

## Sistem Deteksi Kualitas Cabai Rawit Menggunakan Metode YOLO (You Only Look Once)

Abdulloh Haidar Azzam Ash'shobir<sup>1\*</sup>, Kennyo Gendis Putri Harli<sup>2</sup>, Adisty Pramudita Putri Rudi<sup>3</sup>, Ilham Gusti Syah Putro<sup>4</sup>, Octavian Dava Putra Cahyono<sup>5</sup>

<sup>1, 2, 3, 4, 5</sup>Program Studi Teknik Informatika, Politeknik Negeri Jember, Indonesia

Alamat: Jl. Mastrip Po. Box 164, Kec. Sumbersari, Kab. Jember,  
Jawa Timur, Indonesia 68121

Korespondensi penulis: [abdullohhaidar002@gmail.com](mailto:abdullohhaidar002@gmail.com), [kennyogph18@gmail.com](mailto:kennyogph18@gmail.com),  
[adistypramudita03@gmail.com](mailto:adistypramudita03@gmail.com), [ilhamgustisyahputro@gmail.com](mailto:ilhamgustisyahputro@gmail.com), [octaviandavaputra@gmail.com](mailto:octaviandavaputra@gmail.com).

**Abstract.** This research focuses on developing a cayenne pepper quality detection system using the YOLO (You Only Look Once) V5 algorithm. The system is designed to address the limitations of manual post-harvest sorting by classifying cayenne peppers into three categories: "good" (ripe), "bad" (rotten or dry), and "raw" (green), based on their visual characteristics. A dataset consisting of 565 images was manually collected, labeled using Roboflow, and pre-processed through resizing and orientation standardization. Model training was conducted over 150 epochs, achieving high detection performance with a mean average precision (mAP) of 99.5%, precision of 99.6%, and recall of 99.9%. Real-time testing demonstrated the system's capability to detect and classify cayenne peppers with exceptional accuracy. This research is expected to enhance the efficiency and accuracy of the cayenne pepper sorting process, while paving the way for the application of YOLO-based systems to other agricultural commodities. Further research is recommended to expand the dataset and optimize model parameters for improved system performance.

**Keywords:** YOLO algorithm; cayenne pepper quality detection; cayenne pepper classification; postharvest sorting system.

**Abstrak.** Penelitian ini mengembangkan sistem deteksi kualitas cabai rawit menggunakan algoritme YOLO (You Only Look Once) V5. Sistem ini dirancang untuk mengatasi keterbatasan proses sortir pascapanen yang dilakukan secara manual, dengan mengelompokkan cabai ke dalam tiga kategori: "baik" (matang), "buruk" (busuk atau kering), dan "mentah" (hijau), berdasarkan karakteristik visualnya. Dataset yang terdiri dari 565 gambar dikumpulkan secara manual, dilabeli menggunakan Roboflow, dan diproses melalui langkah pra-pengolahan seperti pengubahan ukuran dan standarisasi orientasi. Pelatihan model dilakukan dengan 150 epoch, menghasilkan kinerja deteksi yang tinggi dengan nilai *mean Average Precision* (mAP) sebesar 99,5%, presisi 99,6%, dan *recall* 99,9%. Pengujian secara real-time menunjukkan kemampuan sistem dalam mendeteksi dan mengklasifikasikan cabai dengan tingkat akurasi yang sangat baik. Hasil penelitian ini diharapkan dapat meningkatkan efisiensi dan akurasi proses sortir cabai rawit, sekaligus membuka peluang untuk penerapan sistem berbasis YOLO pada komoditas pertanian lainnya. Penelitian lanjutan disarankan untuk memperluas cakupan dataset dan menyempurnakan parameter model guna meningkatkan performa sistem.

**Kata kunci:** algoritma YOLO; deteksi kualitas cabai; klasifikasi cabai; sistem sortir pascapanen.

### 1. LATAR BELAKANG

Cabai rawit merupakan salah satu komoditas hortikultura yang sangat diminati di Indonesia karena tingginya permintaan pasar terhadap rempah pedas ini. Keberhasilan budidaya cabai rawit didukung oleh kondisi iklim tropis Indonesia yang sangat ideal untuk pertumbuhan tanaman ini, sehingga menjadi faktor utama dalam meningkatkan hasil panen. Petani sering memilih cabai rawit sebagai sumber pendapatan yang menguntungkan karena harga jualnya yang tinggi, meskipun harga komoditas ini kerap fluktuatif dan mengalami kenaikan signifikan di pasaran. Kualitas cabai rawit menjadi penentu utama dalam memberikan cita rasa yang memuaskan bagi konsumen sekaligus memengaruhi nilai ekonomi dan kepuasan

pelanggan. Namun, proses sortir pasca panen masih dilakukan secara manual dengan pengamatan visual langsung, yang sering kali kurang akurat dalam menilai kualitas cabai. Hal ini menjadi tantangan yang harus diatasi untuk meningkatkan efisiensi, akurasi, dan mutu hasil panen cabai rawit di Indonesia.

Penerapan teknologi deteksi kualitas berbasis Deep Learning dengan algoritma YOLO menjadi solusi untuk memastikan cabai rawit memenuhi standar. YOLO memungkinkan deteksi objek real-time dengan klasifikasi cepat dan akurat, meningkatkan efisiensi dan akurasi dalam menilai karakteristik fisik cabai rawit seperti warna dan tekstur. Metode ini mengotomatisasi evaluasi kualitas, mengurangi ketergantungan pada pengamatan manual, dan mempercepat pengambilan keputusan dalam rantai produksi dan distribusi.

Sistem Deteksi Kualitas Cabai Rawit Menggunakan Metode YOLO dirancang untuk evaluasi kualitas cabai rawit secara real-time, meningkatkan efisiensi dan akurasi sortir pasca panen. Rumusan masalah mencakup implementasi YOLO, pengukuran efisiensi dan akurasi, serta deteksi cepat dan akurat. Tujuan sistem ini adalah menghasilkan deteksi otomatis berbasis YOLO, meningkatkan efisiensi dan akurasi sortir, serta menyediakan penilaian kualitas cepat dan real-time. Manfaat yang diharapkan termasuk efisiensi waktu, peningkatan akurasi, dan produktivitas petani, serta penerapan teknologi serupa di sektor pertanian lainnya.

## **2. KAJIAN TEORITIS**

### **Cabai Rawit**

Cabai rawit (*Capsicum frutescens L*) merupakan jenis sayuran yang sangat penting dan paling banyak dibudidayakan di seluruh Indonesia. Keunggulannya terletak pada proses pembudidayaannya yang tidak membutuhkan lahan besar (V. A. Raharjo et al.. 2021). Manfaat cabai rawit tidak hanya terbatas pada konsumsi langsung, tetapi juga dalam bentuk olahan. Umumnya, cabai rawit digunakan sebagai penyedap atau tambahan dalam masakan untuk meningkatkan kualitas dan cita rasa. Selain itu, cabai rawit juga menjadi bahan baku utama dalam industri makanan, seperti pabrik saus, bumbu penyedap berbahan dasar cabai, dan industri farmasi (A. L. Sardanti 2020).

### **Sistem Deteksi**

Sistem deteksi adalah suatu sistem atau teknologi yang dirancang untuk mengidentifikasi atau mendeteksi keberadaan atau karakteristik tertentu dalam suatu lingkungan atau data (objek). Penerapan sistem deteksi dapat melibatkan berbagai bidang, termasuk keamanan, pengolahan citra, pengenalan pola, dan banyak lagi (I. Salamah et al.. 2022).

## **Pengolahan Citra Digital**

Pengolahan citra digital adalah cabang ilmu yang mempelajari teknik pengolahan gambar, baik berupa foto maupun video, dengan menggunakan komputer secara digital. Melalui teknologi ini, kita dapat melakukan manipulasi presisi tinggi untuk meningkatkan kualitas gambar, mengurangi noise, dan melakukan transformasi lainnya. Pengolahan Citra Digital memiliki aplikasi luas di berbagai bidang seperti kedokteran, industri, sistem deteksi, keamanan, dan sebagainya yang memungkinkan analisis visual yang efisien dan mendalam (S. Ratna et al., 2020).

### ***Roboflow***

*Roboflow* merupakan alat konversi universal untuk kumpulan data computer vision (L.B Adrianto et al., 2021). *Roboflow* dapat mengimpor format anotasi apa pun dan mengekspor ke format anotasi lainnya, yang berarti pengguna dapat menghabiskan lebih banyak waktu untuk bereksperimen dan lebih sedikit waktu bergulat dengan skrip konversi satu kali untuk kumpulan data deteksi objek.

### ***Deep Learning***

Deep Learning adalah bagian dari pembelajaran mesin (machine learning) yang menggunakan algoritma pemodelan abstraksi tingkat tinggi pada data. Keunggulan utamanya adalah kemampuannya untuk secara otomatis mengekstraksi fitur-fitur tingkat tinggi dari data tanpa ekstraksi fitur manual (S. Mulyati et al., 2019). Deep Learning, sebagai bagian dari kecerdasan buatan, memiliki aplikasi praktis dalam pengenalan wajah, citra, suara, klasifikasi teks, dan tugas lainnya. Metode ini mengajarkan komputer untuk memproses data seperti otak manusia, menghasilkan wawasan dan prediksi akurat.

### ***You Only Look Once (YOLO)***

*You Only Look Once (YOLO)* adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara real-time. Sistem pendeteksian yang dilakukan adalah dengan menggunakan *repurpose classifier* atau *localizer* untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala (E. Purwawijaya et al., 2022).

## **Teknologi Dalam Deteksi Kualitas Cabai**

Deteksi kualitas cabai telah menjadi fokus penelitian untuk meningkatkan efisiensi dan akurasi pada tahap pasca panen. Menurut Luthfi et al. (2023), metode pengolahan citra berbasis warna menggunakan transformasi YCbCr telah digunakan untuk menentukan tingkat kematangan cabai merah. Pendekatan ini memungkinkan klasifikasi cabai menjadi kategori matang, setengah matang, atau sangat matang berdasarkan nilai *chrominance blue (Cb)* dan *chrominance red (Cr)* (Luthfi et al., 2023). Selain itu, penelitian menggunakan sensor warna,

seperti TCS34725, dalam kombinasi dengan algoritma *K-Nearest Neighbor (KNN)* menunjukkan akurasi hingga 90% dalam klasifikasi tingkat kematangan cabai (Priyani, 2021). Penelitian ini memperkuat pentingnya teknologi berbasis visual untuk mendukung proses sortir otomatis cabai di berbagai tahap distribusi.

### 3. METODE PENELITIAN

Bagian ini menjelaskan metode yang digunakan dalam penelitian untuk mencapai tujuan yang telah ditentukan. Penjelasan pada bagian ini mencakup beberapa aspek utama, yaitu tahapan penelitian, tahapan pengujian, flowchart, dan desain sistem. Setiap aspek diuraikan secara rinci untuk memberikan gambaran lengkap mengenai proses penelitian yang dilakukan.

#### Tahapan Penelitian

Tahapan-tahapan pada penelitian pengembangan sistem deteksi kualitas cabai rawit ini meliputi berbagai proses sebagai berikut.

#### 1. Pengumpulan Data (*Data Acquisition*)

Proses pengumpulan data untuk sistem deteksi kualitas cabai rawit yang sedang dibangun. Data berupa foto cabai dengan 3 kelas baik, busuk, dan mentah. Sumber *dataset* yang digunakan diambil sendiri secara manual dengan memfoto satu per satu.

#### 2. Pengolahan Data (*Reprocessing*)

Tahapan ini juga mempermudah dalam mendapatkan nilai ekstraksi ciri dari suatu objek yang akan di deteksi. Dalam penelitian ini *preprocessing* yang dilakukan adalah *resize* dan *auto orient* pada *Roboflow*.

#### 3. Labelling

Tahap *labelling* adalah langkah pembagian kelas pada setiap foto *dataset* melalui situs *website Roboflow*. Dalam penerapan sistem deteksi kualitas cabai rawit ini, tiga kelas yang diterapkan adalah *good*, *bad*, dan mentah.

#### 4. Data Split

Tahap pembagian data atau data split *dataset* yang sudah dilabel kemudian akan dibagi menjadi tiga kategori, yakni data train, data validasi, dan data uji. Hal ini bertujuan untuk memudahkan proses pelatihan serta mengurangi risiko *overfitting* dan *underfitting* pada *dataset*.

#### 5. Modelling dan Training Data

*Dataset* akan melalui proses pelatihan atau *training dataset* untuk meningkatkan akurasi sistem dan mencegah terjadinya kesalahan dalam pendeteksian objek pada *dataset*. Kemudian

menghasilkan model yang akan di terapkan pada hasil dan pengujian sistem deteksi kualitas cabai rawit.

## 6. Hasil dan Pengujian Sistem

Hasil yang diharapkan sistem dapat mendeteksi kualitas cabai rawit dengan menampilkan *bounding box* pada tiga kelas, yaitu *good*, *bad*, dan mentah, serta memberikan nilai akurasi yang tinggi. Pengujian sistem dilakukan secara manual langsung menggunakan kamera secara realtime. Dan metode *Confusion Matrix Multi Class* juga diterapkan untuk mengukur kemampuan sistem dalam mengklasifikasikan data terstruktur.

### Tahapan Penelitian

*Confusion Matrix Multi Class* adalah metode untuk mengukur kemampuan sistem dalam mengklasifikasikan data terstruktur dengan menghitung nilai aktual dibandingkan dengan jumlah seluruh data. Dengan menggunakan metode ini, nilai akurasi dapat dihasilkan. Alasan pemilihan *Confusion Matrix Multi Class* adalah karena data diklasifikasikan ke dalam beberapa kelas (Kurniawan, Martadinata, & Cahyo, 2023).

Metode ini memanfaatkan empat istilah untuk menunjukkan hasil klasifikasi, yaitu TP (*true positive*) yang merupakan jumlah data positif yang terklasifikasi benar oleh sistem, TN (*true negative*) yang menunjukkan jumlah data negatif yang terklasifikasi benar, FP (*false positive*) yang mengindikasikan jumlah data positif yang terklasifikasi salah, dan FN (*false negative*) yang mencerminkan jumlah data negatif yang terklasifikasi salah oleh sistem. Perhitungan akurasi dalam metode ini didasarkan pada jumlah nilai diagonal *Confusion Matrix* dibagi dengan total nilai prediksi.

Terdapat beberapa matriks dalam *Confusion Matrix* untuk menghitung berbagai kinerja guna mengevaluasi model yang telah dibuat.

#### 1. Accuracy

*Accuracy* mengukur seberapa akurat model dapat mengklasifikasikan data dengan benar. Ini merupakan rasio prediksi benar (positif dan negatif) terhadap keseluruhan data. Dengan kata lain, *accuracy* mencerminkan tingkat kesesuaian nilai prediksi dengan nilai aktual.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

#### 2. Precision (Positive Predictive Value)

*Precision* menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. *Precision* adalah rasio prediksi benar positif terhadap keseluruhan hasil yang diprediksi positif.

$$Precision = \frac{TP}{TP + FP}$$

### 3. Recall (True Positive Rate)

Recall menggambarkan keberhasilan model dalam menemukan kembali informasi tertentu. Recall adalah rasio prediksi benar positif terhadap keseluruhan data yang benar positif.

$$Recall = \frac{TP}{TP + FN}$$

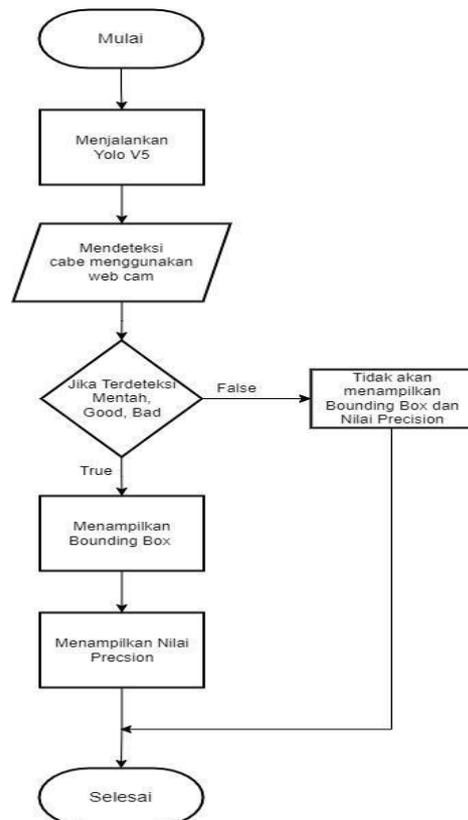
### 4. F-1 Score

F-1 Score menggambarkan perbandingan rata-rata antara precision dan recall yang dibobotkan. Ketika dataset memiliki jumlah data false negatif dan false positif yang mendekati, F1 Score menjadi metrik yang lebih sesuai dibandingkan dengan accuracy.

$$F - 1 \text{ Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

### Flowchart

Berikut merupakan flowchart dari “Sistem Deteksi Kualitas Cabai Rawit Menggunakan Metode YOLO (You Only Look Once)” yang telah dikembangkan.

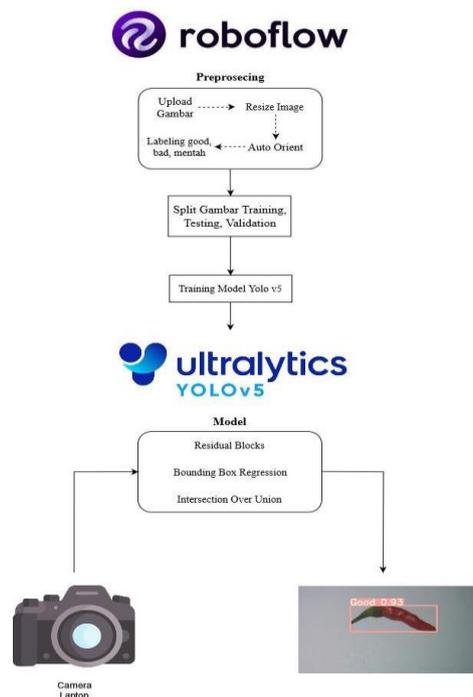


Gambar 1. Flowchart

Pada *flowchart* menggambarkan alur berjalannya program dalam pendeteksian kualitas cabai rawit secara *real-time* menggunakan metode *YOLO V5*. Ketika program dijalankan, proses dimulai dengan mengeksekusi *YOLO V5* untuk melakukan deteksi objek menggunakan kamera. Jika kamera berhasil mendeteksi cabai rawit, program selanjutnya mengklasifikasikan objek tersebut ke dalam salah satu dari tiga kelas, yaitu *good*, *bad*, atau mentah. Apabila kualitas cabai berhasil terdeteksi, hasilnya akan ditampilkan dalam bentuk *bounding box*, disertai dengan nilai *precision* yang mencerminkan akurasi deteksi objek cabai. Namun, jika kamera tidak berhasil mendeteksi adanya cabai, tidak akan ada *bounding box* maupun nilai *precision* yang ditampilkan. Setelah itu, program selesai dijalankan. Dengan demikian, *flowchart* ini menggambarkan langkah-langkah dari awal hingga akhir ketika program berjalan, memberikan pemahaman yang jelas tentang bagaimana sistem bekerja dalam mendeteksi dan mengklasifikasikan kualitas cabai rawit.

### Desain Sistem

Berikut merupakan *design system* dari “Sistem Deteksi Kualitas Cabai Rawit Menggunakan Metode *YOLO (You Only Look Once)*” yang telah dikembangkan.



**Gambar 2. Desain Sistem**

Pada *design system* diatas merepresentasikan terkait sistem yang telah dirancang meliputi tiga tahap untuk mencapai fungsionalitasnya.

## 1. Tahap *Input*

Tahap pertama, yang disebut sebagai tahap *input*, mencakup proses di mana objek, dalam hal ini cabai rawit, dideteksi dan diidentifikasi oleh kamera laptop atau *webcam*. Kamera berfungsi sebagai perangkat perekam visual yang mampu menangkap gambar dari lingkungan sekitarnya, memungkinkan sistem untuk mendapatkan data visual mengenai cabai rawit yang menjadi objek fokus. Tahap *input* ini menjadi langkah awal dalam perjalanan sistem, di mana informasi visual tentang cabai rawit diteruskan ke langkah-langkah selanjutnya.

## 2. Tahap *Proses*

Model ini menggunakan algoritma YOLO Versi 5 untuk deteksi objek *secara real-time* melalui kamera laptop. Proses deteksi melibatkan tiga komponen utama: *residual blocks*, *bounding box regression*, dan *intersection over union*. Dengan ketiganya, sistem dapat menganalisis dan mengidentifikasi objek secara langsung dan akurat, memungkinkan pengembangan lebih lanjut pada sistem deteksi objek *real-time*.

### a) *Residual Blocks*

Pada langkah ini, frame lengkap (gambar) dibagi menjadi kotak-kotak kecil atau grid. Semua grid berada di atas gambar asli dengan bentuk dan ukuran yang sama persis. Ide di balik pembagian ini adalah setiap kotak grid akan mendeteksi berbagai objek di dalamnya.

### b) *Bounding box Regression*

Setelah mendeteksi suatu objek dalam gambar, sebuah kotak pembatas digambar mengelilinginya. Kotak pembatas ini memiliki parameter seperti titik pusat, tinggi, lebar, dan kelas (jenis objek yang terdeteksi).

### c) *Intersection Over Union (IOU)*

IOU, singkatan dari *intersection over union*, digunakan untuk menghitung akurasi model kita. Hal ini dicapai dengan mengukur derajat interseksi dari dua kotak, yaitu kotak nilai sebenarnya (kotak merah pada gambar) dan kotak yang dihasilkan dari hasil kita (kotak biru pada gambar).

Sebelum menghasilkan model, terdapat serangkaian proses *preprocessing* yang harus dilewati, dan proses ini dilakukan melalui aplikasi *Roboflow*.

### a) *Upload Gambar*

Pertama-tama, pada tahap awal, dilakukan pengunggahan (*upload*) kumpulan gambar cabai rawit yang sudah terkumpul, yang kemudian menjadi *dataset* cabai rawit. *Dataset* ini merupakan kumpulan data mentah yang berasal dari gambar-gambar yang telah dikumpulkan sebelumnya. *Dataset* yang digunakan pada pengembangan sistem deteksi kualitas cabai rawit

ini merupakan merupakan kumpulan foto yang diambil secara langsung, bukan diunduh dari internet

**b) *Resize Gambar***

Pada tahap selanjutnya, *dataset* tersebut mengalami proses *resize image*, di mana ukuran gambar-gambar tersebut diubah menjadi 640 x 640 piksel.

**c) *Auto Orient***

Melalui tahap *auto orient*, gambar-gambar tersebut diatur orientasinya secara otomatis untuk memastikan konsistensi dalam pengolahan data.

**d) *Labeling Gambar***

Pada proses *labeling image* menjadi tiga kelas, yaitu "*bad*" (buruk), "*good*" (baik), dan "*mentah*" (belum matang). Label-label ini memungkinkan model untuk memahami dan membedakan karakteristik masing-masing kelas pada gambar. Klasifikasi kelas pada proses *labeling* sebagai berikut.

- *Good*: Cabai berwarna merah cerah, berkualitas baik, dan siap panen.
- *Bad*: Cabai busuk atau kering berwarna merah kehitaman.
- *Mentah*: Cabai belum matang berwarna kehijauan.

**e) *Split Gambar***

Setelah seluruh *dataset* dilabeli, dilakukan proses pemisahan gambar menjadi tiga bagian, yaitu gambar *training*, *testing*, dan *validation*. Data *Train Set* mencakup 63% dari total *dataset* (358 gambar), *Valid Set* mencakup 22% (127 gambar), dan *Test Set* mencakup 14% (80 gambar). Pembagian ini bertujuan untuk melatih, menguji, dan memvalidasi model dengan *dataset* yang beragam.

**f) *Training Model***

Tahap terakhir adalah *training model*, yang merupakan langkah kunci dalam pengembangan *Deep Learning*. Seluruh *dataset* yang telah diproses akan dijadikan input untuk melatih model, dengan jumlah *epoch* sebanyak 150. *Epoch* yang cukup besar ini memberikan kesempatan bagi model untuk mempelajari pola dan karakteristik yang kompleks dari *dataset* cabai rawit, sehingga dapat meningkatkan kemampuan deteksi dan klasifikasi pada sistem deteksi cabai rawit yang dikembangkan.

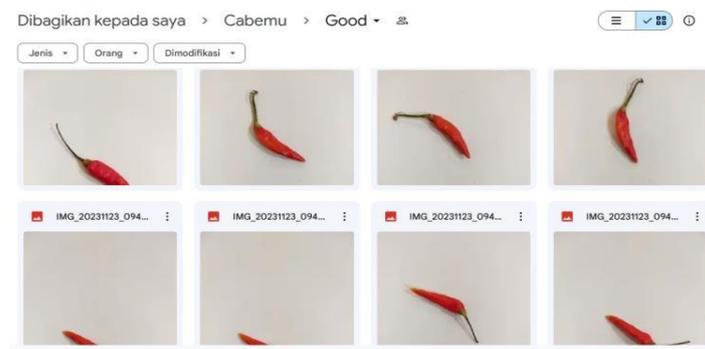
### 3. Output

Setelah diproses luaran dari sistem yaitu dapat mendeteksi cabai rawit dengan 3 kelas diantaranya *good* (matang sempurna), *bad* (busuk atau kering), dan *mentah* (belum matang). Sistem dapat mendeteksi cabai rawit dengan tingkat akurasi yang ditunjukkan melalui satuan angka dengan rentan 0,1 – 1 secara *real-time*.

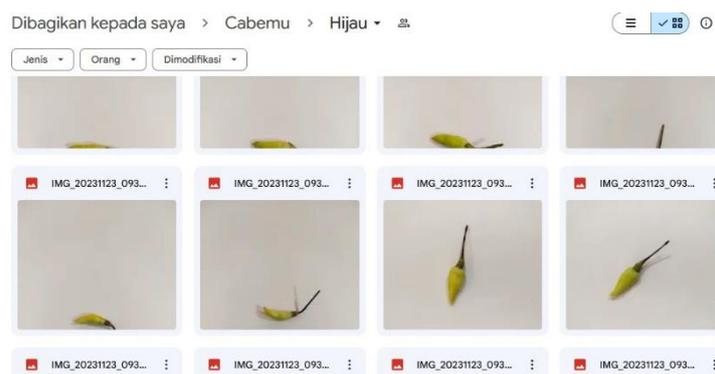
## 4. HASIL DAN PEMBAHASAN

### Pengambilan Dataset

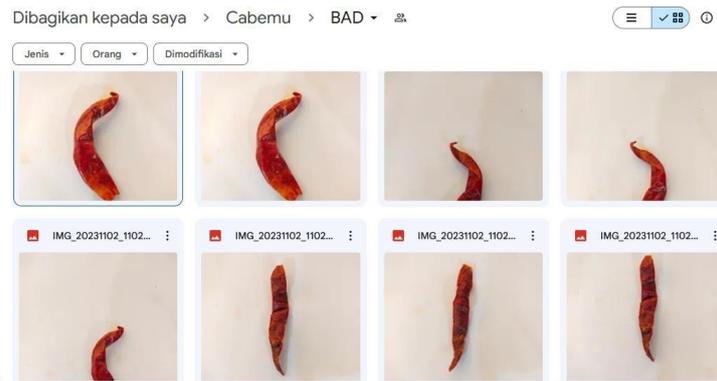
Sumber data untuk pengembangan sistem deteksi kualitas cabai rawit ini diperoleh secara langsung oleh tim kami. Pengumpulan data dilakukan dengan membagi cabai rawit menjadi tiga kelas utama, yaitu cabai rawit merah (*good*), cabai rawit busuk (*bad*), dan cabai rawit belum matang kehijauan (*mentah*). Informasi lebih lanjut mengenai data dapat ditemukan pada gambar di bawah ini.



**Gambar 3. Dataset Cabai Rawit Kelas Good**



**Gambar 4. Dataset Cabai Rawit Kelas Bad**



**Gambar 5. Dataset Cabai Rawit Kelas Mentah**

### **Pengolahan Data (*Reprocessing*)**

Tahap pengolahan data awal dilakukan agar dapat memperoleh hasil akurasi yang maksimal. Tahapan ini juga mempermudah dalam mendapatkan nilai ekstraksi ciri. Dalam penelitian ini *preprocessing* yang dilakukan adalah *resize* dan *auto orient*.



**Gambar 6. *Preprocessing* yang diterapkan pada *Roboflow***

#### **1. *Auto Orient***

*Auto Orient* pada *reprocessing dataset* di *Roboflow* diterapkan dengan tujuan untuk menstandarisasi orientasi citra dalam *dataset*, memastikan konsistensi orientasi untuk mempermudah pelatihan model dan mencegah model mempelajari orientasi sebagai fitur yang tidak relevan. Ini membantu meningkatkan interpretasi anotasi, kemudahan pelabelan, dan potensial peningkatan kinerja model pada tugas pengenalan objek.

#### **2. *Resize***

*Resize* pada *reprocessing dataset* di *Roboflow* dilakukan untuk menyesuaikan ukuran citra dalam *dataset*. Pentingnya proses ini terletak pada fakta bahwa citra dengan ukuran yang berbeda-beda dapat mengakibatkan kesulitan dalam pelatihan model, terutama jika model membutuhkan input dengan dimensi atau ukuran tertentu. Dengan menerapkan *resize* pada citra, ukurannya diubah agar seragam, memastikan model dapat menerima input dengan ukuran yang konsisten selama pelatihan. Hal ini tidak hanya meningkatkan efisiensi dan stabilitas proses pelatihan model, tetapi juga memungkinkan model untuk beroperasi dengan baik pada data yang memiliki variasi ukuran. Pada *dataset* cabai rawit ini, setiap foto *resize* menjadi ukuran  $640 \times 640$  piksel untuk memastikan konsistensi dan keseragaman dalam input model.

## Labelling

Proses *labelling* adalah langkah pembagian kelas pada setiap foto melalui situs *website Roboflow*. Dalam penerapan sistem deteksi kualitas cabai rawit ini, tiga kelas yang diterapkan adalah *good*, *bad*, dan *mentah*. Gambar di bawah ini memperlihatkan foto cabai rawit dalam tiga kondisi tersebut.



Gambar 7. Label *Good*



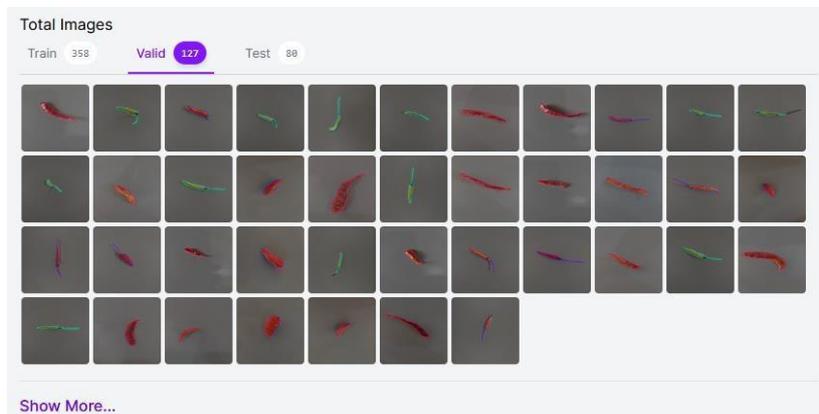
Gambar 8. Label *Bad*



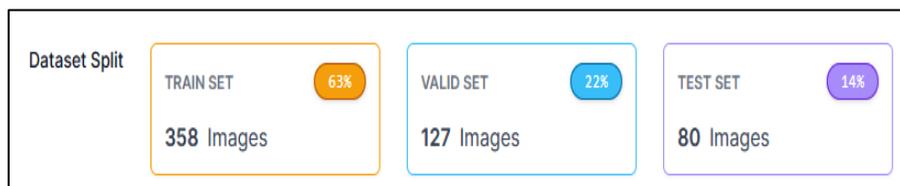
Gambar 9. Label *Mentah*

## Data Split

Proses split data dilakukan pada *website Roboflow* setelah *dataset* diubah ukurannya dan diberi label. Selanjutnya, data dibagi menjadi tiga kategori, yakni 63% data train, 22% data validasi, dan 14% data uji. Hal ini bertujuan untuk memudahkan proses pelatihan serta mengurangi risiko *overfitting* dan *underfitting* pada *dataset*. Gambar berikut menunjukkan hasil dari proses split data yang dilakukan di *website* tersebut.



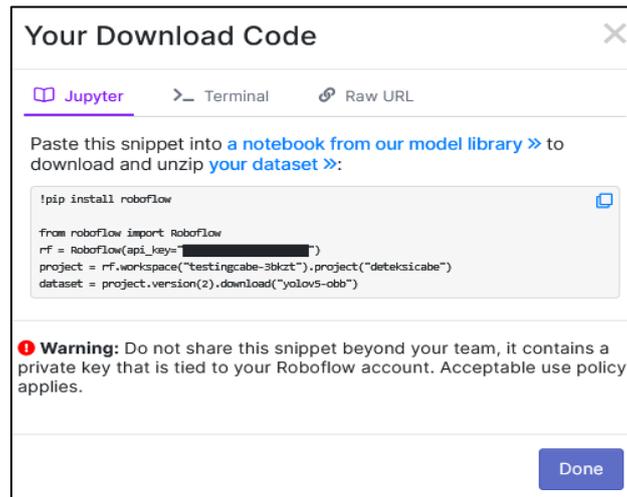
Gambar 10. Data Valid pada *Dataset Split*



Gambar 11. Presentase Gambar pada *Dataset Split*

## Modelling dan Training Data

Setelah *dataset* di *training* pada *Roboflow* maka hasil *dataset* di *export* dengan format *YOLO v5 PyTorch*. Kemudian akan muncul *source code* yang akan dieksekusi untuk mengimpor *dataset* dari *Roboflow*. Setelah menyalin *source code* dari *Roboflow*, kode tersebut dijalankan di *Google Colab*. Setelah *dataset* berhasil diimpor di *Google Colab*, langkah berikutnya adalah melakukan proses pelatihan atau *training dataset* untuk meningkatkan akurasi sistem dan mencegah terjadinya kesalahan dalam pendeteksian objek pada *dataset*.



```

Your Download Code
Jupyter Terminal Raw URL

Paste this snippet into a notebook from our model library >> to download and unzip your dataset >>:

!pip install roboflow

from roboflow import RoboFlow
rf = RoboFlow(api_key="[REDACTED]")
project = rf.workspace("testingcabe-3bkzt").project("detekscabe")
dataset = project.version(2).download("yolov5-obb")

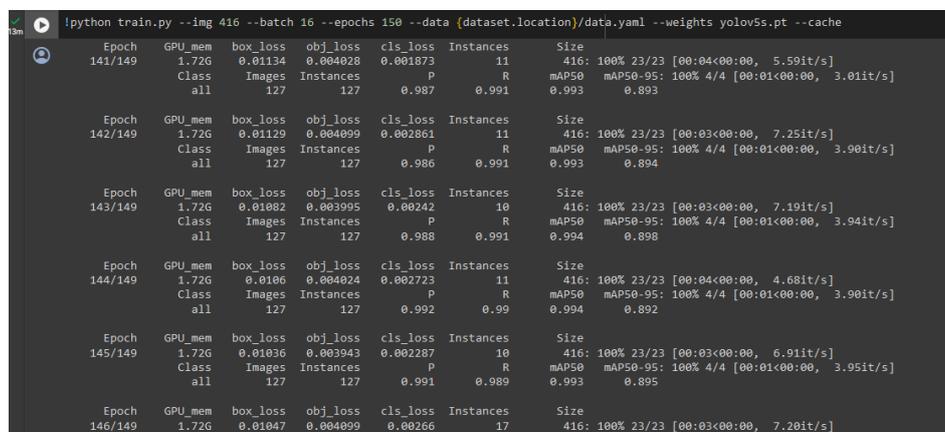
Warning: Do not share this snippet beyond your team, it contains a private key that is tied to your Roboflow account. Acceptable use policy applies.

Done

```

Gambar 12. Source Code Roboflow

*Training* data dilakukan dengan *epoch* sebesar 150, *epoch* sendiri merupakan satu putaran penuh dari seluruh *dataset* yang diberikan kepada model mesin learning. Dalam setiap *epoch*, model melihat dan memproses setiap sampel data satu kali. *Epoch* yang cukup besar ini memberikan kesempatan bagi model untuk mempelajari pola dan karakteristik yang kompleks dari *dataset* cabai rawit, sehingga dapat meningkatkan kemampuan deteksi dan klasifikasi pada sistem deteksi cabai rawit yang dikembangkan.



```

python train.py --img 416 --batch 16 --epochs 150 --data (dataset.location)/data.yaml --weights yolov5s.pt --cache

Epoch 141/149 GPU_mem 1.72G box_loss 0.01134 obj_loss 0.004028 cls_loss 0.001873 Instances 11 Size 416: 100% 23/23 [00:04<00:00, 5.59it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 4/4 [00:01<00:00, 3.01it/s]
all 127 127 0.987 0.991 0.993 0.893

Epoch 142/149 GPU_mem 1.72G box_loss 0.01129 obj_loss 0.004099 cls_loss 0.002861 Instances 11 Size 416: 100% 23/23 [00:03<00:00, 7.25it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 4/4 [00:01<00:00, 3.90it/s]
all 127 127 0.986 0.991 0.993 0.894

Epoch 143/149 GPU_mem 1.72G box_loss 0.01082 obj_loss 0.003995 cls_loss 0.00242 Instances 10 Size 416: 100% 23/23 [00:03<00:00, 7.19it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 4/4 [00:01<00:00, 3.94it/s]
all 127 127 0.988 0.991 0.994 0.898

Epoch 144/149 GPU_mem 1.72G box_loss 0.0106 obj_loss 0.004024 cls_loss 0.002723 Instances 11 Size 416: 100% 23/23 [00:04<00:00, 4.68it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 4/4 [00:01<00:00, 3.90it/s]
all 127 127 0.992 0.99 0.994 0.892

Epoch 145/149 GPU_mem 1.72G box_loss 0.01036 obj_loss 0.003943 cls_loss 0.002287 Instances 10 Size 416: 100% 23/23 [00:03<00:00, 6.91it/s]
Class Images Instances P R mAP50 mAP50-95: 100% 4/4 [00:01<00:00, 3.95it/s]
all 127 127 0.991 0.989 0.993 0.895

Epoch 146/149 GPU_mem 1.72G box_loss 0.01047 obj_loss 0.004099 cls_loss 0.00266 Instances 17 Size 416: 100% 23/23 [00:03<00:00, 7.20it/s]

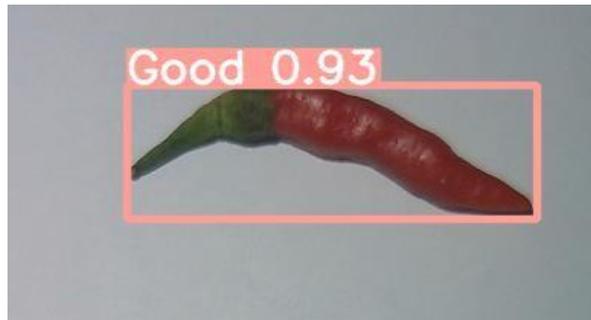
```

Gambar 13. Training Dataset dengan Epoch 150

## Hasil dan Pengujian Sistem

Sistem dijalankan pada *visual studio code* dengan menjalankan perintah Python `detect.py --source 0` pada terminal. Maka kamera akan terbuka dan mendeteksi objek secara *real-time* seperti pada gambar berikut.

Pada pengujian dibawah ini menunjukkan bahwa sistem yang dikembangkan berhasil mendeteksi cabai rawit berkualitas baik kedalam kelas *Good* dengan nilai akurasi sebesar 93%.



**Gambar 14. Hasil Cabai Terdeteksi *Good***

Pada pengujian dibawah ini menunjukkan bahwa sistem yang dikembangkan berhasil mendeteksi cabai rawit berkualitas busuk atau kering kedalam kelas *Bad* dengan nilai akurasi sebesar 96%.



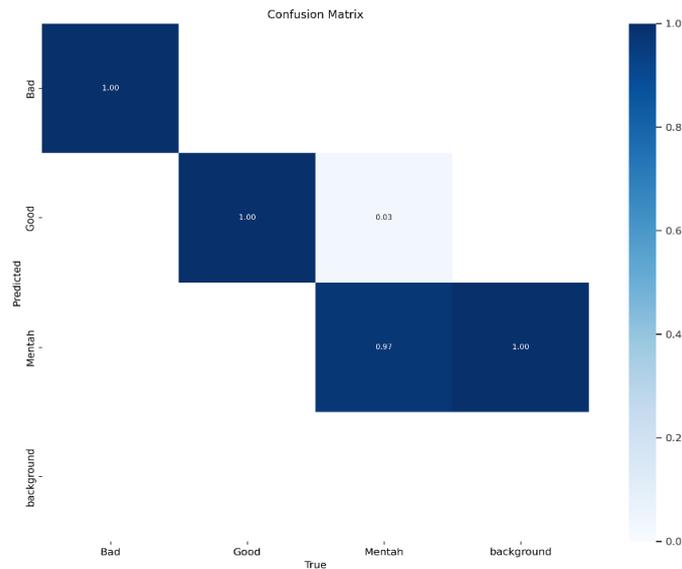
**Gambar 15. Hasil Cabai Terdeteksi *Bad***

Pada pengujian dibawah ini menunjukkan bahwa sistem yang dikembangkan berhasil mendeteksi cabai rawit berkualitas belum matang kedalam kelas *Mentah* dengan nilai akurasi sebesar 95%.

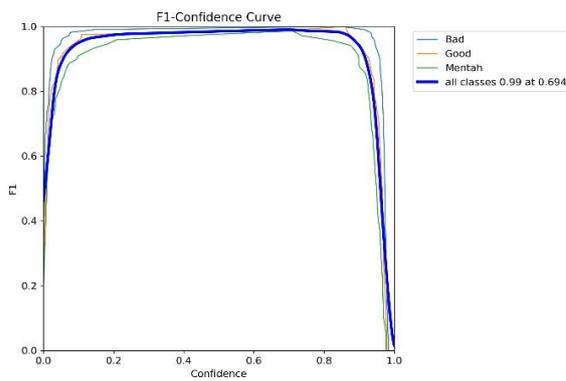


**Gambar 16. Hasil Cabai Terdeteksi *Mentah***

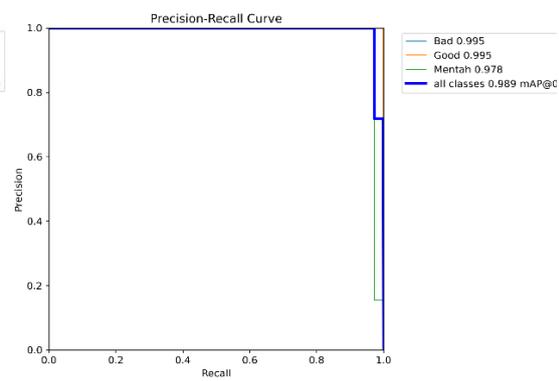
Setelah melakukan pengujian sistem secara *real-time*, dilakukan pula pengujian dengan menggunakan *Confusion Matrix Multi Class* dari hasil *training dataset* menggunakan algoritma *YOLOV5* pada *Google Colab*. Hasilnya ditunjukkan pada kurva berikut. Dijelaskan bahwa pada kelas cabai rawit baik (*good*) diperoleh *true positive* sebesar 1,00. Sedangkan untuk kelas cabai rawit yang busuk (*bad*) tercatat *true positive* sebesar 1,00. Kemudian pada kelas cabai rawit yang belum matang (*mentah*), mendapatkan nilai *true positive* sebesar 0,97 dan *true negative* sebesar 0,03. Selain itu, hasil *training* juga mengidentifikasi background dengan *false positive* sebesar 1,00 pada kelas cabai rawit mentah.



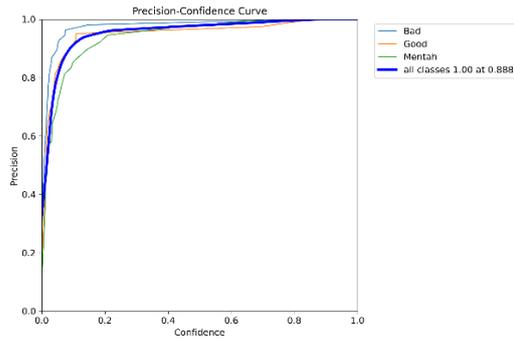
Gambar 17. *Confusion Matrix Class*



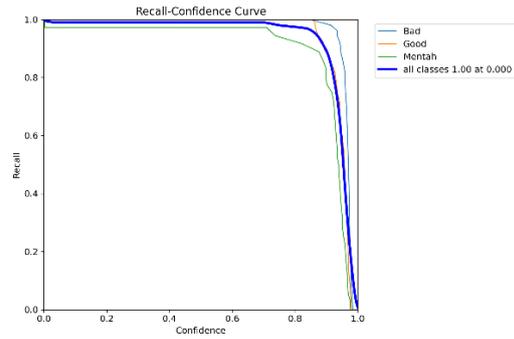
Gambar 18. *F-1 Confidence Curve*



Gambar 19. *Precision-Recall Curve*



Gambar 20. Precision Confidence Curve



Gambar 21. Recall Confidence Curve

### a) F1 Confidence Curve

*F1 Score* adalah ukuran gabungan dari presisi (*precision*) dan *recall* yang memberikan nilai tunggal yang mengukur kinerja suatu model. *F1 Confidence Curve* menunjukkan bagaimana nilai *F1 Score* berkembang seiring dengan variasi tingkat kepercayaan model pada data uji. Kurva ini memberikan wawasan tentang seberapa baik model dapat menjaga keseimbangan antara presisi dan *recall* pada berbagai tingkat kepercayaan (*confidence*).

### b) Precision-Recall Curve

*Precision-Recall Curve* adalah grafik yang mengilustrasikan hubungan antara presisi dan *recall* pada berbagai nilai *threshold* model. Ini membantu dalam mengevaluasi performa model ketika memiliki *trade-off* antara presisi dan *recall*. Kurva ini berguna terutama ketika kelas yang dikejar (*positive class*) memiliki distribusi yang tidak seimbang.

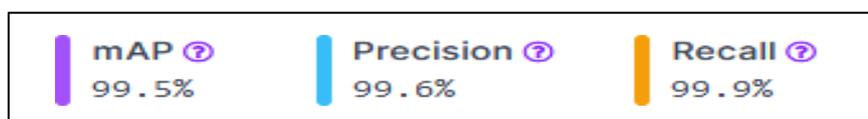
## 3. Precision Confidence Curve

*Precision Confidence Curve* menampilkan bagaimana nilai presisi berubah seiring dengan peningkatan tingkat kepercayaan model pada data uji. Ini memberikan informasi tentang sejauh mana model dapat memberikan prediksi yang akurat dan konsisten pada berbagai tingkat kepercayaan.

## 4. Recall-Confidence Curve

*Recall-Confidence Curve* menunjukkan bagaimana nilai *recall* berkembang seiring dengan variasi tingkat kepercayaan model. Kurva ini memberikan gambaran tentang kemampuan model untuk mengidentifikasi sebanyak mungkin *instance* positif pada berbagai tingkat kepercayaan.

Adapun hasil *mPA*, *Precision*, dan *Recall* pada saat *training dataset* di *Roboflow* sebagai berikut.



Gambar 22. Presentase *mPA*, *Precision*, dan *Recall*

**a) mAP (Mean Average Precision) 99,5%**

mAP merupakan metrik evaluasi yang menggabungkan presisi dan *recall*. Nilai 99,5% menunjukkan tingkat akurasi tinggi dalam mengidentifikasi dan memetakan objek dengan tepat dalam citra.

**b) Precision 99,6%**

*Precision* (presisi) mencerminkan seberapa banyak objek yang diidentifikasi oleh model yang benar. Dengan nilai 99,6%, model cenderung memberikan sedikit sekali hasil palsu positif.

**c) Recall 99,9%**

*Recall* mencerminkan seberapa banyak objek yang seharusnya diidentifikasi oleh model yang berhasil diidentifikasi. Dengan nilai 99,9%, model cenderung sangat efektif dalam menangkap sebagian besar objek yang seharusnya diidentifikasi.

## 5. KESIMPULAN DAN SARAN

Hasil penelitian ini memberikan solusi terhadap permasalahan klasifikasi kualitas cabai rawit dalam konteks *Good*, *Bad*, dan *Mentah*. Pendeteksian dilakukan menggunakan sistem *YOLO V5*, mencapai akurasi sebesar 90%. Sistem berhasil mengenali cabai rawit dalam tiga kondisi, yaitu baik (*good*), buruk (*bad*), dan belum matang (*mentah*). Fokus penelitian ini adalah menciptakan sistem klasifikasi kualitas cabai rawit berbasis *YOLO V5*. Sebanyak 565 foto digunakan sebagai *dataset*, dan proses *labelling* dilakukan dengan menggunakan *Roboflow*. *Dataset* dilabeli menjadi tiga kelas utama: *good*, *bad*, dan *mentah*. Proses *training* dilakukan menggunakan algoritma *YOLO V5* dengan *epoch* sebesar 150. Evaluasi model menunjukkan hasil yang memuaskan, dengan *mAP* mencapai 99,5%, *precision* sebesar 99,6%, dan *recall* sebesar 99,9%. Tahap terakhir melibatkan pengujian secara *real-time*, menegaskan kehandalan dan efektivitas sistem dalam mendeteksi kualitas cabai rawit. Hasil ini diharapkan dapat memberikan kontribusi pada pengembangan teknologi bidang klasifikasi dan deteksi objek berbasis visualisasi menggunakan pendekatan *YOLO V5*.

Saran dari penelitian ini mengimplementasikan metode *YOLO (You Only Look Once)* dalam sistem deteksi kualitas cabai rawit. Sistem berhasil mengenali cabai rawit dalam tiga kondisi, baik (*good*), buruk (*bad*), dan belum matang (*mentah*). Dalam penelitian selanjutnya, disarankan untuk memperluas studi kasus melibatkan berbagai variasi tanaman yang membutuhkan sortir dengan akurat dan cepat, menyempurnakan parameter *YOLO* untuk tingkat ketelitian yang optimal, serta memperbanyak *dataset* agar tingkat akurasi saat mendeteksi objek lebih baik.

## DAFTAR REFERENSI

- Adrianto, L. B., Wahyuddin, M. I., & Winarsih, W. (2021). Implementasi Deep Learning untuk Sistem Keamanan Data Pribadi Menggunakan Pengenalan Wajah dengan Metode Eigenface Berbasis Android. *Jurnal JTIC (Jurnal Teknologi Informasi dan Komunikasi)*, 5(1), 89-96.
- Amin, M., & Prihantini, C. I. Analysis of Production and Production Risk of Rawit Chili in Watunohu District, North Kolaka District. *Agrimor*, 6(1), 15-21.
- Gulo, T., & Harefa, D. (2023). Identifikasi Serangga (Insekta) yang merugikan Pada Tanaman Cabai Rawit di Desa Sisarahili Ekholo Kecamatan Lolowau Kabupaten Nias Sealatan. *Jurnal Sapta Agrica*, 2(1), 50-61.
- Husna, M. N., & Prakoso, J. A. (2022). PERAMALAN HARGA CABAI MERAH DI KOTA SEMARANG DENGAN PENDEKATAN MODEL ARCH GARCH. *JISMA: Jurnal Ilmu Sosial, Manajemen, dan Akuntansi*, 1(4), 607-614.
- Kurniawan, R., Martadinata, A. T., & Cahyo, S. D. (2023). Klasifikasi Tingkat Kematangan Buah Sawit Berbasis Deep Learning dengan Menggunakan Arsitektur Yolov5. *Journal of Information System Research (JOSH)*, 5(1), 302-309. <https://doi.org/10.47065/josh.v5i1.4408>
- Luthfi, A., Sari, A. M., Dewi, G. R., Dwijayanti, Y., Satya, T. P., & Sari, A. R. (2023). Penentuan Klasifikasi Kematangan dan Kualitas Cabai Merah Besar (*Capsicum annum* L.) Menggunakan Aplikasi Color Grab. *Agrointek*, 17(2), 288-294. <https://doi.org/10.21107/agrointek.v17i2.12388>
- Nurajrin, I. (2022). RANCANG BANGUN APLIKASI DETEKSI KEPRIBADIAN DASAR MANUSIA BERDASARKAN BENTUK WAJAH MENGGUNAKAN METODE BACKPROPAGATION. *Jurnal Teknologi Pintar*, 2(4).
- Padmaningrum, D., Suminah, S., Utami, B. W., Ihsaniyati, H., & Widiyanti, E. (2022). Pemberdayaan kelompok tani melalui budidaya cabai sebagai upaya peningkatan pendapatan petani lahan kering di Kabupaten Sukoharjo. *E-Dimas: Jurnal Pengabdian kepada Masyarakat*, 13(1), 158-167.
- Priyani, D. N. (2021). Sortasi Cabai Rawit pada Sistem Konveyor dengan Sensor TCS34725 Menggunakan Metode KNN (K-Nearest Neighbor). Tugas Akhir, Universitas Teknologi Yogyakarta.
- Purwawijaya, E., Singarimbun, R. N., & Pasaribu, H. (2022). Implementasi Face Recognition Pada Absensi Karyawan Menggunakan Local Binary Pattern Histogram dan SHA 256 bit. *Jurnal Media Informatika Budidarma*, 6(4), 2383-2391.
- Putra, E. D. (2021). Identifikasi Kematangan Cabai Menggunakan Operasi Morfologi (Opening dan Closing) dan Metode Backpropagation. *Sistemasi: Jurnal Sistem Informasi*, 10(1), 96-105.
- Rahardjo, V. A., & Setiyadi, D. (2021). Implementasi Sensor Pengukur Kelembapan Tanah Dan Penyiraman Otomatis Serta Monitoring Pada Kebun Tanaman Cabai Rawit. *Aisyah Journal of Informatics and Electrical Engineering*, 3(2), 106-115.

- Rahma, L., Syaputra, H., Mirza, A. H., & Purnamasari, S. D. (2021). Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once). *Jurnal Nasional Ilmu Komputer*, 2(3), 213-232.
- Ratna, S. (2020). Pengolahan citra digital dan histogram dengan phyton dan text editor pycharm. *Technologia: Jurnal Ilmiah*, 11(3), 181-186.
- Riva, L. S., & Jayanta, J. (2023). Deteksi Penyakit Tanaman Cabai Menggunakan Algoritma YOLOv5 Dengan Variasi Pembagian Data. *Jurnal Informatika: Jurnal Pengembangan IT*, 8(3), 248-254.
- Salamah, I., Said, M. R. A., & Soim, S. (2022). Perancangan Alat Identifikasi Wajah Dengan Algoritma You Only Look Once (YOLO) Untuk Presensi Mahasiswa. *Jurnal Media Informatika Budidarma*, 6(3), 1492-1500.
- Sardianti, A. L. (2020). Analisis Pengaruh Permintaan Cabai Rawit (*Capsicum Baccatum*) di Desa Mohungo Kecamatan Tilamuta. *Journal of Agritech Science (JASc)*, 4(1), 52-60.