

## DETEKSI DAN KLASIFIKASI TINGKAT KEMATANGAN BUAH MANGGA HARUM MANIS MENGGUNAKAN YOU ONLY LOOK ONCE (YOLO) V8

Azizah Rismayanti, Reni Rahmadewi

Teknik Elektro, Universitas Singaperbangsa Karawang  
Jalan Raya HS. Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Indonesia  
2110631160036@student.unsika.ac.id

### ABSTRAK

Mangga Harum Manis adalah varietas mangga dari Probolinggo, Jawa Timur, dengan ciri bentuk jorong, sedikit berparuh, dan ujung meruncing. Penyortiran tingkat kematangan mangga Harum Manis di Probolinggo, Jawa Timur, masih dilakukan secara manual dengan metode pengamatan visual, yang dapat menyebabkan inkonsistensi dalam penentuan kualitas. Ketidakteraturan ini berpotensi memengaruhi standar distribusi dan nilai jual buah. Oleh karena itu, penelitian ini mengembangkan sistem otomatis berbasis YOLOv8 untuk mendeteksi dan mengklasifikasikan tingkat kematangan mangga harum manis menjadi tiga kategori: Belum Matang, Setengah Matang, dan Matang. Dataset terdiri dari 540 citra yang dianotasi di Roboflow, dengan pembagian 87% pelatihan, 8% validasi, dan 4% pengujian. Model dilatih selama 50 epoch, batch 16, dan ukuran gambar 500. Hasil menunjukkan model mampu mendeteksi tingkat kematangan dengan akurasi berbeda di tiap kategori. Nilai mAP tertinggi sebesar 82,2% pada kategori Belum Matang, sementara Setengah Matang memiliki performa terendah dengan 76,6%. Secara keseluruhan, Precision model mencapai 81,1%, Recall 81,5%, dan mAP 79,8%. Hasil ini mengindikasikan bahwa YOLOv8 memiliki potensi besar dalam deteksi objek di lingkungan nyata, meskipun diperlukan peningkatan pada kategori tertentu untuk meningkatkan akurasi secara keseluruhan.

**Kata kunci :** Mangga Harum Manis, Dataset, Algoritma YOLOv8, Deep Learning, Deteksi Objek, Roboflow.

### 1. PENDAHULUAN

Teknologi dalam era digital telah memberikan dampak signifikan pada berbagai sektor, termasuk pertanian. Sektor ini berperan penting dalam ekonomi Indonesia, salah satunya adalah buah-buahan tropis yang memiliki nilai ekonomi yang tinggi[1].

Salah satu jenis tanaman buah yang berasal dari India adalah mangga[2].

Mangga adalah salah satu buah tropis yang memiliki nilai ekonomi tinggi dan banyak ditanam di Indonesia, dengan produksi mencapai 2,8 juta ton pada tahun 2020[3].

Mangga, yang memiliki nama ilmiah *Mangifera indica* L., tergolong dalam famili Anacardiaceae[4] Mangga tumbuh dengan baik di dataran rendah dan hawa panas[5]

Namun, tanaman mangga dapat tumbuh hingga 600 mdpl. Kandungan nutrisi mangga yang baik untuk kesehatan, seperti vitamin A dan C, dapat meningkatkan daya tahan tubuh dan mengurangi risiko kerusakan mata dan sariawan[6]

Pohon mangga memiliki batang yang tegak dan cabang-cabang yang cukup kuat. Kulit batangnya tebal dan kasar, dengan banyak celah kecil serta bekas sisik-sisik tangkai daun[7].

Penanaman pohon mangga memberikan berbagai manfaat, baik dari segi ekonomi maupun kesehatan, sehingga banyak orang memilih untuk membudidayakannya, baik untuk kepentingan komersial maupun kebutuhan pribadi. Mangga merupakan buah yang mengalami perubahan tingkat kematangan dalam waktu relatif singkat. Pendistribusian buah mangga ke berbagai daerah menjadikan pentingnya klasifikasi mangga

berdasarkan tingkat kematangan. Menurut Harllee, terdapat enam tingkatan kematangan mangga yang dapat diidentifikasi berdasarkan warnanya, yaitu: *green* (mentah), *breakers*, *turning*, *yellow*, *light orange*, dan *orange*. Dengan demikian, warna buah mangga menjadi indikator penting dalam menentukan tingkat kematangan serta kualitasnya[8]

Beragam jenis mangga dapat ditemukan di masyarakat, seperti Mangga Gedong, Mangga Harum Manis, Mangga Cengkir, Mangga Golek, Mangga Gincu, dan jenis lainnya. Mangga (*Mangifera indica* L.) memiliki berbagai macam varietas, namun varietas arumanis menjadi salah satu yang paling banyak dibudidayakan. Mangga Harum Manis merupakan salah satu varietas mangga yang berasal dari wilayah Probolinggo, Jawa Timur. Ciri khas mangga ini adalah bentuk buahnya yang jorong, sedikit berparuh, dengan ujung yang meruncing.

Masalah yang sering muncul dalam proses pemilihan atau penyortiran buah mangga adalah ditemukannya beberapa mangga yang belum matang. Proses penyortiran sering kali bergantung pada penilaian subjektif manusia, yang dapat menyebabkan ketidakkonsistenan dan membutuhkan waktu yang lama. Faktor-faktor seperti keterbatasan penglihatan, kelelahan, perbedaan persepsi antara pengamat, serta kemungkinan kesalahan manusia akibat kelelahan atau kelalaian dapat memengaruhi akurasi hasil klasifikasi. Hal ini menunjukkan bahwa metode konvensional tidak selalu dapat memastikan semua mangga yang dipilih telah mencapai tingkat kematangan yang sesuai. Oleh karena itu, penerapan sistem berbasis kecerdasan buatan dapat membantu mengurangi variabilitas dan meningkatkan efisiensi dalam proses

tersebut. Penelitian ini bertujuan untuk merancang sebuah sistem yang mampu mendeteksi tingkat kematangan buah Mangga Harum Manis secara efektif.

Seiring dengan berkembangnya teknologi kecerdasan buatan, terutama dalam bidang deteksi objek, terdapat potensi besar untuk meningkatkan efisiensi proses tersebut. Kecerdasan buatan memungkinkan pengenalan dan klasifikasi objek dalam gambar secara lebih efektif. Hal ini merupakan bagian dari computer vision, yang berfokus pada kemampuan komputer untuk menganalisis dan menginterpretasikan gambar dengan cara yang menyerupai persepsi visual manusia[9]. Studi ini memfokuskan pada pengembangan sistem otomatis untuk menentukan tingkat kematangan buah mangga harum manis dengan mengimplementasikan Algoritma YOLO (You Only Look Once). Pemilihan YOLO sebagai metode deteksi objek berbasis bounding box didasarkan pada keunggulannya dalam memberikan hasil analisis yang simultan dan presisi. Karakteristik algoritma ini yang mampu melakukan pemrosesan dengan tingkat akurasi tinggi menjadikannya solusi yang optimal untuk sistem deteksi yang membutuhkan respons cepat dan ketepatan tinggi[10].

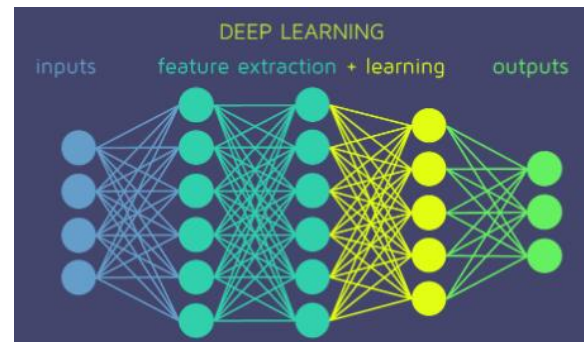
YOLOv8 merupakan iterasi terbaru dalam keluarga algoritma YOLO yang membawa peningkatan signifikan dibandingkan versi sebelumnya, seperti YOLOv5 dan YOLOv7. Berdasarkan penelitian terdahulu, YOLOv8 menunjukkan keunggulan dalam hal akurasi dan efisiensi deteksi objek jika dibandingkan dengan metode lain, seperti Faster R-CNN dan SSD (Single Shot MultiBox Detector). Studi yang dilakukan oleh (Terven et al., 2023) mengungkapkan bahwa YOLOv8 memiliki nilai mean Average Precision (mAP) yang lebih tinggi dibandingkan dengan Faster R-CNN, yang cenderung lebih lambat dalam pemrosesan gambar. Selain itu, YOLOv8 juga lebih unggul dalam kecepatan inferensi dibandingkan dengan SSD, yang masih menghadapi tantangan dalam mendeteksi objek berukuran kecil atau dalam kondisi pencahayaan yang kompleks. Oleh karena itu, pemilihan YOLOv8 dalam penelitian ini didasarkan pada kombinasi kecepatan, ketepatan, dan efisiensi yang optimal dalam pemrosesan gambar secara real-time.

## 2. TINJAUAN PUSTAKA

### 2.1. Deep Learning

Deep learning merupakan cabang dari *machine learning* yang mampu mempelajari pola dan melakukan komputasi secara mandiri dengan pendekatan menyerupai kerja otak manusia. Teknologi ini dirancang untuk menganalisis data secara terus-menerus guna mengambil keputusan yang lebih akurat. Untuk meningkatkan kemampuannya, deep learning menggunakan algoritma *Artificial Neural Network* (ANN), yang terinspirasi dari struktur jaringan biologis pada otak manusia. ANN

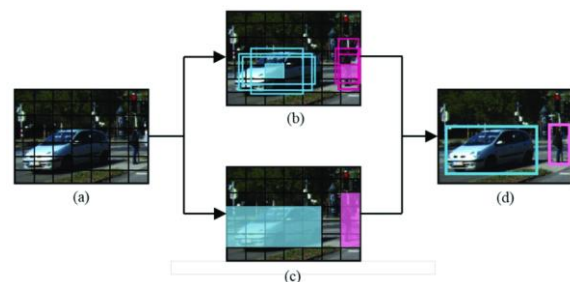
memanfaatkan lapisan-lapisan neuron buatan yang saling terhubung untuk mengolah informasi secara bertahap, memungkinkan pengambilan keputusan berdasarkan data yang kompleks[11]



Gambar 1. Deep Learning

Deep learning telah menjadi teknologi utama dalam berbagai aplikasi, termasuk deteksi objek, pengenalan pola, dan analisis data visual. Salah satu metode paling populer dalam deteksi objek adalah *You Only Look Once* (YOLO). YOLO berfungsi dengan menggunakan jaringan saraf konvolusional untuk menganalisis gambar dan mendeteksi objek-objek yang terdapat di dalamnya dengan hanya satu kali proses pemindaian. YOLO memproses gambar dalam satu kali pengamatan untuk mendeteksi dan mengklasifikasikan objek dengan pendekatan bounding boxes. Algoritma ini telah berkembang pesat, dari YOLO versi awal hingga YOLOv8, yang mengintegrasikan fitur seperti anchor-free detection dan prediksi multiskala untuk meningkatkan akurasi deteksi.

### 2.2. You Only Look Once (YOLO)



Gambar 2. Ilustrasi YOLO

Algoritma *You Only Look Once* (YOLO) adalah sebuah algoritma deteksi objek yang dikembangkan oleh Joseph Redmon, Santosh Divvala, Ross Girshick dan Ali Farhadi pada tahun 2016. Algoritma ini memiliki pendekatan yang berbeda dengan metode deteksi objek lainnya karena melakukan deteksi dan klasifikasi objek secara simultan dalam satu langkah. YOLO, yang merupakan singkatan dari "*You Only Look Once*," adalah metode canggih untuk deteksi objek yang memberikan hasil secara langsung. Pendekatan ini memandang masalah deteksi objek sebagai regresi tunggal, di mana data piksel gambar

digunakan untuk memprediksi kotak pembatas dan probabilitas kelas objek yang terdeteksi. Dengan memanfaatkan satu jaringan saraf, YOLO mampu mengidentifikasi dan mengenali objek secara langsung melalui prediksi kotak pembatas dan kelas objek secara simultan[12]

### 2.3. YOLO v8

YOLOv8, sebagai versi terbaru yang dikembangkan oleh Ultralytics, memperkenalkan berbagai fitur baru dan peningkatan yang signifikan dalam model deteksi objek. Versi ini mendukung berbagai tugas dalam kecerdasan buatan, seperti deteksi, segmentasi, estimasi pose, pelacakan, dan klasifikasi, memberikan fleksibilitas dalam berbagai aplikasi. Dengan integrasi kemajuan terbaru dalam pembelajaran mendalam dan visi komputer, YOLOv8 menjadi model deteksi objek dan segmentasi gambar real-time yang sangat efisien dan dapat diandalkan [13]

### 2.4. Python



Gambar 3. Python

Python pertama kali dikembangkan oleh Guido van Rossum pada akhir 1980-an dan dirilis pada 1991. Sejak itu, Python telah berkembang pesat melalui beberapa versi, dengan Python 2.0 dirilis pada tahun 2000 dan menjadi standar hingga 2020. Kemudian, Python 3.0 diperkenalkan pada 2008 dengan perbaikan sintaks dan fungsionalitas. Python terus berevolusi, dengan pembaruan rutin yang meningkatkan performa, kompatibilitas, dan kemudahan penggunaan. Kini, Python digunakan di berbagai bidang, termasuk web development, data science, dan machine learning[14].

Python adalah bahasa pemrograman tingkat tinggi, interpretatif, dan berorientasi objek. Python populer karena sintaksnya yang sederhana, menyerupai bahasa Inggris, sehingga mudah dipahami oleh pemula dan fleksibel untuk berbagai aplikasi. Bahasa ini mendukung pengembangan berbasis objek, prosedural, dan fungsional, membuatnya ideal untuk pemrograman lintas platform. Python digunakan dalam berbagai aplikasi seperti pengembangan web (Django, Flask), data science (Pandas, TensorFlow), pemrosesan gambar, dan aplikasi GUI. Keunggulan Python meliputi kemudahan belajar, portabilitas, dan dukungan pustaka yang luas. Namun, kelemahannya termasuk kecepatan eksekusi yang relatif lambat dan efisiensi memori yang rendah. Dengan popularitas

yang terus meningkat, Python menjadi salah satu bahasa utama dalam pengembangan modern[15].

### 2.5. Google Colab



Gambar 4. Google Colab

Google Colab merupakan platform komputasi awan yang memungkinkan pengguna untuk menjalankan kode Python langsung melalui browser web tanpa memerlukan instalasi perangkat lunak tambahan. Platform ini terintegrasi dengan Jupyter Notebooks dan menyediakan akses ke sumber daya pemrosesan canggih, seperti unit pemrosesan grafis (GPU) dan unit pemrosesan tensor (TPU), yang sangat berguna dalam bidang pengembangan perangkat lunak, analisis data, serta penerapan algoritma pembelajaran mesin[16].

Google Colab menyediakan kemudahan bagi peneliti dan pengembang untuk melakukan eksperimen serta menjalankan kode dalam lingkungan yang terintegrasi, yang mendukung berbagai kebutuhan pemrosesan data dan implementasi algoritma pembelajaran mesin secara efisien.

### 2.6. Data Preprocessing

Data yang telah dikumpulkan kemudian diproses untuk dianalisis melalui sejumlah tahap, termasuk pembersihan data untuk menghapus anomali atau data yang tidak relevan, penggabungan data dari berbagai sumber untuk memperoleh dataset yang lebih lengkap, transformasi data untuk menyesuaikan format atau struktur sesuai kebutuhan analisis, serta pemilihan atribut penting yang relevan. Langkah-langkah ini bertujuan untuk memastikan bahwa data memiliki kualitas dan format yang optimal untuk digunakan dalam pembangunan model analisis.

### 2.7. Pemodelan Data

Tahap pemodelan data adalah proses pelatihan algoritma yang bertujuan untuk memperoleh pola-pola dari data, yang selanjutnya digunakan untuk prediksi dan pengambilan keputusan[18].

Penelitian ini menggunakan pendekatan klasifikasi menggunakan teknik Deep Learning dengan Convolutional Neural Networks (CNN), yang dirancang untuk memproses input berupa gambar. CNN berperan dalam menentukan bobot signifikan pada berbagai aspek atau objek dalam gambar, serta memisahkannya satu sama lain[17].

Selain itu, YOLOv8 memanfaatkan CNN sebagai tulang punggung (backbone) untuk mengekstrak fitur dari gambar atau peta input. Sebagai model deteksi objek satu tahap (one-stage), YOLOv8 mampu

memproses dan mendeteksi objek dalam satu tahapan, sehingga menawarkan kecepatan yang lebih tinggi dibandingkan dengan model deteksi objek dua tahap (two-stage) konvensional[18].

## 2.8. Testing dan Evaluasi

Setelah tahap pelatihan dataset selesai, langkah selanjutnya adalah pengujian untuk menilai apakah model berhasil dalam mendeteksi tingkat kematangan buah. Dalam YOLOv8, beberapa metrik yang digunakan untuk evaluasi kinerja model meliputi Mean Average Precision (mAP), Precision, Recall, dan F1-Score[19].

*Mean Average Precision* (mAP) yang berfungsi untuk mengukur akurasi dalam deteksi dan klasifikasi objek, *Precision* yang digunakan untuk menilai akurasi deteksi yang benar, *Recall* yang mengukur kemampuan model dalam mendeteksi seluruh objek yang relevan, serta *F1-Score* yang merupakan nilai rata-rata harmonis antara *Precision* dan *Recall*.

### a. Precision

*Precision* mengukur akurasi dari prediksi positif yang dibuat oleh model. *Precision* dihitung sebagai rasio antara *true positives* (TP) dengan jumlah total prediksi positif yang benar dan salah, yaitu *true positives* dan *false positives*. *Precision* yang tinggi menunjukkan bahwa sebagian besar prediksi yang dibuat oleh model adalah benar, dengan sedikit *false positives*.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

### b. Recall

*Recall* merupakan kemampuan model untuk mengidentifikasi semua data positif yang sebenarnya. *Recall* mengukur kemampuan model untuk mengidentifikasi semua instance objek yang relevan. *Recall* dihitung sebagai rasio antara *true positives* dengan jumlah total objek yang relevan, yaitu *true positives* dan *false negatives*. *Recall* yang tinggi berarti model mampu mendeteksi sebagian besar objek yang sebenarnya ada dalam dataset, meskipun beberapa prediksi mungkin salah.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

### c. F1-Score

*F1-Score* merupakan matrik yang menggabungkan *precision* dan *recall* dalam satu nilai yang seimbang. Matrik ini sangat berguna ketika terdapat ketidakseimbangan antara *precision* dan *recall*, karena *F1-Score* menggabungkan kedua aspek tersebut dalam satu nilai.

### d. Mean Average Precision (mAP)

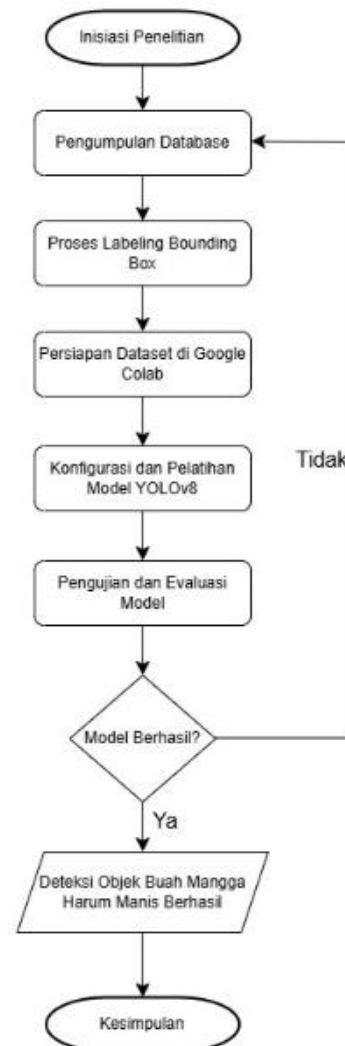
*Mean Average Precision* (mAP) merupakan presisi rata-rata yang sangat sering digunakan dalam mengukur keakuratan model deteksi. Matrik ini menghitung rata-rata dari skor *Average Precision* (AP) untuk beberapa kelas objek. mAP mempertimbangkan berbagai ambang batas *Intersection over Union* (IoU), dari 0,50 hingga

0,95, untuk menilai akurasi deteksi pada berbagai tingkat tumpang tindih antara kotak pembatas yang diprediksi dan kotak pembatas kebenaran dasar (ground truth).

$$mAP = \frac{1}{N} \sum_{k=1}^{k=n} AP_k \times 100\% \quad (3)$$

## 3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental dengan tahapan sistematis. Alur metodologi penelitian dirancang secara terstruktur untuk memastikan hasil deteksi yang optimal.



Gambar 5. Diagram Alir Penelitian

Gambar di atas menunjukkan alur proses penelitian yang terdiri dari beberapa tahapan untuk mendeteksi objek buah mangga Harum Manis menggunakan model YOLOv8. Proses dimulai dengan inisiasi penelitian, di mana tujuan dan kerangka penelitian ditentukan. Tahap berikutnya adalah pengumpulan database, yaitu pengumpulan data berupa gambar-gambar mangga yang akan digunakan untuk pelatihan model.

Setelah database terkumpul, dilakukan proses labeling bounding box, yaitu memberi tanda atau kotak

pada area gambar yang menunjukkan lokasi buah mangga. Data yang telah dilabeli kemudian dipersiapkan di Google Colab, platform komputasi berbasis cloud yang digunakan untuk memproses dan melatih model. Langkah selanjutnya adalah konfigurasi dan pelatihan model YOLOv8, di mana algoritma dioptimalkan menggunakan dataset yang telah disiapkan.

Hasil dari model diuji dan dievaluasi pada tahap pengujian dan evaluasi model untuk mengukur tingkat keberhasilannya. Jika model tidak berhasil, proses kembali ke langkah sebelumnya untuk perbaikan. Jika model berhasil, maka sistem dapat digunakan untuk deteksi objek buah mangga Harum Manis. Penelitian ini kemudian ditutup dengan tahap kesimpulan, di mana hasil akhir dirangkum untuk menjawab tujuan penelitian.

#### 4. HASIL DAN PEMBAHASAN

Pada tahap ini berhasil mengimplementasikan algoritma YOLOv8 untuk mendeteksi dan mengklasifikasikan tingkat kematangan buah mangga harum manis secara otomatis. Proses ini dilakukan dengan melatih model menggunakan dataset gambar mangga harum manis yang telah diberi anotasi berdasarkan tiga kategori kematangan, yaitu belum matang, setengah matang, dan matang. Pada dataset foto buah mangga harum manis yang dibuat di roboflow kemudian dilakukan latihan model YOLOv5 dengan website Google Colab. Proses pengujian terdiri dari tiga tahap, yaitu pengumpulan dataset, data preprocessing, pelatihan model YOLOv8, dan evaluasi kinerja model YOLOv8.

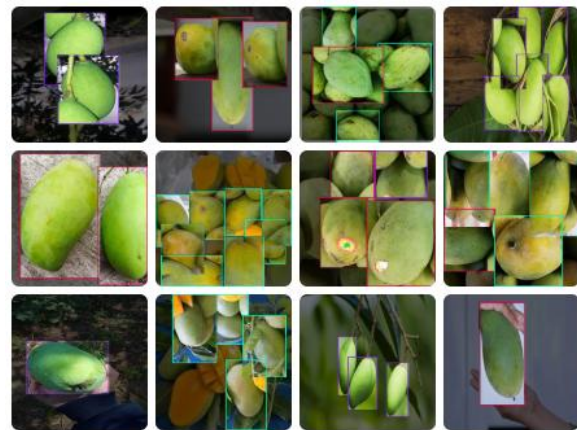
##### 4.1. Pengumpulan Data

Peneliti ini memanfaatkan dataset yang dibuat dari Roboflow. Dataset ini mencakup total 540 citra buah mangga harum manis yang telah dibagi menjadi tiga bagian. Sebanyak 87% dari dataset digunakan untuk proses pelatihan model (training), 8% dialokasikan untuk evaluasi kinerja model (validasi), dan 4% sisanya digunakan untuk pengujian (testing).

Tabel 1. Dataset Kematangan Buah Mangga

| Train Set  | Valid Set | Test Set  |
|------------|-----------|-----------|
| 472 Images | 45 Images | 23 Images |

Tabel 1. di atas menunjukkan pembagian dataset kematangan buah mangga harum manis ke dalam tiga bagian utama: train set, valid set, dan test set. Train set sebanyak 472 citra (87%) digunakan untuk pelatihan model, yang berfungsi untuk mengenali pola dalam data. Valid set sebanyak 45 citra (8%) dialokasikan untuk validasi, guna mengevaluasi kinerja model selama pelatihan dan mencegah overfitting. Sementara itu, test set sebanyak 23 citra (4%) digunakan untuk pengujian akhir model, dengan tujuan mengukur akurasi sebenarnya pada data yang belum pernah dilihat model sebelumnya.

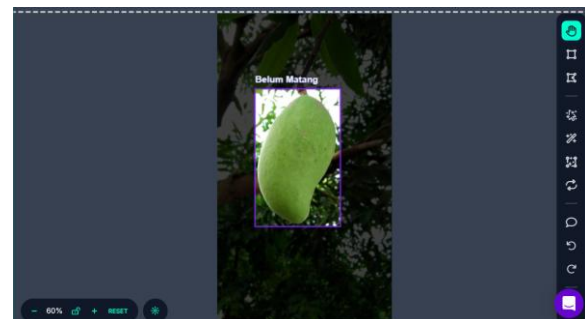


Gambar 6. Anotasi Buah Mangga Harum Manis dengan Bounding Box

Gambar di atas menampilkan kumpulan citra buah mangga harum manis yang telah dianotasi dan dikelompokkan berdasarkan tingkat kematangannya menggunakan bounding box. Terdapat tiga kategori utama yang digunakan untuk klasifikasi ini: Belum Matang, Setengah Matang, dan Matang. Pada tahap belum matang, kulit mangga berwarna hijau tua dengan kilau tipis, mencerminkan daging yang keras dan asam. Ketika mulai setengah matang, warna kulit berubah menjadi hijau kekuningan. Sementara itu, pada tahap matang sempurna, warna kulit mangga menjadi kuning cerah atau jingga dengan aroma harum khas.

##### 4.2. Data Preprocessing

Pada tahap ini, dilakukan pra-processing dataset yang mencakup pelabelan. Dalam Data Preprocessing akan dilakukan anotasi citra buah mangga harum manis dengan 3 label yaitu Belum Matang, Setengah Matang dan Matang, yang akan digunakan sebagai dataset. Dalam proses anotasi ini digunakan Platform Roboflow dengan 540 dataset. Anotasi dilakukan untuk memberikan kotak penanda dan label nama pada objek di suatu gambar yang biasa disebut dengan *bounding box*.



Gambar 7. Proses Label dengan Bounding Box

Gambar di atas menampilkan proses pelabelan dataset buah mangga harum manis menggunakan *bounding box* yang dilakukan melalui platform Roboflow. Dalam proses ini, setiap buah mangga harum manis diidentifikasi dengan kotak pembatas

untuk menentukan area objek yang relevan. Label yang diterapkan adalah "Belum Matang," yang mengacu pada tingkat kematangan mangga harum manis berdasarkan ciri-ciri visual seperti warna hijau dominan pada kulit buah. Proses pelabelan ini bertujuan untuk mempersiapkan dataset bagi pelatihan model deteksi objek berbasis *deep learning*, seperti YOLOv8. *Bounding box* digunakan untuk memberi model informasi posisi dan kategori objek dalam gambar, sehingga model dapat belajar mengenali dan membedakan kategori tersebut. Dalam gambar ini, setiap objek mangga harum manis di dalam gambar diberi beberapa *bounding box*, menunjukkan tingkat ketelitian proses pelabelan.

#### 4.3. Modeling

Dalam proses pelatihan model YOLOv8, dataset pelatihan dimanfaatkan untuk melatih model dalam mengenali dan mengklasifikasikan tingkat kematangan buah mangga harum manis. Sebelum melakukan pelatihan, dataset pada Roboflow harus di panggil terlebih dahulu.

#### 4.4. Persiapan Dataset

Dataset "detection-mango" dikelola melalui platform Roboflow untuk memastikan kompatibilitasnya dengan YOLOv8. Berikut adalah langkah-langkah dalam proses persiapannya:

- Menginstal Pustaka Robflow  
Instalasi pustaka Roboflow dilakukan menggunakan perintah `pip install roboflow` guna mempermudah pengelolaan dataset secara otomatis.
- Auntentikasi dan Pengunduhan Dataset  
Proses autentikasi dilakukan dengan menggunakan API key, yang menghubungkan sistem dengan workspace pengguna di Roboflow. Setelah itu, dataset diunduh dalam format yang sesuai dengan YOLOv8, termasuk file gambar serta anotasi dalam format .yaml.
- Menyusun Direktori Dataset  
Dataset yang telah diunduh kemudian diorganisir agar sesuai dengan standar struktur direktori YOLOv8. Folder utama mencakup `train/`, `valid/`, dan `test/`, masing-masing berisi gambar serta file anotasi dalam format YOLO.

Gambar 8 merupakan kode yang digunakan untuk mempersiapkan dataset yang diakses melalui platform Roboflow, sebagai bagian dari proses pelatihan model YOLOv8. Direktori khusus dibuat untuk menyimpan dataset, dan pustaka Roboflow diinstal untuk memungkinkan pengelolaan data secara

otomatis. Autentikasi dilakukan dengan menggunakan kunci API, yang kemudian menghubungkan program dengan ruang kerja pengguna yang berisi dataset "detection-mango" Versi dataset yang dipilih adalah versi pertama, yang selanjutnya diunduh dalam format kompatibel dengan YOLOv8, termasuk gambar dan anotasi.

```
mkdir -p {HOME}/datasets
cd {HOME}/datasets

pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="TLlyt54ujFQ58oRX68iE")
project = rf.workspace("azizah-rismayanti-lae2d").project("detection-mango")
version = project.version(1)
dataset = version.download("yolov8-obb")
```

Gambar 8. Install Dataset dengan memanggil Roboflow

#### 4.5. Konfigurasi Model YOLOv8

Sebelum memulai proses pelatihan, model YOLOv8 perlu dikonfigurasi agar dapat mendeteksi serta mengklasifikasikan tingkat kematangan mangga secara optimal. Beberapa parameter utama yang disesuaikan meliputi gambar dibawah ini.

```
!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=50 imgsz=500 plots=True
```

Gambar 9. Konfigurasi Model YOLOv8

Gambar diatas merupakan instruksi untuk melakukan pelatihan model YOLOv8n menggunakan pustaka Ultralytics YOLOv8.

- Model yang digunakan: YOLOv8n dipilih karena memiliki ukuran yang lebih ringan dan efisiensi tinggi untuk inferensi secara real-time.
- Jumlah epoch: Model dilatih selama 50 epoch agar dapat memahami pola dalam dataset secara optimal tanpa mengalami overfitting.
- Batch size: Sebanyak 16 batch digunakan untuk mempercepat proses pelatihan tanpa mengorbankan stabilitas model.
- Ukuran gambar: Resolusi gambar disesuaikan menjadi 500 piksel, agar kompatibel dengan dataset yang digunakan.

#### 4.6. Pelatihan Model YOLOv8

Kemudian selanjutnya melakukan pelatihan model menggunakan YOLOv8 dengan konfigurasi tertentu, seperti jumlah *epoch* sebanyak 50, *batch* 16, dan ukuran gambar 500 piksel serta dataset yang telah dianotasi sebelumnya. Perintah pelatihan model yang digunakan dapat dilihat pada tabel dan gambar berikut.

Tabel 2. Model Data yang dilatih

| Epoch | Box Loss | Cls Loss | Dfl Loss | Precision | Recall | mAp   |
|-------|----------|----------|----------|-----------|--------|-------|
| 1     | 1,328    | 2,326    | 1,564    | 0,853     | 0,789  | 0,813 |
| 2     | 1,095    | 1,597    | 1,337    | 0,793     | 0,801  | 0,798 |
| 3     | 1,12     | 1,593    | 1,37     | 0,762     | 0,814  | 0,776 |
| 4     | 1,124    | 1,519    | 1,379    | 0,887     | 0,782  | 0,816 |

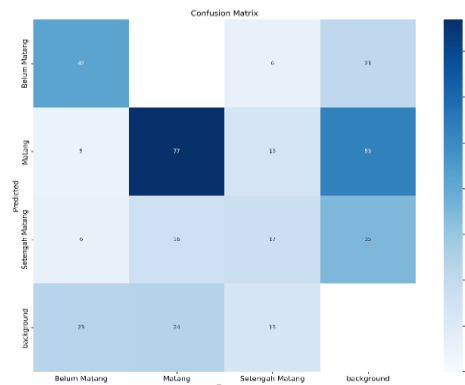
| Epoch | Box Loss | Cls Loss | Dfl Loss | Precision | Recall | mAp   |
|-------|----------|----------|----------|-----------|--------|-------|
| 5     | 1,064    | 1,446    | 1,338    | 0,765     | 0,800  | 0,802 |
| 6     | 1,005    | 1,384    | 1,311    | 0,824     | 0,853  | 0,788 |
| 7     | 1,003    | 1,38     | 1,305    | 0,795     | 0,766  | 0,804 |
| 8     | 1,013    | 1,316    | 1,301    | 0,803     | 0,771  | 0,791 |
| 9     | 0,9511   | 1,292    | 1,271    | 0,768     | 0,825  | 0,809 |
| 10    | 0,9073   | 1,244    | 1,241    | 0,831     | 0,804  | 0,785 |
| 11    | 0,9162   | 1,258    | 1,266    | 0,798     | 0,769  | 0,788 |
| 12    | 0,9038   | 1,198    | 1,249    | 0,815     | 0,826  | 0,836 |
| 13    | 0,8857   | 1,22     | 1,219    | 0,852     | 0,776  | 0,804 |
| 14    | 0,9187   | 1,194    | 1,25     | 0,865     | 0,840  | 0,819 |
| 15    | 0,8784   | 1,179    | 1,231    | 0,799     | 0,768  | 0,802 |
| 16    | 0,8602   | 1,137    | 1,222    | 0,800     | 0,813  | 0,812 |
| 17    | 0,8329   | 1,107    | 1,212    | 0,798     | 0,791  | 0,788 |
| 18    | 0,8306   | 1,062    | 1,201    | 0,856     | 0,762  | 0,795 |
| 19    | 0,8398   | 1,041    | 1,195    | 0,769     | 0,785  | 0,801 |
| 20    | 0,8104   | 1,047    | 1,193    | 0,852     | 0,770  | 0,776 |
| 21    | 0,7966   | 1,019    | 1,1      | 0,865     | 0,778  | 0,785 |
| 22    | 0,804    | 1,047    | 1,186    | 0,776     | 0,802  | 0,807 |
| 23    | 0,7922   | 1,018    | 1,173    | 0,818     | 0,846  | 0,788 |
| 24    | 0,75     | 0,976    | 1,156    | 0,798     | 0,861  | 0,808 |
| 25    | 0,7081   | 0,923    | 1,136    | 0,819     | 0,853  | 0,802 |
| 26    | 0,7764   | 0,969    | 1,165    | 0,807     | 0,834  | 0,793 |
| 27    | 0,7748   | 0,996    | 1,169    | 0,825     | 0,802  | 0,785 |
| 28    | 0,729    | 0,926    | 1,146    | 0,804     | 0,778  | 0,807 |
| 29    | 0,708    | 0,909    | 1,136    | 0,791     | 0,844  | 0,812 |
| 30    | 0,7025   | 0,888    | 1,126    | 0,842     | 0,803  | 0,785 |
| 31    | 0,701    | 0,873    | 1,126    | 0,807     | 0,859  | 0,799 |
| 32    | 0,6644   | 0,819    | 1,105    | 0,837     | 0,818  | 0,792 |
| 33    | 0,6535   | 0,803    | 1,097    | 0,865     | 0,800  | 0,793 |
| 34    | 0,6403   | 0,780    | 1,081    | 0,799     | 0,844  | 0,808 |
| 35    | 0,6707   | 0,792    | 1,11     | 0,813     | 0,814  | 0,784 |
| 36    | 0,6281   | 0,778    | 1,085    | 0,801     | 0,830  | 0,811 |
| 37    | 0,6453   | 0,801    | 1,099    | 0,784     | 0,785  | 0,793 |
| 38    | 0,6282   | 0,757    | 1,079    | 0,844     | 0,793  | 0,792 |
| 39    | 0,6269   | 0,739    | 1,08     | 0,812     | 0,809  | 0,785 |
| 40    | 0,6056   | 0,727    | 1,067    | 0,826     | 0,856  | 0,810 |
| 41    | 0,5445   | 0,698    | 1,036    | 0,784     | 0,802  | 0,799 |
| 42    | 0,532    | 0,609    | 1,023    | 0,802     | 0,830  | 0,802 |
| 43    | 0,5074   | 0,578    | 0,991    | 0,799     | 0,815  | 0,800 |
| 44    | 0,4771   | 0,529    | 0,978    | 0,846     | 0,832  | 0,799 |
| 45    | 0,4754   | 0,536    | 0,976    | 0,815     | 0,789  | 0,803 |
| 46    | 0,4599   | 0,500    | 0,968    | 0,828     | 0,847  | 0,792 |
| 47    | 0,4296   | 0,464    | 0,949    | 0,808     | 0,821  | 0,807 |
| 48    | 0,4361   | 0,455    | 0,944    | 0,841     | 0,804  | 0,811 |
| 49    | 0,4405   | 0,478    | 0,962    | 0,815     | 0,801  | 0,802 |
| 50    | 0,4269   | 0,452    | 0,956    | 0,859     | 0,812  | 0,809 |

Pada tabel 2. diatas menunjukkan perkembangan kinerja model secara bertahap. Penurunan nilai loss pada setiap epoch menunjukkan bahwa model semakin mampu mempelajari pola data dan mengoptimalkan prediksinya, baik dalam mendeteksi lokasi bounding box maupun mengklasifikasikan kategori objek. Nilai Precision, Recall, dan mAP juga menunjukkan peningkatan stabil sepanjang pelatihan. Precision rata-rata mencapai nilai 0,811, menandakan bahwa model mampu mengurangi jumlah false positives dalam prediksi. Recall rata-rata sebesar 0,815 mengindikasikan kemampuan model yang baik dalam

mendeteksi objek dari semua kategori. Selain itu, matrik mAP masing-masing berada pada rata-rata 0,798, yang menunjukkan bahwa model memiliki performa yang konsisten.

#### 4.7. Evaluasi dan Testing

Pada proses pelatihan dan pengujian pada setiap epoch, sistem menunjukkan perkembangan dan peningkatan kinerja secara bertahap. Hal ini juga dapat diamati melalui visualisasi hasil yang disajikan pada gambar 10.



Gambar 10. Confusion Matrix

Berdasarkan *confusion* matrix yang ditampilkan pada Gambar 9. Analisis hasil prediksi menunjukkan bahwa model mampu mengklasifikasikan tingkat kematangan buah mangga harum manis dengan tingkat akurasi yang berbeda di setiap kategori. Sebagai alat evaluasi yang digunakan untuk menilai kinerja algoritma klasifikasi, *confusion* matrix menggambarkan perbandingan antara hasil klasifikasi yang sebenarnya dengan hasil prediksi yang dihasilkan oleh model.

Dari hasil pengujian tersebut, diperoleh nilai *precision*, *mAP*, dan *recall* untuk setiap kelas dalam model deteksi tingkat kematangan buah mangga.

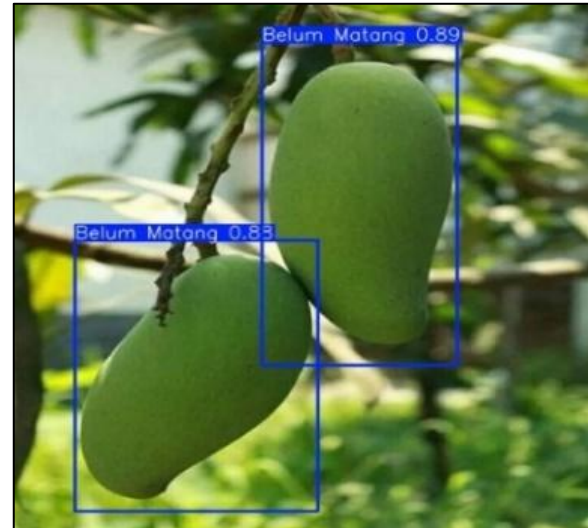
Tabel 3. Evaluasi Pengujian

| No | Class           | Precision | Recall | mAP50-95 |
|----|-----------------|-----------|--------|----------|
| 1  | All             | 0,811     | 0,815  | 0,798    |
| 2  | Belum Matang    | 0,835     | 0,856  | 0,822    |
| 3  | Setengah Matang | 0,793     | 0,765  | 0,766    |
| 4  | Matang          | 0,807     | 0,836  | 0,805    |

Berdasarkan tabel 3. di atas, model menunjukkan kinerja yang berbeda dalam mengklasifikasikan objek berdasarkan tingkat kematangan. Secara keseluruhan, model memiliki akurasi *Precision* sebesar 81,1%, *Recall* sebesar 81,5%, dan *mAP50-95* sebesar 79,8%. Hasil ini mencerminkan bahwa model secara umum mampu mengenali objek dengan baik, meskipun masih terdapat ruang untuk perbaikan pada beberapa kategori. Pada kelas Belum Matang, model menunjukkan performa terbaik dengan *Precision* sebesar 83,5%, *Recall* sebesar 85,6%, dan *mAP50-95* sebesar 82,2%. Hal ini mengindikasikan bahwa model memiliki kemampuan yang sangat baik dalam mendeteksi objek dengan tingkat kematangan ini secara konsisten. Sebaliknya, untuk kelas Setengah Matang, performa model relatif lebih rendah, dengan *Precision* sebesar 79,3%, *Recall* sebesar 76,5%, dan *mAP50-95* sebesar 76,6%. Kinerja yang lebih rendah ini menunjukkan bahwa model mengalami kesulitan dalam mengenali objek dengan tingkat kematangan setengah matang, kemungkinan akibat kompleksitas visual pada kategori ini. Sementara itu, pada kelas Matang, model memiliki performa yang cukup baik dengan *Precision* sebesar 80,7% , *Recall* sebesar

83,6%, dan *mAP50-95* sebesar 80,5%. Hasil ini menunjukkan bahwa model mampu mengidentifikasi objek matang dengan akurasi yang tinggi, meskipun tetap terdapat beberapa kesalahan prediksi.

Berikut merupakan hasil testing dari model tingkat kematangan buah pada mangga.



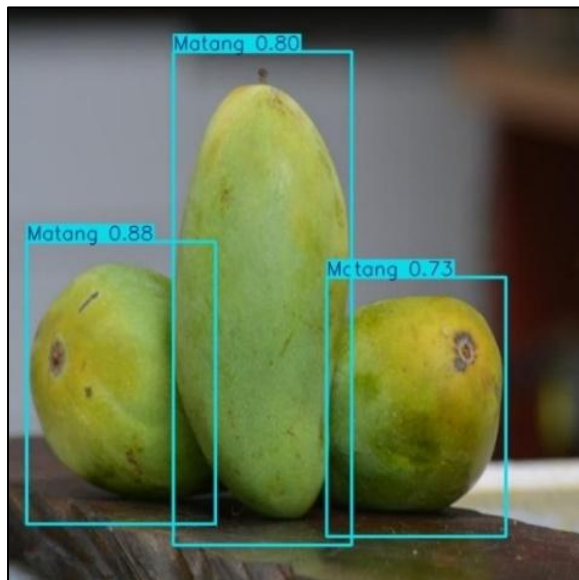
Gambar 11. Hasil Testing Buah Mangga Harum Manis "Belum Matang"

Hasil pengujian yang ditampilkan pada Gambar 10. menunjukkan penerapan model deteksi objek berbasis YOLOv8 yang telah dilatih menggunakan dataset dari Roboflow. menunjukkan bahwa kedua buah mangga harum manis yang terdeteksi masuk dalam kategori "Belum Matang" dengan tingkat keyakinan masing-masing sebesar 0,89 dan 0,83. Model berhasil mengidentifikasi area buah mangga harum manis dengan memberikan bounding box di sekitar objek yang relevan. Akurasi keyakinan yang tinggi (>80%) menandakan bahwa model memiliki performa yang baik dalam mengenali buah mangga harum manis yang belum matang pada kondisi pencahayaan dan latar belakang tertentu.



Gambar 12. Hasil Testing Buah Mangga Harum Manis "Setengah Matang"

Hasil pengujian klasifikasi kematangan buah mangga harum manis berdasarkan model deteksi objek yang telah diterapkan pada Gambar 11. Dalam hal ini, mangga harum manis diatas terdeteksi dalam kategori "Setengah Matang" dengan tingkat keyakinan (confidence score) sebesar 0.85. Informasi tersebut ditampilkan menggunakan bounding box yang mengelilingi objek buah mangga harum manis, serta label yang menjelaskan tingkat kematangan beserta nilainya.



Gambar 13. Hasil Testing Buah Mangga Harum Manis "Matang"

Hasil pengujian klasifikasi kematangan buah mangga harum manis berdasarkan model deteksi objek yang telah diterapkan pada Gambar 12. Pada gambar tersebut, model berhasil mendeteksi tiga objek berupa buah mangga Harum Manis dengan label "Matang" serta tingkat kepercayaan (confidence score) masing-masing sebesar 0,88, 0,80, dan 0,73. Setiap objek yang dikenali ditandai dengan bounding box berwarna biru, yang mengindikasikan area deteksi model pada citra. Keberhasilan model dalam mendeteksi dan memberikan klasifikasi dengan nilai confidence tinggi menunjukkan bahwa proses pelatihan telah menghasilkan model yang cukup andal untuk mengidentifikasi kondisi kematangan buah mangga Harum Manis dalam lingkungan nyata. Namun, perbedaan nilai confidence pada setiap deteksi dapat menjadi bahan evaluasi lebih lanjut untuk meningkatkan akurasi model, terutama dalam membedakan tingkat kematangan yang serupa.

## 5. KESIMPULAN DAN SARAN

Kesimpulan dari penelitian ini menunjukkan bahwa algoritma YOLOv8 berhasil diimplementasikan untuk mendeteksi dan mengklasifikasikan tingkat kematangan buah mangga harum manis secara otomatis berdasarkan kategori Belum Matang, Setengah Matang, dan Matang. Model

dilatih menggunakan dataset beranotasi yang dibuat melalui platform Roboflow dengan konfigurasi pelatihan sebanyak 50 epoch, batch size 16, dan image size 500 piksel. Dataset dibagi menjadi tiga bagian, yaitu 87% untuk pelatihan, 8% untuk validasi, dan 4% untuk pengujian. Hasil pengujian secara keseluruhan, model memiliki Precision rata-rata sebesar 81,1%, Recall sebesar 81,5%, dan mAP50-95 sebesar 79,8%. Performanya terbaik pada kategori Belum Matang dengan nilai mAP sebesar 82,2%, sementara kategori Setengah Matang menunjukkan performa terendah dengan mAP sebesar 76,6%. Kategori Matang memiliki performa cukup baik dengan mAP sebesar 80,5%. Keberhasilan model dalam mendeteksi buah mangga harum manis dengan tingkat keyakinan tinggi pada beberapa kategori, seperti Belum Matang dan Matang, mengindikasikan potensi aplikasinya untuk tugas deteksi objek di lingkungan nyata. Namun, kekurangannya adalah kinerja yang kurang konsisten pada kategori tertentu, yang dapat diatasi dengan penambahan variasi data pelatihan atau pengoptimalan parameter model.

Salah satu faktor yang dapat mempengaruhi hal ini adalah keterbatasan variasi dalam dataset pelatihan. Untuk meningkatkan akurasi deteksi, terutama pada kategori Setengah Matang, diperlukan perluasan dataset dengan mempertimbangkan berbagai kondisi pencahayaan, sudut pengambilan gambar, serta variasi warna buah mangga. Selain itu, penerapan teknik augmentasi seperti perubahan kontras, rotasi, dan flipping dapat membantu memperkaya dataset serta membuat model lebih adaptif terhadap berbagai kondisi nyata. Dengan langkah-langkah ini, diharapkan model dapat mencapai performa yang lebih stabil dan akurat pada semua kategori kematangan.

## DAFTAR PUSTAKA

- [1] "Detail Publikasi," *Pertanian.go.id*, 2021. <https://satudata.pertanian.go.id/details/publikasi/452>.
- [2] V. K. Sari, Halimatus Sa'diyah, Riza Yuli Rusdiana, Kacung Hariyono, and Sri Hartatik, "Keragaman Mangga ( *Mangifera indica* L.) di Jawa Timur Berdasarkan Karakter Morfologi: Studi Kasus di Kabupaten Jember," *Vegetalika*, vol. 13, no. 1, pp. 90–103, 2024, Accessed: Feb. 04, 2025. [Online]. Available: <https://jurnal.ugm.ac.id/jbp/article/view/87516/38643>.
- [3] Fajar Ramadhan, Muhammad Mubiar Ramadana, M. R. Hastin, and Ateng Supriatna, "Identifikasi Karakter Morfologi Mangga (*Mangifera Indica* L.) di Kecamatan Panyileukan, Jawa Barat, Indonesia," *Botani : Publikasi Ilmu Tanaman dan Agribisnis*, vol. 1, no. 3, pp. 27–37, 2024, doi: <https://doi.org/10.62951/botani.v1i3.79>.
- [4] Kurrota Ayyun, I. Rosyda, Nandiyyah Atikah, Shabrina Putri Arianti, and Arista Wahyu Ningsih, "Artikel Riwiew: Profil Studi Fitokimia

- Dan Aktivitas Farmakologi Buah Mangga (*Mangifera Indica L.*),” vol. 1, no. 2, pp. 60–68, Nov. 2023, doi: <https://doi.org/10.62379/jfkes.v1i2.346>.
- [5] FebyAulya AyuW, “Morfologi dan Anatomi Tanaman Mangga (*Mangifera indica L.*,” Feby Aulya Ayu Wandira, 2024. [https://www.academia.edu/122507632/Morfologi\\_dan\\_Anatomi\\_Tanaman\\_Mangga\\_Mangifera\\_indica\\_L](https://www.academia.edu/122507632/Morfologi_dan_Anatomi_Tanaman_Mangga_Mangifera_indica_L).
- [6] Admin Perpustakaan, “Buku Lapang Mangga 2021,” *Pertanian.go.id*, Oct. 27, 2024. <https://hortikultura.pertanian.go.id/buku-lapang-mangga-2021/>.
- [7] D. Hidayat, “Klasifikasi Jenis Mangga Berdasarkan Bentuk Dan Tekstur Daun Menggunakan Metode Convolutional Neural Network (CNN),” *INTECOMS Journal of Information Technology and Computer Science*, vol. 5, no. 1, pp. 98–103, Jun. 2022, doi: <https://doi.org/10.31539/intecom.v5i1.3401>.
- [8] Immardion Djara Rohi and Immardion Djara Rohi, “Klasifikasi Tingkat Kematangan Buah Mangga Harum Manis Berdasarkan Ciri Warna Menggunakan Metode Backpropagation Dengan Pembelajaran Levenberg Marquardt,” *Setiadi*, 2023, doi: <https://doi.org/2023-06-27>.
- [9] Dadang Iskandar Mulyana and M Ainur Rofik, “Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5,” *Jurnal Pendidikan Tambusai*, vol. 6, no. 3, pp. 13971–13982, Jul. 2022, doi: <https://doi.org/10.31004/jptam.v6i3.4825>.
- [10] F. Kamil, “Pengolahan Citra Digital Menggunakan Metode Yolo Untuk Mendeteksi Kualitas Dari Biji Kopi Berbasis Android,” *AI dan SPK : Jurnal Artificial Intelligent dan Sistem Penunjang Keputusan*, vol. 1, no. 1, pp. 120–125, 2023, Accessed: Feb. 04, 2025. [Online]. Available: <https://jurnalmahasiswa.com/index.php/aidanspk/article/view/240>.
- [11] Akram Kemal Dewantara, “Optimasi Pengambilan Keputusan dengan Neural Network: Menuju Era Keputusan Pintar,” *Indonesian Journal of Computer Science*, vol. 13, no. 5, Oct. 2024, doi: <https://doi.org/10.33022/ijcs.v13i5.4300>.
- [12] M. Hussain, “YOLOV5, YOLOV8 AND YOLOV10: THE GO-TO DETECTORS FOR REAL-TIME VISION,” 2024. Accessed: Feb. 12, 2025. [Online]. Available: <https://arxiv.org/pdf/2407.02988>
- [13] I. P. Sary, S. Andromeda, and E. U. Armin, “Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images,” *Ultima Computing : Jurnal Sistem Komputer*, pp. 8–13, Jun. 2023, doi: <https://doi.org/10.31937/sk.v15i1.3204>.
- [14] A. Rawat, “A Review on Python Programming,” *International Journal of Research in Engineering, Science and Management*, vol. 3, no. 12, pp. 8–11, 2020, Accessed: Feb. 04, 2025. [Online]. Available: <https://journal.ijresm.com/index.php/ijresm/article/view/395>.
- [15] J. Unpingco, “Python Programming for Data Analysis,” SpringerLink, 2021, doi: <https://doi.org/10.1007-978-3-030-68952-0>.
- [16] M. Kuroki, “Using Python and Google Colab to teach undergraduate microeconomic theory,” *International Review of Economics Education*, vol. 38, pp. 100225–100225, Sep. 2021, doi: <https://doi.org/10.1016/j.iree.2021.100225>.
- [17] N. J. Hayati, D. Singasatia, M. R. Muttaqin, T. Informatika, S. Tinggi, and T. Wastukencana, “OBJECT TRACKING MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE (YOLO)v8 UNTUK MENGHITUNG KENDARAAN,” *KOMPUTA : Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 2, 2023, [Online]. Available: <https://universe.roboflow.com/>.
- [18] S. Shubathra, P. C. D. Kalaivaani, and S. Santhoshkumar, “Clothing Image Recognition Based on Multiple Features Using Deep Neural Networks,” in 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 166–172. doi: <https://doi.org/10.1109/icesc48915.2020.9155959>.
- [19] R. Padilla, S. L. Netto, and Eduardo, “A Survey on Performance Metrics for Object-Detection Algorithms,” pp. 237–242, Jul. 2020, doi: <https://doi.org/10.1109/iwssip48289.2020.9145130>.
- [20] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: <https://doi.org/10.3390/make5040083>.