

# Implementation of YOLOv8 Algorithm for Web-Based Detection of Coffee Fruit Ripeness

Alfito Dwi Putra<sup>1</sup>, Guntur Eka Saputra<sup>2</sup>

<sup>1</sup> Informatika, Fakultas Teknologi Industri, Universitas Gunadarma, Depok, 16424, Indonesia

<sup>2</sup> Informatika, Fakultas Teknologi Industri, Universitas Gunadarma, Depok, 16424, Indonesia

## Informasi Artikel

Diterima : 25 April 2025  
Revisi : 11 Mei 2025  
Publikasi : 20 Juni 2025

## Kata Kunci:

Pertama : Pertanian Cerdas  
Kedua : *Computer Vision*  
Ketiga : YOLOv8  
Keempat : Streamlit  
Kelima : Kematangan Buah Kopi

## ABSTRAK

Penelitian ini membahas pemanfaatan teknologi *computer vision* dalam bidang *smart agriculture*, khususnya untuk mendeteksi tingkat kematangan buah kopi. Algoritma YOLOv8 digunakan untuk menghasilkan model deteksi yang diintegrasikan ke dalam aplikasi berbasis web menggunakan Streamlit *framework*. Python digunakan untuk mengimplementasikan YOLOv8 dan mendukung deteksi objek secara *real-time*. Pengembangan model menggunakan pendekatan CRISP-DM, sementara pengembangan aplikasi menggunakan metode *prototyping*. *Dataset* yang digunakan terdiri dari 100 gambar primer dari Kebun Raya Bogor dan 4547 gambar sekunder dari Roboflow, yang dibagi menjadi data latih 3253 gambar, data validasi 930 gambar, dan data uji 464 gambar. Hasil pelatihan model menunjukkan akurasi mAP50 sebesar 82,9%, dengan rincian kelas kering 90,2%, matang 76,2%, mentah 80,9%, dan setengah matang 84,3%, sehingga memenuhi target akurasi sebesar 80%. Aplikasi yang dikembangkan memiliki fitur deteksi kematangan buah kopi melalui *upload* gambar dan deteksi *real-time* menggunakan kamera. Berdasarkan uji usability dengan 16 responden menggunakan *System Usability Scale* (SUS), aplikasi memperoleh skor rata-rata 90, termasuk kategori "Excellent" dengan *grade A*. Hasil ini menunjukkan bahwa aplikasi mudah digunakan dan mampu memenuhi kebutuhan pengguna dalam mendeteksi kematangan buah kopi secara praktis dan akurat.

## ABSTRACT

This research focuses on the application of computer vision technology in smart agriculture, particularly for detecting the ripeness level of coffee cherries. The YOLOv8 algorithm was utilized to build a detection model, which was then integrated into a web-based application developed using Streamlit framework. Python was used to implement YOLOv8 and support real-time object detection. The model development process followed the CRISP-DM approach, while the application development adopted a prototyping method. The dataset consisted of 100 primary images collected from Kebun Raya Bogor and 4547 secondary images from Roboflow, divided into 3253 training images, 930 validation images, and 464 testing images. The model achieved an overall mAP50 accuracy of 82.9%, with class-wise accuracy of 90.2% for dry, 76.2% for ripe, 80.9% for unripe, and 84.3% for half-ripe coffee cherries, exceeding the success criteria of 80%. The developed application provides features for detecting coffee cherry ripeness through image uploads and real-time detection using a camera. Usability testing conducted with 16 respondents using the System Usability Scale (SUS) resulted in an average score of 90, classified as "Excellent" with a grade of A. This indicates that the application is highly usable and effectively supports users in detecting coffee cherry ripeness.

This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license



---

**\*Penulis Koresponden**Email: [guntur@staff.gunadarma.ac.id](mailto:guntur@staff.gunadarma.ac.id)

Cara sitasi IEEE::

A. D. Putra & G. E. Saputra "Implementation of YOLOv8 Algorithm for Web-Based Detection of Coffee Fruit Ripeness" *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 2, pp. 432-445, Juni 2025. doi: 10.30811/jaise.v5i2.6730

---

**1. PENDAHULUAN**

Peningkatan teknologi komunikasi dan informasi dalam pertanian, yang dikenal sebagai revolusi pertanian 4.0, telah mendorong inovasi di sektor ini, termasuk penerapan *artificial intelligence* (AI) dan *computer vision* [1]. Salah satu aplikasi AI yang menonjol adalah penggunaan algoritma *You Only Look Once* (YOLO) untuk deteksi objek, yang menawarkan kecepatan dan akurasi tinggi, sehingga cocok untuk mendukung *smart agriculture* [2]-[3]. Integrasi YOLOv8 ke dalam platform web berbasis Streamlit memungkinkan otomatisasi proses bisnis dan operasional serta memudahkan akses pengguna melalui berbagai perangkat [4]. Python, sebagai bahasa pemrograman yang sangat populer dan serbaguna [5], digunakan dalam penelitian ini untuk mengimplementasikan algoritma YOLOv8.

Meminum kopi telah menjadi bagian dari gaya hidup masyarakat perkotaan. Kedai kopi terkenal mudah ditemukan, terutama di pusat perbelanjaan seperti Excelso, Coffee Luwak, JCo Donuts, dan Starbucks [6]. Kopi merupakan komoditas penting bagi ekonomi Indonesia, dengan pemerintah berencana meningkatkan produksinya [7]. Dengan adanya aplikasi ini, diharapkan proses pemetikan kopi dapat dilakukan pada waktu yang tepat, sehingga kualitas biji kopi yang dihasilkan lebih optimal.

Deteksi kematangan buah kopi merupakan tahap krusial dalam produksi kopi berkualitas tinggi, di mana warna kulit buah kopi digunakan sebagai indikator kematangan. Berbagai penelitian sebelumnya telah dilakukan untuk mendeteksi dan mengklasifikasi kematangan buah serta penyakit tanaman menggunakan metode berbasis kecerdasan buatan. Penelitian pertama menggunakan algoritma Fuzzy C-Means untuk klasifikasi kematangan buah kopi dan menghasilkan 75% akurasi [8]. Penelitian kedua menerapkan metode KNN dan PCA dan memperoleh akurasi sebesar 97,77% [9]. Ketiga menggunakan YOLOv5 untuk mendeteksi kematangan tomat dengan F1-score 0,97 dan precision 1,00, meski terbatas pada variasi dataset [10].

Penelitian keempat menggunakan YOLOv3-tiny pada buah pepaya berbasis Android, namun dipengaruhi kondisi pencahayaan dan sudut kamera [11]. Kelima menggunakan CNN Inception V3 untuk klasifikasi kematangan kopi kuning dengan akurasi 92% [12]. Keenam menerapkan metode KNN dan mencapai akurasi 95% [13]. Ketujuh menggunakan YOLOv4 untuk identifikasi jenis tanaman mangga dan memperoleh akurasi 95%, namun dengan data yang terbatas [14].

Penelitian kedelapan mendeteksi hama dan penyakit pada daun cabai menggunakan YOLO dengan akurasi 73% [15]. Kesembilan mendeteksi lahan terdampak hama tikus dengan YOLOv5 dan rata-rata akurasi 85% [16]. Sementara itu, Penelitian terakhir menggunakan YOLOv7 untuk klasifikasi penyakit daun kopi dan memperoleh hasil evaluasi performa yang sangat baik, dan tanpa keterangan kekurangan [17]. Namun, penelitian ini mengimplementasikan algoritma YOLOv8 dalam sebuah prototipe berbasis web menggunakan Python dan Streamlit, yang diharapkan dapat membantu petani dalam menentukan kematangan buah kopi secara *real-time*. Kelebihan YOLOv8 terletak pada peningkatan arsitektur dibandingkan versi sebelumnya, seperti penggunaan pendekatan *anchor-free* yang menyederhanakan proses deteksi, serta efisiensi waktu inferensi yang lebih baik. Selain itu, model ini mampu menerima input gambar dengan berbagai resolusi, menjadikannya lebih fleksibel dalam implementasi di berbagai kondisi lapangan. Arsitektur YOLOv8 juga mengadopsi modul baru seperti C3 dan C2f, serta menyempurnakan jaringan *Feature Pyramid Network* (FPN) dan *Cross-Layer Connection* (CLC), yang secara signifikan meningkatkan akurasi deteksi objek pada berbagai skala [18].

Tujuan dari penelitian ini adalah untuk memanfaatkan algoritma *You Only Look Once* versi 8 (YOLOv8) dalam menghasilkan model yang mampu mendeteksi kematangan buah kopi, merancang dan mengembangkan aplikasi berbasis web untuk melakukan deteksi tersebut menggunakan Streamlit, serta mengintegrasikan model YOLOv8 ke dalam aplikasi berbasis web guna mempermudah proses deteksi kematangan buah kopi melalui platform Streamlit.

## 2. METODE

Penelitian ini menggunakan dua pendekatan utama, yaitu metode untuk pengembangan model *machine learning* dan metode untuk pengembangan aplikasi.

### 2.1 Metode Pengembangan Model *Machine Learning*

Metode pengembangan model yang digunakan dalam penulisan ilmiah ini menggunakan metode *Cross-Industry Standard Process for Data Mining* (CRISP-DM). Berikut tahap dari metode CRISP-DM yang dilakukan pada penulisan ilmiah ini.

#### 1. *Business Understanding*

Penelitian bertujuan mengembangkan model YOLOv8 untuk mendeteksi kematangan buah kopi dengan akurasi minimal 80%, guna mengurangi penilaian manual dan memberikan teknologi yang mudah diakses bagi petani.

#### 2. *Data Understanding*

Pengumpulan data merupakan langkah krusial dalam penelitian ini untuk memastikan variasi data yang memadai dan representatif dari kondisi sebenarnya. Data yang digunakan dalam penelitian ini berasal dari dua sumber: data sekunder dan data primer. Data sekunder diambil dari dataset Roboflow, yang menyediakan gambar buah kopi dalam berbagai tingkat kematangan.

#### 3. *Data Preparation*

Melakukan pra-pemrosesan dan augmentasi data untuk meningkatkan variasi, serta membagi data menjadi data latih, validasi, dan uji untuk pengembangan model.

#### 4. *Modeling*

Menerapkan model YOLOv8, melatihnya menggunakan data latih, dan memvalidasinya dengan data uji sesuai kebutuhan deteksi kematangan. Berikut ini arsitektur YOLOv8 yang digunakan, yang terdiri dari tiga komponen utama: *backbone*, *neck*, dan *head*. *Backbone* bertugas mengekstraksi fitur menggunakan *Feature Pyramid Network* (FPN), *neck* memperkuat representasi fitur melalui *Cross-Layer Connections* (CLC), dan *head* memprediksi *bounding box* serta klasifikasi objek [18]. Arsitektur YOLOv8 dapat dilihat pada Gambar 1.



## 2.2 Metode Pengembangan Aplikasi

Metode pengembangan aplikasi menggunakan metode *prototyping*. Mengumpulkan kebutuhan sistem adalah proses *prototyping*, yang melibatkan kerja sama pengembang dan pengguna untuk menentukan tujuan, fungsi, dan kebutuhan operasional sistem [19]. Berikut adalah tahap-tahap yang dilakukan dalam metode *prototyping*.

### 1. Identifikasi Kebutuhan

Mengumpulkan informasi terkait fitur yang diinginkan, antarmuka pengguna, dan kriteria keberhasilan aplikasi. Mengidentifikasi masalah yang dihadapi oleh petani dalam menentukan kematangan buah kopi.

### 2. Desain Cepat

Membuat representasi awal perangkat lunak yang mencakup aspek-aspek yang terlihat oleh pengguna, seperti input, pendekatan, dan *output*.

### 3. Pembuatan Prototipe

Mengembangkan versi awal perangkat lunak berdasarkan desain cepat untuk memberikan gambaran tentang bagaimana perangkat lunak akan bekerja. Prototipe ini mencakup fitur dasar seperti *upload image*, penggunaan kamera untuk *real-time detection*, dan antarmuka pengguna yang intuitif. Streamlit digunakan untuk mempercepat proses pembuatan prototipe.

### 4. Rekayasa Produk

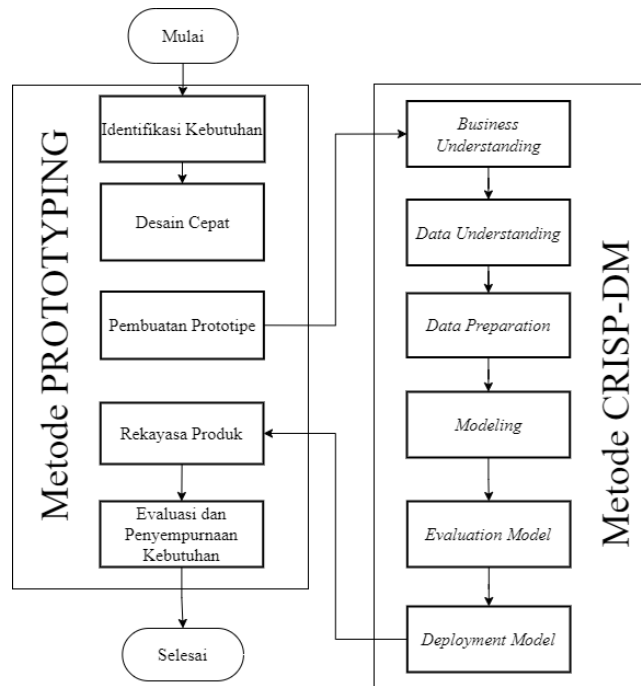
Mengembangkan perangkat lunak berdasarkan kebutuhan yang telah disempurnakan, memastikan semua persyaratan terpenuhi dan siap digunakan. Setelah prototipe diterima oleh pengguna, aplikasi di-*deploy* untuk memastikan performa optimal dan kompatibilitas dengan berbagai perangkat. Aplikasi yang telah di-*deploy* dimonitor secara rutin untuk memastikan performa dan akurasi deteksi tetap terjaga.

### 5. Evaluasi dan Penyempurnaan Kebutuhan

Prototipe dievaluasi oleh pelanggan dan perancang untuk mendapatkan umpan balik dan menyempurnakan kebutuhan perangkat lunak. Pengujian performa aplikasi dilakukan untuk mengukur tingkat akurasi, presisi, dan *recall* dari hasil deteksi.

## 3. HASIL DAN PEMBAHASAN

Penelitian ini bertujuan mengembangkan aplikasi web untuk mendeteksi kematangan buah kopi menggunakan teknologi *computer vision* dan algoritma YOLOv8, yang diterapkan untuk mendukung *smart agriculture*. *Framework* Streamlit digunakan untuk membangun aplikasi web interaktif yang dapat diakses dari berbagai perangkat [20]. Metode pengembangan model *machine learning* menggunakan pendekatan CRISP-DM, sementara pengembangan prototipe aplikasi dilakukan dengan pendekatan *prototyping*. Secara garis besar pengembangan aplikasi dilakukan dengan sejumlah tahapan utama seperti pada gambar 3 berikut.



Gambar 3. Alur Penelitian

Gambar 3 merupakan alur penelitian ini menggabungkan dua pendekatan: Metode *Prototyping* dan Metode CRISP-DM. Dalam metode *prototyping*, proses dimulai dengan "Identifikasi Kebutuhan," diikuti oleh "Desain Cepat" dan "Pembuatan Prototipe." Prototipe ini kemudian digunakan sebagai dasar dalam metode CRISP-DM.

Metode CRISP-DM dimulai dengan "*Business Understanding*," diikuti oleh "*Data Understanding*," "*Data Preparation*," dan "*Modeling*." Setelah model dibangun, tahap "*Evaluation Model*" dilakukan untuk memastikan kinerja model. Setelah model dievaluasi dan di-deploy, kembali ke metode *prototyping* dengan tahap "Rekayasa Produk" untuk penyempurnaan prototipe. Proses ini iteratif, memastikan produk akhir efektif dan siap diimplementasikan setelah melalui tahap "Evaluasi dan Penyempurnaan Kebutuhan." Metode *prototyping* menyediakan dasar bagi CRISP-DM untuk memenuhi kebutuhan bisnis dan teknis secara iteratif dan berkelanjutan.

### 3.1. Analisa Kebutuhan Sistem

Analisa kebutuhan sistem mencakup perangkat keras dan perangkat lunak yang diperlukan untuk membangun sistem.

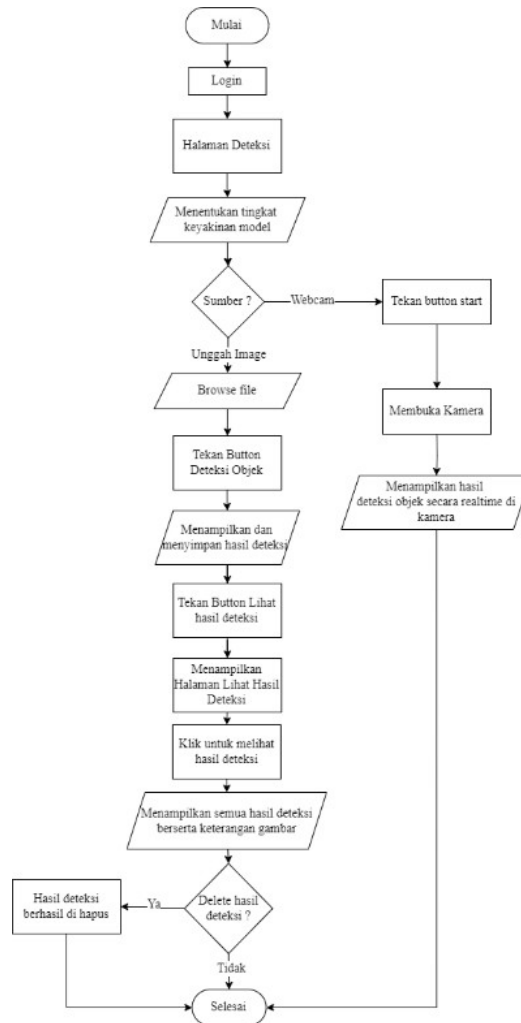
Tabel 1. Spesifikasi Perangkat Keras

Perangkat Keras	Keterangan
CPU	AMD Ryzen 5 5600U 2.3GHz
RAM	16GB DDR4
STORAGE	SSD 512GB
GPU	Nvidia GeForce RTX 3050

Tabel 2. Spesifikasi Perangkat Lunak

Perangkat Lunak	Keterangan
OS	Windows 11 Version 22H2 Home Single Language
IDE	Visual Studio Code 1.73.1
Bahasa Pemrograman	Python 3.10.2
Environment	Google Colaboratory
Framework	Streamlit versi 1.35.0

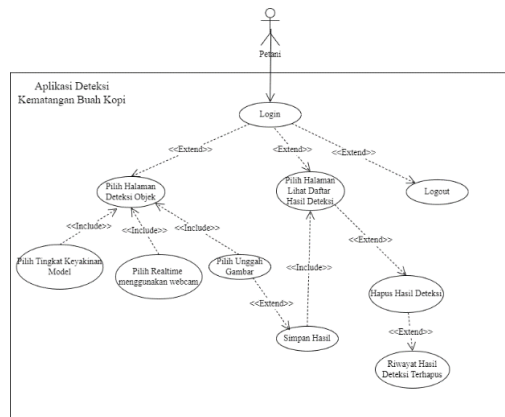
### 3.2. Flowchart Program



Gambar 4. Flowchart

Gambar 4 menunjukkan *flowchart* proses sistem deteksi objek untuk menentukan tingkat kematangan buah kopi. Proses dimulai saat pengguna *login*, lalu diarahkan ke halaman deteksi untuk menentukan tingkat keyakinan model dan memilih sumber *input*, yaitu *webcam* atau unggah gambar. Jika memilih *webcam*, sistem menampilkan hasil deteksi secara *real-time*. Sedangkan jika unggah gambar, pengguna memilih *file* dan menekan tombol "Deteksi Objek." Hasil deteksi kemudian ditampilkan dan disimpan, dengan opsi untuk melihat atau menghapus hasil. Proses berakhir saat pengguna selesai menggunakan sistem.

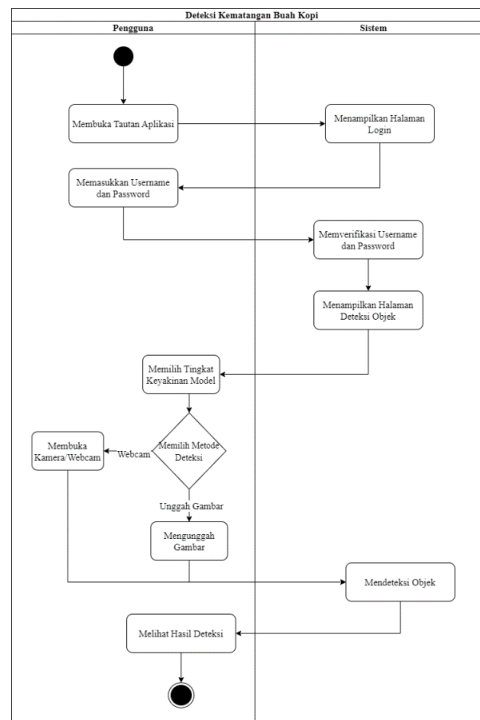
### 3.3. Use Case Diagram



Gambar 5. Use Case Diagram

Gambar 5 merupakan *use case diagram* yang menggambarkan interaksi antara petani dan sistem dalam aplikasi deteksi kematangan buah kopi. Petani *login* untuk mengakses halaman deteksi objek, menentukan tingkat keyakinan model, lalu memilih deteksi menggunakan *webcam* atau unggah gambar. Sistem memproses gambar dan menampilkan hasil deteksi yang dapat disimpan. Petani juga bisa melihat atau menghapus hasil deteksi sebelumnya, serta *logout* dari sistem. Relasi "*extend*" digunakan untuk akses halaman deteksi setelah *login*, sedangkan relasi "*include*" menunjukkan pemilihan tingkat keyakinan model sebagai bagian dari proses deteksi.

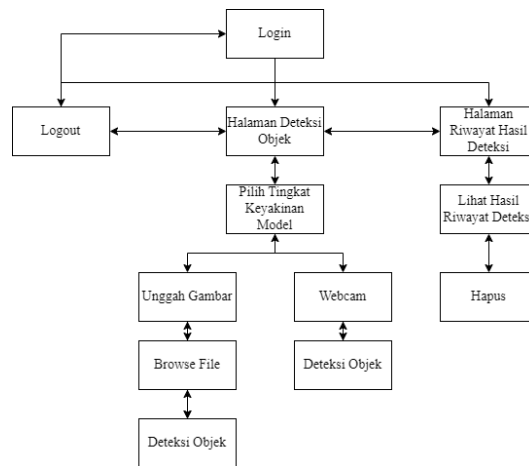
3.4. Activity Diagram



Gambar 6. Activity Diagram

Gambar 6 merupakan *activity diagram* yang menggambarkan proses deteksi objek pada aplikasi web. Pengguna memulai dengan membuka program dan *login* menggunakan *username* dan *password*. Setelah *login* berhasil, sistem menampilkan halaman deteksi objek. Pengguna kemudian memilih tingkat keyakinan model dan metode deteksi baik menggunakan *webcam* untuk deteksi *real-time* atau mengunggah gambar. Sistem mendeteksi objek dalam gambar yang diambil atau diunggah, dan setelah deteksi selesai, pengguna dapat melihat hasilnya.

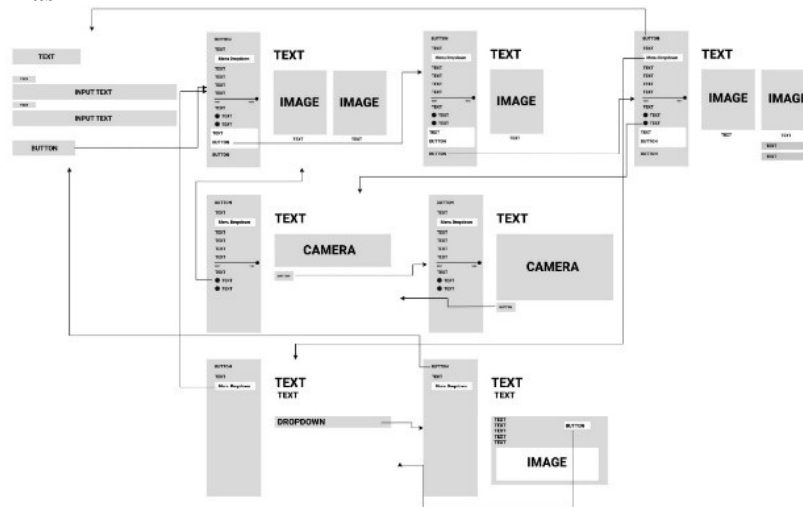
### 3.5. Struktur Navigasi Prototipe Aplikasi



Gambar 7. Struktur Navigasi

Gambar 7 merupakan struktur navigasi aplikasi web deteksi objek yang menunjukkan alur pengguna secara detail. Pengguna memulai dengan halaman *login*, kemudian diarahkan ke halaman deteksi objek, halaman riwayat hasil deteksi, dan *logout*. Di halaman deteksi objek, pengguna mengatur tingkat keyakinan model dan memilih sumber gambar—unggah gambar atau *webcam*. Hasil deteksi dapat dilihat di halaman riwayat hasil deteksi, di mana pengguna juga dapat menghapus hasil jika diinginkan. Pengguna dapat *logout* dari sistem kapan saja.

### 3.6. Storyboard Aplikasi



Gambar 8. Storyboard Aplikasi

Gambar 8 adalah *Storyboard* yang menggambarkan antarmuka pengguna dan alur interaksi dalam aplikasi. Setiap layar pada storyboard menunjukkan elemen-elemen penting seperti *input* teks, tombol, gambar, dan kamera yang digunakan untuk melakukan deteksi.

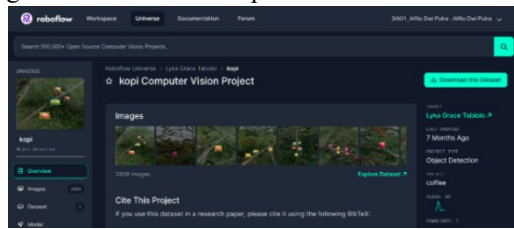
### 3.7. Pengembangan Model Menggunakan Metode CRISP-DM

#### 1. Business Understanding

Penelitian ini bertujuan mengembangkan model YOLOv8 untuk mendeteksi kematangan buah kopi, dengan fokus pada peningkatan akurasi deteksi dan mengurangi ketergantungan pada penilaian manual yang subjektif. Teknologi ini diharapkan mudah diakses oleh petani dan efektif dalam meningkatkan kualitas hasil panen. Kriteria keberhasilan proyek adalah mencapai akurasi deteksi minimal 80%.

#### 2. Data Understanding

Pengumpulan data adalah langkah penting dalam penelitian ini, yang mencakup dataset primer dan sekunder. Dataset primer diambil sebanyak 100 gambar dan data sekunder diambil dari dataset Roboflow yang menyediakan gambar buah kopi dalam berbagai tingkat kematangan. Link dataset sekunder adalah: <https://universe.roboflow.com/lyka-grace-tabiolo-9kido/kopi-cn77t/dataset/1>.



Gambar 9. Data Sekunder

Selain data sekunder, dilakukan juga pengumpulan data primer dengan mengambil foto langsung buah kopi dari kebun kopi. Proses pengumpulan data primer dilakukan dengan mengunjungi perkebunan kopi di Kebun Raya Bogor. Data primer dapat dilihat pada gambar 10.



Gambar 10. Data Primer

Selanjutnya dilakukan pemeriksaan kualitas data untuk memastikan bahwa gambar yang digunakan tidak buram atau rusak sehingga dapat mengganggu proses pelatihan model. Setiap gambar diperiksa untuk mendeteksi adanya cacat seperti buram, *overexposure*, *underexposure*, atau bagian gambar yang terpotong. Gambar-gambar yang tidak memenuhi standar kualitas dihapus sebelum digunakan dalam proses pelatihan model.

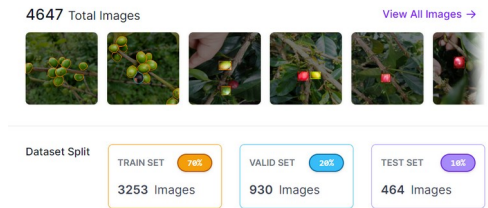
### 3. Data Preparation

Setelah data dikumpulkan, pelabelan dilakukan menggunakan platform Roboflow. Setiap gambar buah kopi diberi label berdasarkan empat kelas kematangan (mentah, setengah matang, matang, kering) dengan menggunakan *bounding box*. *Bounding box* membantu mengisolasi buah kopi dari latar belakang, memudahkan algoritma dalam menganalisis dan mengklasifikasikan kematangan berdasarkan warna kulitnya, sehingga model *machine learning* dapat belajar dengan lebih efektif dan akurat seperti pada gambar 11 [21].



Gambar 11. *Bounding Box*

Setelah itu dilakukan pembagian data menjadi tiga bagian yaitu data latih, data valid, dan data uji. Untuk data latih 70% dengan jumlah 3253 gambar, data valid 20% dengan jumlah 930 gambar, dan data uji dengan jumlah 464 gambar, maka jumlah gambar sebelum dilakukan augmentasi sebesar 4647 gambar.



Gambar 12. Pembagian Data

Setelah data dikumpulkan, dilakukan pra-pemrosesan untuk meningkatkan kualitas gambar, yaitu *resize* menjadi 640 x 640 piksel dan augmentasi data seperti *flip horizontal* dan *vertikal*, rotasi  $-10^\circ$  hingga  $+10^\circ$ , serta penyesuaian *brightness* 0% hingga +5%. Selanjutnya, data dibagi menjadi 87% data latih (9752 gambar), 8% data validasi (930 gambar), dan 4% data uji (464 gambar), sehingga total dataset berjumlah 11.146 gambar. Pembagian ini dilakukan untuk memastikan model dapat dilatih, divalidasi, dan diuji secara proporsional guna meningkatkan performa dan generalisasi model.

#### 4. Modeling

Selanjutnya, model YOLOv8 diterapkan untuk deteksi objek di Google Colaboratory. Proses dimulai dengan menginstal pustaka *ultralytics* dan *roboflow* serta mengimpor modul yang diperlukan. YOLOv8 dipilih untuk kemampuannya dalam deteksi *real-time* yang akurat [22]. Pengaturan model dilakukan dengan menginstal pustaka *roboflow*, mengimpor *dataset*, dan melatih model dengan parameter seperti *optimizer* AdamW, *learning rate* 0,00125, momentum 0,9, ukuran gambar 640, epoch 30, dan batch 16. Proses pelatihan model dilakukan sesuai pengaturan tersebut.

#### 5. Evaluation

Tahap evaluasi model dilakukan dengan menggunakan metrik akurasi, presisi, *recall*, mAP50, dan mAP50-95. Akurasi mengukur persentase prediksi yang benar, presisi mengukur akurasi prediksi positif, *recall* mengukur deteksi kasus positif, mAP50 menilai presisi pada *threshold* IoU 0,50, dan mAP50-95 memberikan gambaran kemampuan deteksi pada berbagai tingkat ketepatan [23].

```

Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11127132 parameters, 0 gradients, 28.4 GFLOPs
Class      Images  Instances  Box(P)      R      mAP50  mAP50-95: 100% 30/30 [00:10<00:00, 2.74it/s]
  all       930     5423      0.749      0.763   0.829   0.781
  kering    930     173       0.791      0.856   0.902   0.805
  matang    930     1899      0.684      0.691   0.762   0.739
  mentah    930     1603      0.774      0.717   0.809   0.778
  setengah matang 930     1748      0.747      0.787   0.843   0.802
Speed: 0.2ms preprocess, 4.4ms inference, 0.0ms loss, 3.1ms postprocess per image
  
```

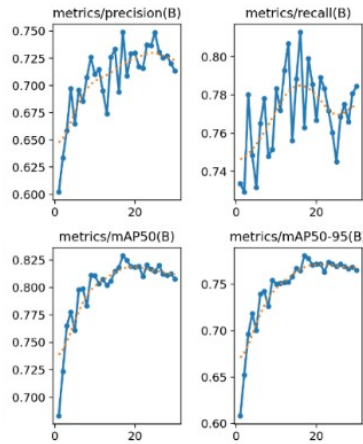
Gambar 13. Hasil Validasi Model

Gambar 13 model YOLOv8 menunjukkan performa sangat baik dalam mendeteksi kematangan buah kopi dengan akurasi mAP50 sebesar 82,9% dan mAP50-95 sebesar 78,1%. Performa deteksi per kelas adalah sebagai berikut: kelas kering (mAP50: 90,2%, mAP50-95: 80,5%), kelas matang (mAP50: 76,2%, mAP50-95: 73,9%), kelas mentah (mAP50: 80,9%, mAP50-95: 77,8%), dan kelas setengah matang (mAP50: 84,3%, mAP50-95: 80,2%). Kecepatan proses juga efisien, dengan waktu *preprocessing* 0,2ms, *inference* 4,4ms, dan *postprocess* 3,1ms per gambar. Matriks dapat dilihat pada gambar 14.



Gambar 14. Confusion Matrix

Gambar 14 Matriks menunjukkan jumlah prediksi benar dan salah per kelas kematangan. Angka diagonal mencerminkan prediksi yang benar: kering (160), matang (1571), mentah (1377), setengah matang (1494), dan background (926). Angka di luar diagonal menunjukkan kesalahan, seperti kering diprediksi sebagai matang (1) atau mentah sebagai matang (123). Model performa baik pada kelas matang dan setengah matang, namun kesalahan terjadi terutama dalam membedakan kelas kematangan dari *background*. Warna gelap pada matriks menandakan jumlah prediksi yang lebih tinggi.



Gambar 15. Grafik Evaluasi Model

Berdasarkan gambar 15 grafik menunjukkan tren peningkatan pada semua metrik evaluasi: presisi meningkat dari 0,602 ke 0,749, *recall* dari 0,729 ke 0,813, *mAP@0.5* dari 0,683 ke 0,829, dan *mAP@0.5:0.95* dari 0,608 ke 0,781. Hal ini menunjukkan perbaikan akurasi dan kemampuan deteksi model seiring bertambahnya epoch.

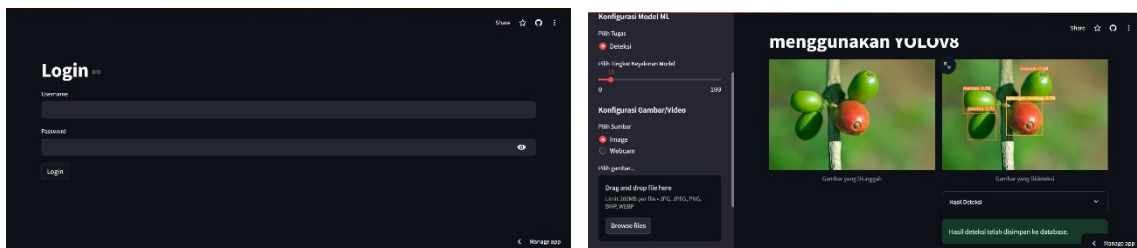
**6. Deployment Model**

Pada tahap deployment, model YOLOv8 yang telah dievaluasi digunakan untuk pengembangan aplikasi deteksi kematangan buah kopi. Model disimpan dalam format *best.pt*, file hasil pelatihan yang dioptimalkan, dan siap diintegrasikan ke dalam aplikasi untuk deteksi dengan akurasi tinggi dan efisiensi.

**3.8. Pengembangan Aplikasi Dan Penerapan Model**

Pengembangan aplikasi diawali dengan pembuatan prototipe untuk menggambarkan fungsi dasar aplikasi. Prototipe ini mencakup fitur utama seperti unggah gambar dari *file* dan deteksi *real-time* melalui *webcam*. Streamlit digunakan untuk mempercepat proses pengembangan. Tampilan tiap halaman aplikasi ditunjukkan berikut ini.

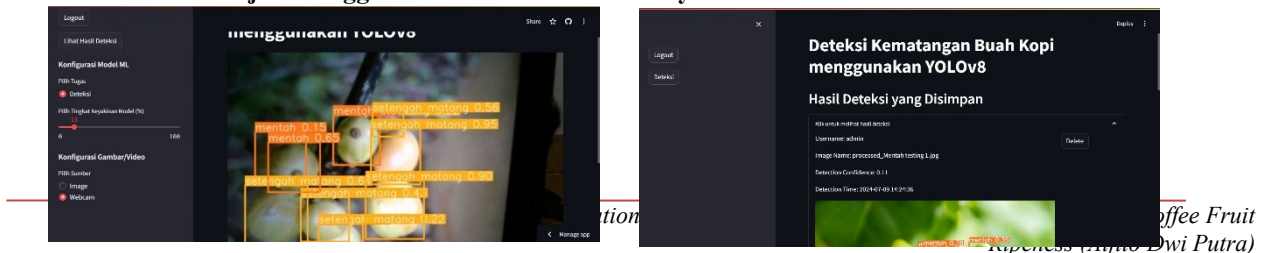
**Halaman Login dan Deteksi Objek Menggunakan Unggah Gambar**



(a) (b)

Gambar 16. Halaman Login (a) Halaman Deteksi Objek Dengan Unggah Gambar (b)

**Halaman Deteksi Objek Menggunakan Webcam dan Riwayat Hasil Deteksi**




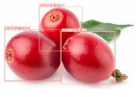

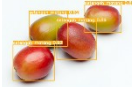
















coffee Fruit  
Ripeness (Agus Dwi Putra)

(a) (b)  
Gambar 17. Halaman Deteksi Objek Dengan Webcam (a) Halaman Riwayat Hasil Deteksi (b)

### 3.9. Evaluasi Dengan Uji Data

Evaluasi ini bertujuan untuk mengukur performa aplikasi deteksi kematangan buah kopi menggunakan data uji yang tidak digunakan dalam proses pelatihan model. Berikut tabel 3 yang merupakan hasil 10 (sepuluh) uji data menggunakan aplikasi Streamlit dan waktu respon secara aktual (*real-time*) untuk deteksi via webcam dalam *milliseconds* (ms).

Tabel 3. Uji Data dan Waktu Respon (*real-time*)

No	Gambar Original	Hasil Deteksi	Keterangan	Waktu Respon ( <i>real-time</i> ) (ms)
1			Berhasil deteksi semua objek	Waktu yang dibutuhkan model dalam mendeteksi secara <i>real-time</i> menggunakan <i>webcam</i> adalah 234,4 ms
2			Berhasil deteksi semua objek	Waktu yang dibutuhkan model dalam mendeteksi secara <i>real-time</i> menggunakan <i>webcam</i> adalah 205,0 ms
3			Berhasil deteksi semua	Waktu yang dibutuhkan model dalam mendeteksi secara <i>real-time</i> menggunakan <i>webcam</i> adalah 253,3 ms
4			Berhasil deteksi semua objek	Waktu yang dibutuhkan model dalam mendeteksi secara <i>real-time</i> menggunakan <i>webcam</i> adalah 219,4 ms
5			Berhasil deteksi semua objek	Waktu yang dibutuhkan model dalam mendeteksi secara <i>real-time</i> menggunakan <i>webcam</i> adalah 212,9 ms
6			Terdapat satu objek yang tidak terdeteksi	Waktu yang dibutuhkan model dalam mendeteksi secara <i>real-time</i> menggunakan <i>webcam</i> adalah 212,2 ms
7			Terdapat satu objek yang tidak terdeteksi	Waktu yang dibutuhkan model dalam mendeteksi secara <i>real-time</i> menggunakan <i>webcam</i> adalah 210,7 ms
8			Berhasil deteksi semua objek	Waktu yang dibutuhkan model dalam mendeteksi secara <i>real-time</i> menggunakan <i>webcam</i> adalah 219,3 ms
9			Terdapat satu objek yang tidak terdeteksi	Waktu yang dibutuhkan model dalam mendeteksi secara <i>real-time</i> menggunakan <i>webcam</i> adalah 255,8 ms
10			Berhasil deteksi semua objek dengan benar	Waktu yang dibutuhkan model dalam mendeteksi secara <i>real-time</i> menggunakan <i>webcam</i> adalah 206,3 ms

### 3.10. Evaluasi Usability Aplikasi

Evaluasi usability menggunakan *System Usability Scale* (SUS) untuk mengukur kemudahan penggunaan aplikasi. SUS terdiri dari 10 pertanyaan mengenai pengalaman pengguna, seperti kemudahan penggunaan [24], kebutuhan bantuan, dan konsistensi fitur. Pengguna menilai setiap pertanyaan pada skala 1 (Sangat Tidak Setuju) hingga 5 (Sangat Setuju), dan hasilnya digunakan untuk menilai dan meningkatkan usability aplikasi.

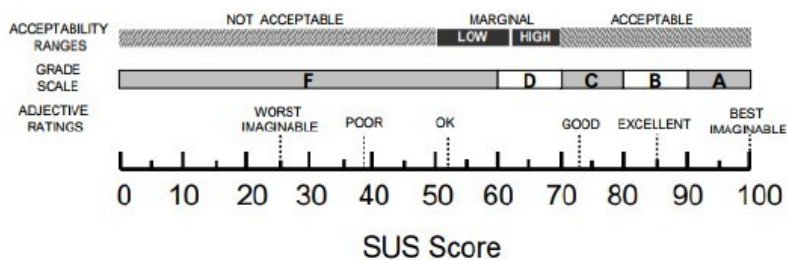
Table 1. Evaluasi Menggunakan SUS

No	Reponden	Skor Asli									
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Mario Martua A.	5	1	5	1	5	2	4	1	5	2
2	Satria Harya S.	4	1	5	2	4	1	4	2	5	2
3	Alif Ainul	4	1	5	1	4	1	5	2	5	2
4	Andika Satria P.	5	2	4	1	5	1	5	2	5	2
5	Bagas	5	2	4	1	5	3	5	1	5	2
6	Fanza	4	2	5	1	5	2	4	2	5	1
7	Ferdiansyah A.	5	2	5	1	4	1	5	2	5	1
8	Haikal Abizar	4	1	4	2	5	1	5	1	5	1
9	Jonathan Chandra	5	1	5	2	3	1	5	2	5	1
10	M. Fierza	5	1	4	2	5	2	4	1	5	2
11	Rasya Rachman	5	1	5	2	4	1	5	2	5	2
12	Salma Zahra	4	1	5	1	4	1	5	1	5	2
13	Thareq	4	1	5	1	4	1	5	3	5	2
14	Jovi Cahyono	4	1	5	2	5	2	5	1	5	1
15	Gede Ananta	5	2	5	1	5	2	4	1	5	1
16	I Putu Flandy	5	1	4	1	5	1	4	1	5	1

Tabel 4 menampilkan hasil evaluasi aplikasi deteksi kematangan buah kopi menggunakan *System Usability Scale* (SUS) dari 16 responden. SUS terdiri dari 10 pertanyaan, dengan pembagian pertanyaan positif dan negatif. Skor pertanyaan positif dikurangi 1, sedangkan pertanyaan negatif dikurangi dari 5. Total skor setiap responden kemudian dijumlahkan dan dikalikan 2,5 untuk memperoleh skor SUS akhir yang menggambarkan tingkat kegunaan aplikasi [25].

Table 2. Hasil Akhir Evaluasi SUS

Skor Hasil Hitung											Jumlah	Nilai (Jumlah x 2.5)
Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10			
4	4	4	4	4	3	3	4	4	3	37	93	
3	4	4	3	3	4	3	3	4	3	34	85	
3	4	4	4	3	4	4	3	4	3	36	90	
4	3	3	4	4	4	4	3	4	3	36	90	
4	3	3	4	4	2	4	4	4	3	35	88	
3	3	4	4	4	3	3	3	4	4	35	88	
4	3	4	4	3	4	4	3	4	4	37	93	
3	4	3	3	4	4	4	4	4	4	37	93	
4	4	4	3	2	4	4	3	4	4	36	90	
4	4	3	3	4	3	3	4	4	3	35	88	
4	4	4	3	3	4	4	3	4	3	36	90	
3	4	4	4	3	4	4	4	4	3	37	93	
3	4	4	4	3	4	4	2	4	3	35	88	
3	4	4	3	4	3	4	4	4	4	37	93	
4	3	4	4	4	3	3	4	4	4	37	93	
4	4	3	4	4	4	3	4	4	4	38	95	
Skor Rata-rata (Hasil Akhir)												90



Gambar 20. Skor Pada SUS [26]

Berdasarkan survei dari 16 responden, aplikasi deteksi kematangan buah kopi memperoleh rata-rata skor *System Usability Scale* (SUS) sebesar 90. Skor ini termasuk kategori *Excellent* dengan *grade A* [26], yang menunjukkan bahwa aplikasi dinilai sangat mudah digunakan. Hasil ini menjadi indikasi positif bahwa aplikasi berhasil memenuhi kebutuhan dan harapan pengguna.

#### 4. KESIMPULAN

Penelitian ini membuktikan bahwa algoritma YOLOv8 efektif dalam mendeteksi kematangan buah kopi, dengan akurasi mAP50 sebesar 82,9% yang melampaui target dan mendukung penerapan secara *real-time*. Aplikasi web berbasis Streamlit yang dikembangkan menyediakan antarmuka yang intuitif, memungkinkan pengguna untuk mengunggah gambar atau menggunakan kamera secara langsung (*real-time*) dalam proses deteksi. Integrasi model ke dalam aplikasi berhasil dilakukan dengan baik, menghasilkan solusi praktis yang bermanfaat bagi petani dan pelaku industri kopi. Aplikasi ini memperoleh skor rata-rata 90 pada pengujian *System Usability Scale* (SUS), yang menunjukkan tingkat kegunaan yang sangat memuaskan. Meskipun demikian, penelitian ini masih memiliki keterbatasan, seperti akurasi model yang masih dapat ditingkatkan dan variasi dataset yang terbatas.

#### UCAPAN TERIMAKASIH

Ucapan terima kasih disampaikan kepada Universitas Gunadarma atas dukungan yang diberikan selama pelaksanaan penelitian ini. Penghargaan juga diberikan kepada Program Studi Informatika Fakultas Teknologi Industri Universitas Gunadarma atas arahan dan masukan yang sangat berharga selama proses penelitian ini berlangsung.

#### REFERENSI

- [1] R. R. Rachmawati, "Smart Farming 4.0 untuk mewujudkan pertanian Indonesia maju, mandiri, dan modern," *Forum Penelit. Agro Ekon.*, vol. 38, no. 2, pp. 137–154, 2020.
- [2] O. E. Karlina and D. Indarti, "Pengenalan objek makanan cepat saji pada video dan real time webcam menggunakan metode you look only once (YOLO)," *J. Ilm. Inform. Komput.*, vol. 24, no. 3, pp. 199–208, 2020.
- [3] A. L. Rishika, C. Aishwarya, A. Sahithi, and M. Premchender, "Real-time vehicle detection and tracking using YOLO-based deep sort model: A computer vision application for traffic surveillance," *Turk. J. Comput. Math. Educ.*, vol. 14, no. 1, pp. 255–264, 2023.
- [4] M. Ibrahim and U. Latifa, "Penerapan algoritma YOLOV8 dalam deteksi waktu panen tanaman Pakcoy berbasis website," *JATI J. Mhs. Tek. Inform.*, vol. 7, no. 4, pp. 2489–2495, 2023.
- [5] T. M. Kadarina and M. H. I. Fajar, "Pengenalan bahasa pemrograman Python menggunakan aplikasi games untuk siswa/i di wilayah Kembangan Utara," *J. Abdi Masy.*, vol. 5, no. 1, pp. 11–16, 2019.
- [6] S. Solikatur, D. T. Kartono, and A. Demartoto, "Perilaku konsumsi kopi sebagai budaya masyarakat konsumsi (Studi fenomenologi pada peminum kopi di kedai kopi Kota Semarang)," *J. Anal. Sociol.*, vol. 4, no. 1, 2015.
- [7] R. A. Sirait, "Menguatkan jalan produksi perkebunan kopi melalui PISEW dan Dana Desa," *Budget Issue Brief Ind. Pembangunan*, vol. 2, no. 3, pp. 3–4, Mar. 2022.
- [8] M. R. Irfa'i, B. Fatkhurrozi, and I. Setyowati, "Klasifikasi tingkat kematangan buah kopi menggunakan algoritma Fuzzy C-Means," *Theta Omega J. Electr. Eng., Comput. Inform. Technol.*, vol. 2, no. 1, pp. 37–43, 2021.
- [9] S. Raysyah, V. Arinal, and D. I. Mulyana, "Klasifikasi tingkat kematangan buah kopi berdasarkan deteksi warna menggunakan metode KNN dan PCA," *J. Sist. Inform.*, pp. 88–95, 2021.
- [10] S. Aras, P. Tanra, and M. Bazhar, "Deteksi tingkat kematangan buah tomat menggunakan YOLOv5," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, pp. 623–628, 2024.
- [11] F. Agustina and M. Sukron, "Deteksi kematangan buah pepaya menggunakan algoritma YOLO berbasis Android," *J. Ilm. Infokam*, vol. 18, no. 2, pp. 70–78, 2022.
- [12] U. Ungkawa and G. Al Hakim, "Klasifikasi warna pada kematangan buah kopi kuning menggunakan metode CNN Inception V3," *ELKOMIKA J. Tek. Energi Elektrik, Tek. Telekomun. Tek. Elektron.*, vol. 11, no. 3, p. 731, 2023.
- [13] N. Amelia, M. Garonga, and J. Rusman, "Penerapan metode K-Nearest Neighbor (KNN) untuk klasifikasi kematangan buah kopi," *Indones. J. Comput. Sci.*, vol. 12, no. 2, 2023.
- [14] P. A. Widjaja and J. R. Leonesta, "Determining mango plant types using YOLOv4," *Formosa J. Sci. Technol.*, vol. 1, no. 8, pp. 1143–1150, 2022.
- [15] N. M. Yasen, S. Rifka, R. Vitria, Y. Yulindon, et al., "Pemanfaatan YOLO untuk deteksi hama dan penyakit pada daun cabai menggunakan metode deep learning," *Elektron J. Ilm.*, pp. 63–71, 2023.
- [16] K. A. Baihaqi and C. Zonyfar, "Deteksi lahan pertanian yang terdampak hama tikus menggunakan YOLOv5: Indonesia," *Syntax J. Inform.*, vol. 11, no. 2, pp. 1–11, 2022.
- [17] N. F. Hasan et al., "Deteksi dan klasifikasi penyakit pada daun kopi menggunakan YOLOv7," *J. Sisfokom Sist. Inform. Dan Komput.*, vol. 12, no. 1, pp. 30–35, 2023.
- [18] Y. Yanto, F. Aziz, dan I. Irmawati, "YOLO-V8: Peningkatan Algoritma untuk Deteksi Pemakaian Masker Wajah," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 3, pp. 1441–1448, 2023.
- [19] P. M. Ogedebe and B. P. Jacob, "Software prototyping: a strategy to use when user lacks data processing experience," *ARNP J. Syst. Softw.*, vol. 2, no. 6, pp. 219–224, 2012.
- [20] M. B. Julyano, S. A. Wibowo, and K. Usman, "Desain dan implementasi website Harvest Lens prediksi harga beras menggunakan framework Streamlit," *eProceedings of Engineering*, vol. 11, no. 4, pp. 2856–2860, 2024.
- [21] Q. Aini, N. Lutfiani, H. Kusumah, and M. S. Zahran, "Deteksi dan pengenalan objek dengan model machine learning: Model YOLO," *CESS (J. Comput. Eng. Syst. Sci.)*, vol. 6, no. 2, pp. 192–198, 2021.
- [22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.
- [23] S. Yohanandan, "mAP (mean Average Precision) might confuse you," *Towards Data Science*, vol. 9, 2020.
- [24] A. Sidik, "Penggunaan System Usability Scale (SUS) sebagai evaluasi website berita mobile," *Technologia J. Ilm.*, vol. 9, no. 2, pp. 83–88, 2018.
- [25] R. R. Andalas and M. A. Gustalika, "Evaluasi usability Google Meet pada pembelajaran daring menggunakan metode Cognitive Walkthrough (CW) dan System Usability Scale (SUS)," *JATI (J. Mahasiswa Tek. Inform.)*, vol. 6, no. 2, pp. 601–608, 2022.
- [26] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," *J. Usability Stud.*, vol. 4, pp. 114–123, 2009.