasis Web API yang menghubungkan modul akuisisi data, prapemrosesan, ekstraksi fitur dan klasifikasi, evaluasi internal, hingga aktuator pada perangkat ESP32. Aliran data yang dimulai dari sensor kamera di atas konveyor, diproses oleh server Python, dan diakhiri dengan pergerakan motor servo menunjukkan bahwa integrasi antara sisi perangkat lunak (software) dan perangkat keras (hardware) telah berjalan secara sinkron dan real-time. Hal ini menjadi dasar penting yang memungkinkan sistem dapat berfungsi sebagai pemilah otomatis berbasis kecerdasan buatan.

83 Hasil pengujian fungsional dengan metode Black Box Testing terhadap  
36 tujuh fungsi utama sistem menunjukkan bahwa seluruh fitur, mulai dari inisialisasi program, deteksi tomat, klasifikasi kelas fresh dan rotten, penampilan confidence score, integrasi aktuator, hingga tombol keluar, berhasil berjalan sesuai dengan hasil yang diharapkan tanpa ditemukan kegagalan fungsi. Antarmuka sistem yang menampilkan bounding box putus-putus, label kelas, serta nilai confidence score secara real-time turut membuktikan bahwa proses pelokalan objek dan klasifikasi kualitas tomat telah terintegrasi dengan baik antara sisi model deteksi dan sisi antarmuka pengguna. Keberhasilan pengujian pada kelas fresh dan rotten, yang masing-masing diikuti oleh respons aktuator motor servo secara tepat, juga mengindikasikan bahwa rangkaian proses dari deteksi visual hingga aksi mekanis telah berfungsi sebagaimana yang direncanakan pada tahap perancangan.

3 Pada pengujian non-fungsional, evaluasi kinerja sistem melalui grafik riwayat pelatihan menunjukkan tren penurunan loss yang stabil dan konvergen  
24 baik pada data pelatihan maupun data validasi, tanpa indikasi overfitting maupun underfitting yang signifikan. Hasil ini didukung oleh heatmap confusion matrix yang memperlihatkan dominasi nilai prediksi benar pada diagonal utama, serta nilai kesalahan klasifikasi yang relatif minim. Secara kuantitatif, performa model tercermin dari nilai akurasi total sebesar 90% pada 191 sampel data uji, dengan kelas fresh memperoleh precision 92% dan recall

86%, sementara kelas rotten memperoleh precision 87% dan recall 93%. Nilai f1-score yang seimbang, yaitu 89% untuk kelas fresh dan 90% untuk kelas rotten, serta macro average dan weighted average yang konsisten di angka 90%, menunjukkan bahwa model SSD MobileNetV2 yang dibangun memiliki performa yang andal dan seimbang dalam membedakan kedua kelas tomat tersebut.

Meskipun demikian, hasil pengujian skenario prototipe pada Tabel 4.3 menunjukkan bahwa performa model di lapangan sangat dipengaruhi oleh dua faktor utama, yaitu kecepatan konveyor dan kondisi pencahayaan. Pada kecepatan 80 RPM dengan pencahayaan cahaya ruangan maupun lampu LED, sistem mampu mencapai akurasi tinggi antara 90% hingga 100%. Namun, performa ini menurun drastis seiring peningkatan kecepatan konveyor, di mana pada kecepatan 240 RPM akurasi hanya berkisar 0% hingga 30%. Kondisi tanpa cahaya juga terbukti menjadi faktor paling kritis, karena model sama sekali tidak mampu mendeteksi objek tomat pada seluruh variasi kecepatan, sehingga menghasilkan akurasi 0%. Temuan ini mengindikasikan bahwa meskipun model memiliki performa internal yang sangat baik pada lingkungan pengujian terkontrol, kemampuan deteksi secara real-time pada kondisi fisik di lapangan masih sangat bergantung pada kecukupan intensitas cahaya dan kecepatan pergerakan objek yang sesuai dengan kapasitas pengolahan citra model.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa sistem deteksi dan pemilah tomat otomatis berbasis SSD MobileNetV2 berhasil diimplementasikan dan berfungsi dengan baik, baik dari sisi fungsional maupun performa model secara internal. Namun demikian, keterbatasan yang teridentifikasi pada pengujian non-fungsional, khususnya terkait sensitivitas terhadap kondisi pencahayaan rendah dan kecepatan konveyor yang tinggi, menjadi catatan penting yang perlu diperhatikan untuk pengembangan lebih lanjut. Penambahan sumber pencahayaan tambahan pada area deteksi serta penyesuaian kecepatan konveyor dengan kapasitas pemrosesan model dapat

menjadi langkah strategis untuk meningkatkan keandalan sistem ini ketika diterapkan secara nyata pada skala industri pertanian

## BAB V

### PENUTUP

#### A. Kesimpulan

Sistem sortir otomatis untuk klasifikasi kematangan buah tomat berhasil dirancang menggunakan arsitektur hybrid SSD-MobileNetV2, di mana MobileNetV2 berperan sebagai backbone ekstraksi fitur dan SSD berperan sebagai head detector. Hasil pengujian internal model menunjukkan bahwa arsitektur ini mampu mencapai akurasi sebesar 90% pada 191 sampel data uji, dengan nilai precision dan recall yang seimbang pada kelas fresh maupun rotten, sehingga rancangan sistem ini terbukti memenuhi tujuan untuk menghasilkan kecepatan komputasi yang ringan sekaligus akurasi klasifikasi yang tinggi.

Pengujian terhadap model SSD-MobileNetV2 menggunakan confusion matrix membuktikan bahwa kelayakan sistem dari segi kecepatan deteksi dan akurasi tinggi cukup baik untuk diimplementasikan secara real-time. Namun demikian, hasil pengujian non-fungsional pada skenario prototipe menunjukkan bahwa performa deteksi sangat dipengaruhi oleh kondisi pencahayaan dan kecepatan konveyor, di mana akurasi tertinggi hingga 100% diperoleh pada kecepatan 80 RPM dengan pencahayaan cukup, sementara akurasi menurun drastis hingga 0% pada kondisi tanpa cahaya maupun kecepatan konveyor yang terlalu tinggi (240 RPM). Hal ini menunjukkan bahwa kelayakan model untuk implementasi industri secara real-time masih memerlukan kondisi operasional yang terkendali, khususnya pada aspek pencahayaan dan kecepatan jalur konveyor.

Sistem sortir otomatis berbasis model SSD-MobileNetV2 berhasil diimplementasikan pada konveyor pemilah berbasis IoT secara real-time, yang dibuktikan melalui keberhasilan seluruh pengujian fungsional menggunakan metode Black Box Testing pada tujuh fungsi utama, mulai dari inisialisasi program, deteksi tomat, klasifikasi kelas fresh dan rotten, penampilan confidence score, hingga integrasi aktuator motor servo pada mikrokontroler ESP32. Keberhasilan ini menunjukkan bahwa aliran data dari sensor kamera,

proses inferensi model, hingga aksi mekanis penyortiran telah terintegrasi secara sinkron sesuai dengan rancangan sistem.

## **B. Saran**

Berdasarkan hasil penelitian dan keterbatasan yang ditemukan selama proses pengujian, Saran untuk pengembangan penelitian selanjutnya untuk menambahkan lampu LED pada area deteksi untuk menjaga stabilitas kontras piksel dalam kondisi cahaya minim, serta perluasan kuantitas dan variasi dataset citra tomat dari aspek sudut pengambilan gambar maupun keragaman varietas tomat demi memperkuat kemampuan generalisasi model di lingkungan lapangan yang dinami.