

## **IMPLEMENTASI ALGORITMA DEEP LEARNING YOLO (YOU ONLY LOOK ONCE) UNTUK DETEKSI KUALITAS KENTANG SEGAR DAN BUSUK SECARA REAL TIME**

Ocha Alfiano<sup>1</sup>, dan Santi Rahayu<sup>2</sup>

<sup>1</sup>Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia  
e-mail: <sup>1</sup>Ochaalfiano007@gmail.com

<sup>2</sup>Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia  
e-mail: <sup>2</sup>dosen02666@unpam.ac.id

### **Abstract**

*Potatoes are one of the important sources of carbohydrates whose quality greatly affects the food industry. The potato quality inspection process that is still carried out manually is often time-consuming and prone to human error. This research developed a quality detection system for fresh and rotten potatoes using the YOLOv8n version of the You Only Look Once (YOLO) algorithm. The study began with the collection of 1000 potato photos that were split into 85% for training, 10% for validation, and 5% for testing. The dataset was then labeled using the Roboflow platform and was aggregated to bring the total to 2304 photos. The training results showed that the YOLOv8n model achieved 99.9% accuracy, 100% recall, 99.5% mAP50, and 97.9% mAP50-90. The model is implemented in a Flask-based website to enable real-time detection. Although the model produces good performance, there are some errors in recognizing object classes. Overall, this system is capable of effectively detecting the quality of potatoes, reducing waste, and maintaining product quality.*

*Keywords: YOLO; deteksi objek; kualitas kentang; deep learning; real-time detection.*

### **Abstrak**

Kentang adalah salah satu sumber karbohidrat penting yang kualitasnya sangat mempengaruhi industri pangan. Proses inspeksi kualitas kentang yang masih dilakukan secara manual sering kali memakan waktu dan rentan terhadap kesalahan manusia. Penelitian ini mengembangkan sistem deteksi kualitas kentang segar dan busuk menggunakan algoritma You Only Look Once (YOLO) versi YOLOv8n. Penelitian dimulai dengan pengumpulan 1000 foto kentang yang di-split menjadi 85% untuk training, 10% untuk validation, dan 5% untuk testing. Dataset kemudian dilabeli menggunakan platform Roboflow dan diaugmentasi sehingga totalnya menjadi 2304 foto. Hasil pelatihan menunjukkan model YOLOv8n mencapai precision 99.9%, recall 100%, mAP50 99.5%, dan mAP50-90 97.9%. Model diimplementasikan dalam website berbasis Flask untuk memungkinkan deteksi secara real-time. Meskipun model menghasilkan performa yang baik, tetapi terdapat beberapa kesalahan dalam mengenali kelas objek. Secara keseluruhan, sistem ini mampu mendeteksi kualitas kentang dengan baik, mengurangi pemborosan, dan menjaga kualitas produk.

Kata Kunci: YOLO; deteksi objek; kualitas kentang; deep learning; real-time detection

## 1. PENDAHULUAN

Kentang adalah tanaman dikotil dari keluarga Solanaceae. Tanaman ini bisa berkembang biak secara vegetatif melalui umbinya. Kentang tumbuh dengan baik di daerah beriklim dingin dan menjadi salah satu sumber karbohidrat penting bagi manusia. [1]

Sedangkan menurut diva dan agmasari Kentang merupakan tanaman umbi-umbian yang kaya akan karbohidrat dan dapat diolah menjadi berbagai jenis makanan. Untuk memastikan kualitas kentang yang baik sebelum diolah, ada beberapa ciri yang perlu diperhatikan. Kentang segar ditandai dengan tekstur yang keras, kulit yang halus tanpa noda hitam, warna kuning pucat, dan bau tanah alami. Sebaliknya, kentang busuk memiliki tekstur lembek, noda hitam pada kulit, warna yang berubah menjadi hijau, dan bau yang tidak sedap. [2]

Kualitas kentang memegang peranan penting dalam industri pangan. Kentang yang baik tidak hanya mempengaruhi rasa dan tekstur masakan, tetapi juga meningkatkan kepercayaan pelanggan dan reputasi penjual. Dengan menjaga konsistensi kualitas kentang, penjual dapat menciptakan pengalaman memasak yang lebih baik, meningkatkan efisiensi operasional, dan meminimalkan potensi pemborosan bahan baku.

Oleh karena itu, pemilihan kualitas kentang pada pengolahan makanan merupakan aspek penting dalam keberhasilan seorang penjual. Namun, proses inspeksi kualitas kentang yang masih dilakukan secara manual menjadi masalah utama yang dihadapi oleh sektor pertanian dan pengolahan makanan. Selain memakan waktu, proses ini rentan terhadap kesalahan manusia, terutama jika dilakukan dalam skala besar.

Berdasarkan hal tersebut terdapat solusi yang cepat, tepat, dan efisien untuk mendeteksi kualitas kentang secara otomatis, sehingga sangat penting untuk mengatasi masalah ini. Karena campuran kentang busuk dengan kentang segar dapat menyebabkan penurunan kualitas pada produk akhir. Maka dari itu, dihadirkan sebuah sistem algoritma You Only Look Once (YOLO) yang dapat mendeteksi kualitas kentang secara langsung sehingga dapat mempermudah penjual dalam memilih kualitas kentang segar dan busuk.

YOLO merupakan singkatan dari "You Only Look Once", YOLO adalah algoritma yang dirancang khusus untuk melakukan deteksi objek secara real-time. Proses deteksi pada YOLO dilakukan dengan cara yang efisien dan cepat, menggabungkan prediksi kotak pembatas (bounding box) dan probabilitas kelas objek dalam satu evaluasi. YOLO dapat memungkinkan sistem untuk langsung menganalisis seluruh gambar secara keseluruhan dan menghasilkan deteksi objek dengan waktu yang cepat. Melalui teknik ini, YOLO menjadi pilihan populer untuk aplikasi real-time seperti deteksi objek pada video dan pengolahan citra secara cepat.[3]

Berdasarkan pada penelitian sebelumnya yang telah dilakukan oleh Wibowo dkk. Penelitian ini bertujuan untuk mengembangkan sebuah sistem yang dapat mendeteksi kualitas buah segar dan busuk dengan memanfaatkan algoritma You Only Look Once (YOLO) untuk 3 jenis buah yaitu apel, pisang, dan jeruk. Dengan

total keseluruhan 1200 citra untuk data latih dan 330 citra untuk data validasi dan 6 citra untuk test menggunakan algoritma YOLOv5. Hasil pada penelitian ini berhasil menerapkan algoritma You Only Look Once (YOLO) untuk mendeteksi kualitas segar dan busuk pada beberapa buah dengan hasil akurasi mencapai 90%. [4]

Berdasarkan permasalahan yang telah dijelaskan sebelumnya, keberhasilan perkembangan teknologi ini memberikan dampak bagi sebagian besar masyarakat serta menjadikan komputer dapat belajar dan beradaptasi terhadap data, sehingga membuka jalan pada berbagai inovasi. Salah satu bentuk dari perkembangan teknologi saat ini yaitu seperti kecerdasan buatan (AI), machine learning, dan deep learning. Bukan hanya itu, Salah satu perkembangan teknologi saat ini juga terdapat pada bidang deteksi objek. Deteksi objek merupakan prosedur menentukan keberadaan, ruang lingkup, dan lokasi objek pada gambar.[5]

Deteksi objek merupakan suatu teknik dalam bidang Computer Vision yang bertujuan untuk mengidentifikasi keberadaan objek dalam gambar atau video. Algoritma yang diterapkan dalam proses deteksi objek menggunakan metode pembelajaran mesin mendalam untuk menghasilkan hasil yang optimal. Pendekatan ini

memungkinkan sistem untuk secara otomatis dan efisien mengenali dan menandai objek dalam berbagai konteks visual. Dengan memanfaatkan kecerdasan buatan dan model neural network yang kompleks, deteksi objek dapat memberikan kinerja yang tinggi dalam tugas identifikasi objek.[6]

Salah satu algoritma terbaru yang telah menghasilkan deteksi objek dengan akurasi yang tinggi serta real time yaitu algoritma deep learning YOLO (You Only Look Once). Algoritma YOLO merupakan metode deteksi dengan model terpadu (unified), yang mana dengan jaringan saraf tunggal (single neural network) dapat memprediksi kotak pembatas (Bounding Box) dan probabilitas kelas secara langsung dalam satu gambar penuh pada sekali tangkapan. Keunggulan utama sistem algoritma YOLO adalah kecepatan serta akurasinya yang tinggi, sehingga memungkinkan pendeteksian kentang dapat dilakukan secara cepat dan tepat.[7]

Berdasarkan uraian di atas, peneliti mengharapkan penelitian ini dapat membantu mengembangkan suatu sistem deteksi otomatis yang dapat diandalkan menggunakan algoritma You Only Look Once (YOLO) yang dapat membedakan kualitas kentang segar dan busuk secara cepat dan otomatis. Sehingga teknologi ini dapat meningkatkan efisiensi, mengurangi pemborosan, dan menjaga kualitas produk.

## **2. PENELITIAN YANG TERKAIT**

Penelitian terkait ini akan menjadi acuan bagi penelitian yang dilakukan oleh penulis dan membantu dalam menyelesaikan penelitian ini. Terdapat beberapa penelitian terdahulu mengenai topik terkait yang dijadikan acuan dalam pengembangan proposal ini, yaitu :

Penelitian yang dilakukan oleh Sarosa dan Muna dengan judul “Implementasi Algoritma You Only Look Once (YOLO) Untuk Deteksi Korban Bencana Alam” bertujuan mengembangkan sistem deteksi korban bencana alam berbasis citra. Penelitian ini menggunakan dua varian YOLO, yakni YOLOv3 dan YOLOv3 Tiny, yang diuji dengan 100 data train dan 100 data validasi. Hasil menunjukkan bahwa YOLOv3 mencapai F1 Score 95,3%.[8]

Penelitian Jannah dkk. berjudul “Implementasi Algoritma YOLO untuk Deteksi

Rias Adat Nusantara” mengkaji pentingnya tata rias pengantin tradisional di Indonesia. Menggunakan YOLOv4-Tiny, penelitian ini melatih 1478 citra kepala memelai wanita dari 14 jenis rias adat. Hasilnya, aplikasi mampu mendeteksi rias adat dengan akurasi rata-rata 95,20% dan waktu deteksi 327ms.[3]

Penelitian Janapriya berjudul “Pengenalan Jenis Rambu Lalu Lintas Menggunakan Metode YOLO V5” bertujuan meningkatkan kesadaran masyarakat akan aturan rambu lalu lintas melalui sistem berbasis kecerdasan buatan. Menggunakan YOLO V5, penelitian ini menganalisis 1100 citra rambu lalu lintas, dibagi menjadi 990 citra latih dan 110 citra validasi. Hasilnya menunjukkan precision 0,923, recall 0,826, mAP50 0,965, dan mAP50-95 0,924.[9]

Penelitian Rahma dkk. berjudul “Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO” menggunakan YOLO untuk mengenali 31 jenis makanan khas Palembang. Dengan 1955 gambar sebagai data latih dan 31 gambar untuk uji, hasilnya menunjukkan akurasi rata-rata 96% dan kecepatan deteksi 40.486.129 milidetik. Model YOLOv3 yang digunakan terbukti memiliki akurasi dan kecepatan yang baik.[10]

Penelitian Handayani berjudul “Implementasi Metode YOLO untuk Deteksi Kesegaran Telur Ayam Berdasarkan Citra Cangkang” mengatasi keterbatasan teknologi dalam pemilahan telur di Indonesia. Dengan menggunakan 1.898 gambar cangkang telur, model YOLO dilatih untuk mendeteksi kualitas telur dalam dua kategori: segar dan tidak segar. Data terdiri dari 1.599 gambar pelatihan dan 299 gambar validasi. Hasil evaluasi menunjukkan presisi 0,847, recall 0,904, mAP50 0,951, dan mAP50-95 0,932, membuktikan model mampu mendeteksi dan mengklasifikasikan kualitas telur dengan baik. [11]

Penelitian Harun berjudul “Implementasi Deep Learning Menggunakan Metode YOLO untuk Mendeteksi Rokok” membahas keterbatasan dalam pencegahan aktivitas merokok di tempat umum yang sering bergantung pada pengawasan manual. Penelitian ini menggunakan YOLO untuk mendeteksi rokok secara real-time. Model YOLOv3, YOLOv3-Tiny, dan YOLOv4 diuji pada dataset dengan 1.169 gambar rokok dan 832

gambar bukan rokok. Hasil terbaik dicapai dengan YOLOv4, mencapai mAP 92,54 dan F1 Score 0,89. Sistem ini mampu mendeteksi rokok hingga jarak 4,5 meter dengan akurasi 99,03% pada jarak 1 meter. [12]

Penelitian Staya dkk. berjudul “Sistem Pendeteksi Plat Nomor Polisi Kendaraan Dengan Arsitektur YOLOv8” menggunakan YOLOv8 untuk deteksi objek dan PaddleOCR untuk mengenali karakter plat nomor. Dataset terdiri dari 2.167 gambar dengan 5 label, dibagi menjadi Training Set (79%), Validation Set (20%), dan Testing Set (1%). Model YOLOv8 mencapai mAP50 0,868 dan mAP50-95 0,753, serta precision 0,871 dan recall 0,85, menunjukkan performa deteksi yang baik. PaddleOCR juga berhasil mengenali karakter plat nomor pada gambar berkualitas rendah.[13]

Penelitian Hayati dkk. berjudul “Object Tracking Menggunakan Algoritma YOLOv8 Untuk Menghitung Kendaraan” mengembangkan sistem deteksi dan pencacahan kendaraan menggunakan YOLOv8. Penelitian ini memanfaatkan dataset multiclass dari Roboflow dengan 4.680 data dan 9 kelas kendaraan. Implementasi metodologi AI Project Cycle mencakup problem scoping hingga evaluasi menggunakan confusion matrix. Hasilnya, YOLOv8 mencapai akurasi 89%, dengan precision 89%, recall 90%, dan F1-Score 89%, menunjukkan efektivitas dalam deteksi dan pencacahan kendaraan.[14]

Penelitian Ibrahim dan Latifa berjudul “Penerapan Algoritma YOLOv8 Dalam Deteksi Waktu Panen Tanaman Pakcoy Berbasis Website” mengembangkan sistem deteksi tanaman pakcoy siap panen menggunakan YOLOv8 untuk meningkatkan akurasi dan objektivitas penentuan waktu panen. Dataset terdiri dari 6800 foto, dibagi menjadi training set (1700 foto), validation set (180 foto), dan testing set (80 foto). Hasil pelatihan model menunjukkan precision 97.1%, recall 97.4%, mAP50 98%, dan mAP50-95 61%. Model ini diimplementasikan dalam sistem website dan diuji dengan hasil yang memuaskan.[15]

Penelitian Maleh dkk. berjudul “Implementasi Algoritma YOLO untuk Object Detection Sarang Orang Utan” menggunakan YOLO untuk mendeteksi sarang orangutan di hutan

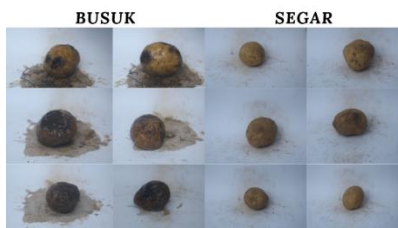
gambut Kalimantan, yang berperan penting dalam menjaga keseimbangan lingkungan. Dengan 1970 citra latih dan 414 citra sarang orangutan, model mencapai precision 0,973, recall 0,949, mAP\_0.5 0,969, dan mAP\_0.5:0.95 0,630. Setelah 58 jam pelatihan dan 217 epoch, model berhasil mendeteksi sarang orangutan dengan akurasi 99,9%.[7][7]

Dapat dilihat dari penelitian terdahulu bahwa model YOLO sebelumnya telah terbukti memiliki kecepatan dan keakuratan yang tinggi dalam mendeteksi objek. Penelitian ini memiliki perbedaan dari penelitian sebelumnya yaitu dengan menggunakan model terbaru YOLOv8 untuk mendeteksi kualitas kentang segar dan busuk. Pada penelitian ini, model YOLOv8n digunakan dengan pembagian dataset yaitu 85% untuk training, 10% untuk validation, dan 5% untuk testing. Selain itu, dilakukan juga augmentasi pada dataset untuk memperkaya dataset dalam berbagai kondisi. Model YOLO yang sudah dilatih kemudian akan diimplementasikan ke dalam sebuah website agar dapat mendeteksi video dan melakukan deteksi secara real-time menggunakan webcam. Dengan menggunakan YOLOv8, diharapkan penelitian ini dapat menghasilkan deteksi yang lebih maksimal dan akurat dalam menentukan kualitas kentang segar dan busuk. Dengan demikian, penelitian ini diharapkan mampu memberikan hasil yang lebih baik dari penelitian sebelumnya dalam hal deteksi dengan keakuratan yang tinggi.

### **3. METODE PENELITIAN**

#### **5.1 Pengumpulan Dataset**

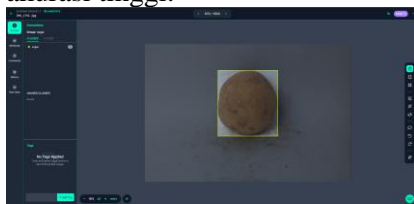
Dataset dalam penelitian ini terdiri dari 1.000 gambar kentang segar dan busuk, masing-masing 500 gambar. Dataset ini dibuat oleh peneliti dengan memotret kentang dalam berbagai kondisi, dari yang sangat segar hingga busuk. Pengumpulan data ini bertujuan untuk melatih algoritma YOLO dalam mendeteksi kualitas kentang dengan akurasi tinggi. Gambar dibawah menampilkan beberapa contoh dataset, yang dirancang untuk memastikan model mampu mengenali perbedaan kualitas kentang secara efektif.



Gbr 1. Dataset

## 5.2 Pelabelan Dataset

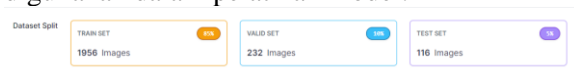
Setelah mengumpulkan dataset, gambar dilabeli menggunakan platform Roboflow. Dataset yang terdiri dari 1.000 gambar kentang segar dan busuk ini diproses dengan menandai area penting yang menunjukkan kualitas kentang. Proses pelabelan pada Roboflow memudahkan peneliti untuk mengklasifikasikan gambar sesuai kategori yang ditentukan. Gambar dibawah menunjukkan proses pelabelan yang dilakukan untuk mempersiapkan dataset dalam melatih model YOLOv8n guna mendeteksi kualitas kentang dengan akurasi tinggi.



Gbr 2. Pelabelan Dataset

## 5.3 Pembagian Dataset

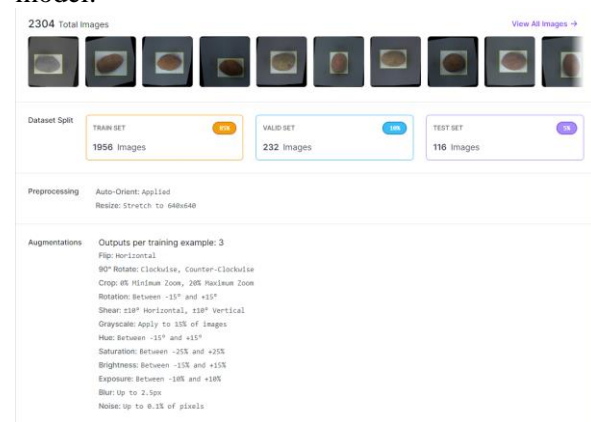
Setelah proses pelabelan menggunakan Roboflow selesai, dataset kentang dibagi menjadi tiga bagian: 85% untuk pelatihan, 10% untuk validasi, dan 5% untuk pengujian. Pembagian ini penting untuk mengevaluasi kinerja model YOLOv8n secara obyektif dan memastikan kemampuannya dalam mendeteksi kualitas kentang dengan baik. Dataset awal yang terdiri dari 1.000 gambar kentang segar dan busuk kemudian mengalami proses augmentasi, meningkatkan jumlah total gambar menjadi 2.304. Augmentasi ini bertujuan untuk memperkaya variasi data yang digunakan dalam pelatihan model.



Gbr 3. Split Dataset

## 5.4 Augmentasi Dataset

Proses augmentasi dilakukan pada dataset kentang segar dan busuk untuk memperkaya variasi data dengan metode seperti flip, crop, dan rotation di Roboflow. Teknik-teknik ini meningkatkan jumlah gambar dari 1.000 menjadi 2.304, menciptakan dataset yang lebih beragam untuk melatih model YOLOv8n agar mampu mengenali kualitas kentang dengan lebih akurat. Setelah dataset dibagi untuk pelatihan, validasi, dan pengujian, serta melalui augmentasi, dataset tersebut diunggah ke Google Drive. Penyimpanan di Google Drive memungkinkan peneliti mengakses dan mengelola dataset dengan mudah selama pelatihan di Google Colab, memastikan efisiensi dalam proses training model.



Gbr 4. Augmentasi Dataset

## 2.5. Perancangan YOLO dan pelatihan

Setelah mempersiapkan dataset, langkah berikutnya adalah menginstal library Ultralytics di Google Colab menggunakan code `!pip install ultralytics`. Library ini penting untuk proses pelatihan model deteksi objek seperti YOLO. Kode diatas menginstruksikan instalasi library di Google Colab, memastikan semua dependensi siap digunakan.

Selanjutnya, mengimpor library Ultralytics dengan kode `import ultralytics` dan menjalankan `ultralytics.checks()` untuk memastikan sistem siap digunakan. Untuk memulai pelatihan, kode `from ultralytics import YOLO` mengimpor class YOLO dari library ini, memungkinkan penggunaan metode yang disediakan untuk pelatihan, validasi, dan inferensi model. Kode `model = YOLO('yolov8n.pt')` untuk mengunduh model YOLOv8n yang telah dipretrain, mempersingkat

waktu pengembangan dan meningkatkan efisiensi pelatihan.

Pelatihan model menggunakan YOLOv8n pretrain dilakukan dengan dataset yang disimpan di Google Drive, yang terdiri dari 2.304 gambar yang dibagi menjadi 85% data pelatihan, 10% validasi, dan 5% pengujian. Selama pelatihan, model memperbarui bobotnya untuk meningkatkan akurasi prediksi kotak pembatas pada objek. Proses ini memakan waktu 2 jam 5 menit, dan hasil pelatihan menunjukkan tingkat precision 99.9%, recall 100%, mAP50 99.5%, dan mAP50-95 97.9%, yang dirangkum dalam Gambar dibawah ini.

```

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
98/100   0.846    0.1852   0.1725   0.9338    19         640 100% 98/98 [01:05:00.00, 1.561t/s]
Class   Images  Instances  Box(P)  R      mAP50  mAP50-95 | 100% 6/6 [00:05:00.00, 1.651t/s]
all     232     232       0.999   1      0.995  0.971

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
99/100   0.256    0.389   0.1748   0.9405    19         640 100% 99/98 [01:05:00.00, 1.491t/s]
Class   Images  Instances  Box(P)  R      mAP50  mAP50-95 | 100% 6/6 [00:05:00.00, 1.341t/s]
all     232     232       0.999   1      0.995  0.972

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
100/100  0.170    0.3861  0.1736   0.9351    19         640 100% 100/98 [01:05:00.00, 1.581t/s]
Class   Images  Instances  Box(P)  R      mAP50  mAP50-95 | 100% 6/6 [00:05:00.00, 1.921t/s]
all     232     232       0.999   1      0.995  0.973

200 epochs completed in 2.091 hours.
optimizer stripped from runs/detect/train/weights/last.pt, 52.000
optimizer stripped from runs/detect/train/weights/best.pt, 52.000

Validating runs/detect/train/weights/best.pt...
Utilities: yolov8.py @ Python: 3.10.12 torch: 2.3.0+cu121 CUDA: 0 (Tesla T4, 15360MiB)
model summary (total): 218 layers, 2688920 parameters, 0 gradients, 75.7 GiB
Class   Images  Instances  Box(P)  R      mAP50  mAP50-95 | 100% 6/6 [00:05:00.00, 1.894t/s]
all     232     232       0.999   1      0.995  0.979
busuk   56      56        0.999   1      0.995  0.984
segar   176     176       0.998   1      0.995  0.974
Speed: 0.3ms preprocess, 18.0ms inference, 0.8ms loss, 5.4ms postprocess per image
Results saved to runs/detect/train
Learn more at https://docs.ultralytics.com/modes/train
    
```

Gbr 5. Hasil Training

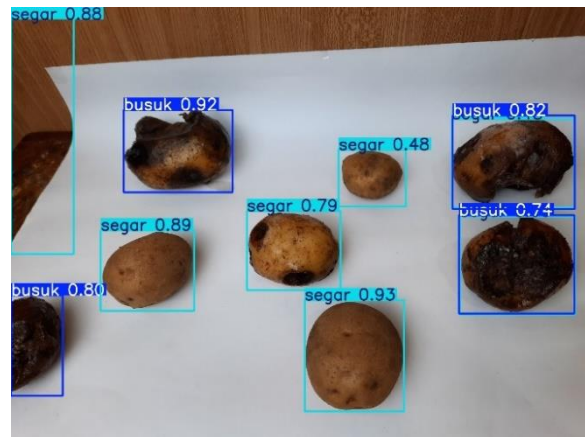
#### 4. HASIL DAN PEMBAHASAN

##### 6.1 Pengujian Model

Table I. Pengujian Dataset Test

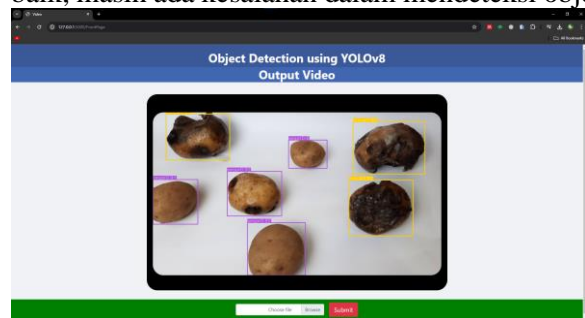
Kategori	Total Item	Deteksi Benar	Deteksi Salah	Akurasi
Segar	58	58	0	100%
Busuk	58	58	0	100%

Table di atas menampilkan hasil pengujian model YOLOv8n menggunakan dataset yang terdiri dari 58 sampel segar dan 58 sampel busuk. Hasil pengujian menunjukkan bahwa model dapat mendeteksi kedua jenis kualitas dengan akurat, menegaskan efektivitasnya dalam mengidentifikasi objek pada data tes. Meskipun model ini menunjukkan tingkat kepercayaan yang tinggi dalam prediksi, ada beberapa gambar atau video, terutama dari Google, yang tidak terdeteksi dengan baik. Hal ini kemungkinan disebabkan oleh keterbatasan variasi dalam dataset yang digunakan.



Gbr 6. Pengujian Multiobject

Gambar di atas menunjukkan hasil pengujian model YOLOv8n dengan 3 kentang segar dan 5 kentang busuk. Model mendeteksi 4 kentang segar dan 4 kentang busuk, salah mendeteksi satu kentang busuk sebagai segar. Selain itu, model juga mendeteksi bagian background sebagai kentang segar. Ini menunjukkan bahwa meskipun YOLOv8n memiliki kemampuan deteksi yang baik, masih ada kesalahan dalam mendeteksi objek



Gbr 7. Hasil Pengujian Video Website

Model deteksi kualitas kentang telah diimplementasikan pada sebuah website untuk mendeteksi kualitas kentang secara realtime menggunakan webcam dan video, namun tidak mendukung deteksi pada gambar. Keterbatasan dataset membuat model hanya mampu mendeteksi kentang segar dan busuk tertentu dalam dataset pengujian. Gambar di atas menunjukkan hasil pengujian model YOLOv8n yang diimplementasikan di website, menggunakan video dengan frame rate rendah karena keterbatasan perangkat. Meskipun model mendeteksi sebagian besar objek dengan benar, masih ada beberapa

kesalahan. Secara keseluruhan, model menunjukkan kinerja yang baik.

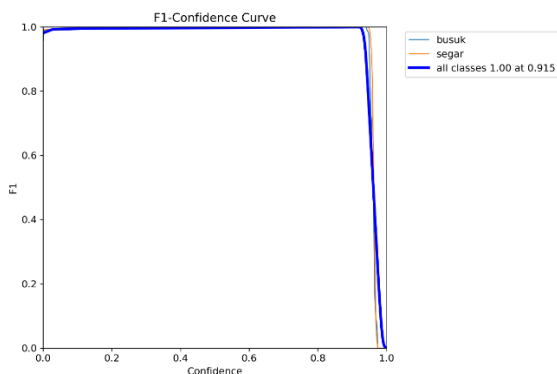


Gbr 8 Hasil Pengujian Real-Time Website

Gambar di atas menunjukkan pengujian model YOLOv8n yang diintegrasikan ke website untuk deteksi kualitas kentang secara real-time menggunakan webcam. Meskipun fps rendah karena keterbatasan perangkat, YOLOv8n dapat mendeteksi objek dengan akurat, meskipun ada beberapa kesalahan deteksi. Secara keseluruhan, model ini menunjukkan performa baik dalam situasi real-time.

## 6.2 Akurasi Hasil

### 6.2.1 F1-Score

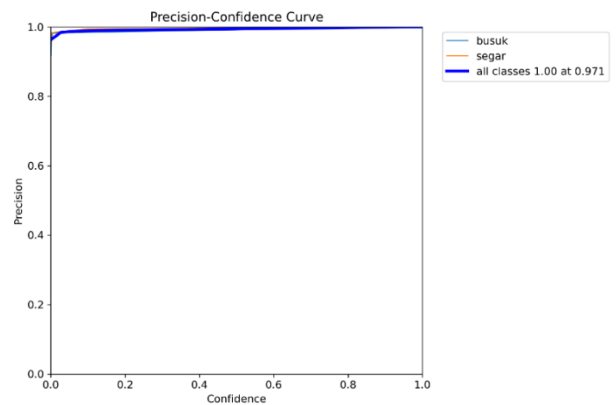


Gbr 9. F1-Score

Gambar di atas menunjukkan kurva F1-Confidence untuk mengevaluasi kinerja model klasifikasi. Sumbu horizontal menunjukkan nilai confidence, sedangkan sumbu vertikal menunjukkan F1-score. Kurva ini menggambarkan hubungan antara kepercayaan prediksi dan performa model untuk dua kelas: "busuk" (biru muda) dan "segar" (oranye). Kurva "all classes" (semua kelas, yang digambarkan dengan garis tebal biru, menunjukkan nilai F1-score 1.00 pada tingkat confidence 0.9, menunjukkan performa model yang baik. Secara umum, nilai F1-score

mendekati 1 pada berbagai tingkat confidence, tetapi mulai menurun mendekati confidence 1.0, yang bisa mengindikasikan overfitting atau kurangnya data representatif pada tingkat confidence sangat tinggi.

### 6.2.2 Precision

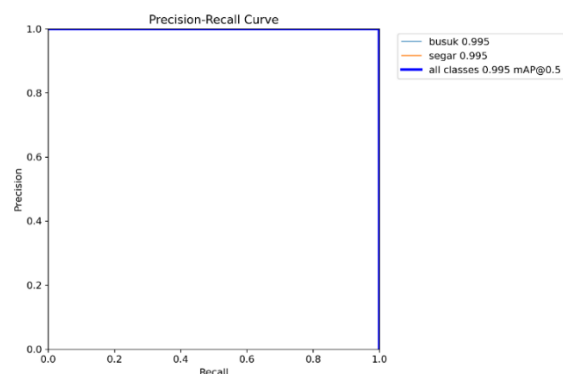


Gbr 10 Precision

Gambar di atas menampilkan kurva Precision-Confidence yang mengevaluasi kinerja model klasifikasi. Kurva ini menunjukkan hubungan antara tingkat kepercayaan prediksi (sumbu horizontal) dan precision (sumbu vertikal) untuk kelas "busuk" (biru muda) dan "segar" (oranye).

Kurva "all classes" (garis tebal biru) mencapai precision 1.00 pada confidence 0.971, menunjukkan performa yang baik. Secara umum, precision mendekati 1 pada sebagian besar tingkat confidence, mengindikasikan kemampuan klasifikasi yang baik.

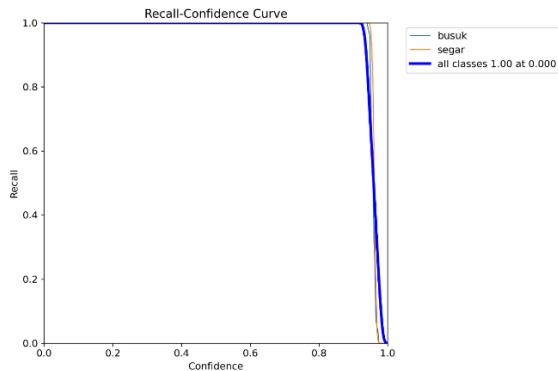
### 6.2.3 Precision-Recall



Gbr 11. Precision-Recall

Grafik validasi di atas menunjukkan bahwa nilai precision dan recall untuk prediksi mencapai sekitar 0.995, menandakan akurasi dan label yang tinggi. Model ini menunjukkan performa hampir sempurna dalam mengidentifikasi dan mengklasifikasikan objek sesuai label.

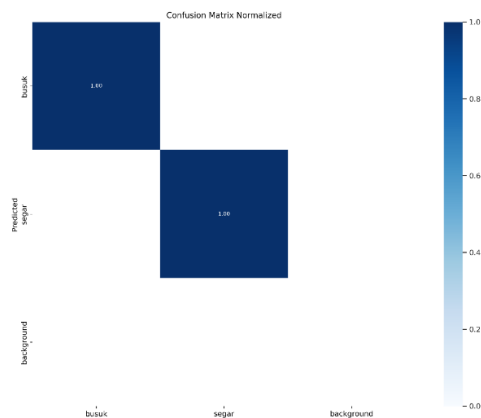
#### 6.2.4 Recall



Gbr 12 Recall

Grafik validasi di atas menunjukkan bahwa semua kelas memiliki nilai precision 1 dan recall di atas 0,8. Ini menandakan bahwa model tidak hanya menghasilkan prediksi yang akurat tetapi juga konsisten dalam mengklasifikasikan objek dengan benar. Nilai recall yang tinggi menunjukkan efektivitas model dalam mendeteksi sebagian besar instance dari setiap kelas, menghasilkan performa yang baik.

#### 6.2.5 Confusion Matrix



Gbr 13. Confusion Matrix

Gambar di atas menampilkan confusion matrix untuk dua kelas: "busuk" dan "segar". Sumbu horizontal menunjukkan kelas sebenarnya, sedangkan sumbu vertikal menunjukkan kelas yang

diprediksi. Semua nilai pada matriks menunjukkan prediksi yang benar, dengan nilai 100% pada setiap kelas, menandakan tidak ada kesalahan klasifikasi. Model ini menunjukkan performa klasifikasi yang sempurna dan dapat diandalkan untuk hasil yang konsisten dan akurat.

### 5. KESIMPULAN

Model YOLOv8n dikembangkan untuk mendeteksi kualitas kentang segar dan busuk. Penelitian dimulai dengan pengumpulan 1000 foto kentang, dibagi menjadi 85% untuk pelatihan, 10% untuk validasi, dan 5% untuk pengujian, serta dilabeli menggunakan Roboflow. Setelah augmentasi, dataset meningkat menjadi 2304 foto. Model menunjukkan performa baik dengan precision 99.9%, recall 100%, mAP50 99.5%, dan mAP50-90 97.9% pada evaluasi. Model juga diimplementasikan di website berbasis Flask untuk deteksi video dan webcam secara real-time. Meskipun ada beberapa kesalahan pengenalan kelas, model secara keseluruhan memberikan hasil deteksi yang baik dengan kesalahan minimal.

### DAFTAR PUSTAKA

- [1] A. Fuadi And A. Suharso, "Perbandingan Arsitektur Mobilenet Dan Nasnetmobile Untuk Klasifikasi Penyakit Pada Citra Daun Kentang," *Jipi (Jurnal Ilm. Penelit. Dan Pembelajaran Inform.*, Vol. 7, No. 3, Pp. 701–710, 2022, Doi: 10.29100/Jipi.V7i3.3026.
- [2] I. S. Diva And S. Agmasari, "4 Cara Pilih Kentang Bagus, Jangan Pilih Yang Lembek." [Online]. Available: <https://www.kompas.com/food/read/2023/05/09/161000275/4-Cara-Pilih-Kentang-Bagus-Jangan-Pilih-Yang-Lembek>
- [3] Z. S. Jannah And F. A. Sutanto, "Implementasi Algoritma Yolo ( You Only Look Once ) Untuk Deteksi Rias Adat Nusantara," Vol. 22, No. 3, Pp. 1490–1495, 2022, Doi: 10.33087/Jiubj.V22i3.2421.
- [4] A. Wibowo, L. Lusiana, And T. K. Dewi, "Implementasi Algoritma Deep Learning You Only Look Once (Yolov5) Untuk Deteksi Buah Segar Dan Busuk," *Paspalum J. Ilm. Pertan.*, Vol. 11, No. 1, P. 123, 2023, Doi: 10.35138/Paspalum.V11i1.489.
- [5] S. N. Mashita, "Implementasi Deep Learning Object Detection Rambu K3 Pada Video Menggunakan Metode Convolutional Neural



- Network (Cnn) Dengan Tensorflow,” *Skripsi, Stat. Fak. Mat. Dan Ilmu Pengetah. Alam, Univ. Islam Indones. Yogyakarta*, Pp. 1–89, 2020, [Online]. Available: [https://Dspace.Uii.Ac.Id/Bitstream/Handle/123456789/28781/16611128\\_Syinta\\_Nuri\\_Mashita.Pdf?Sequence=1&Isallowed=Y](https://Dspace.Uii.Ac.Id/Bitstream/Handle/123456789/28781/16611128_Syinta_Nuri_Mashita.Pdf?Sequence=1&Isallowed=Y)
- [6] P. R. Aningtiyas, A. Sumin, And S. Wirawan, “Pembuatan Aplikasi Deteksi Objek Menggunakan Tensorflow Object Detection Api Dengan Memanfaatkan Ssd Mobilenet V2 Sebagai Model Pra-Terlatih,” Vol. 19, No. September, Pp. 421–430, 2020, [Online]. Available: <https://Scholar.Archive.Org/Work/Itmgg3rr5fdh5armvk6ivykb2m/Access/Wayback/Https://Ejournal.Jak-Stik.Ac.Id/Index.Php/Komputasi/Article/Download/68/152>
- [7] I. M. D. Maleh, R. Teguh, A. S. Sahay, S. Okta, And M. P. Pratama, “Implementasi Algoritma You Only Look Once (Yolo) Untuk Object Detection Sarang Orang Utan,” *J. Inform.*, Vol. 10, No. 1, Pp. 19–27, 2023, Doi: 10.31294/Inf.V10i1.13922.
- [8] M. Sarosa And N. Muna, “Implementasi Algoritma You Only Look Once (Yolo) Untuk Deteksi Korban Bencana Alam,” *J. Teknol. Inf. Dan Ilmu Komput.*, Vol. 8, No. 4, Pp. 787–792, 2021, Doi: 10.25126/Jtiik.202184407.
- [9] A. A. G. Bagus Janapriya, “Pengenalan Jenis Rambu Lalu Lintas Menggunakan Metode Yolo V5,” *Jeliku (Jurnal Elektron. Ilmu Komput. Udayana)*, Vol. 11, No. 4, P. 1011, 2023, Doi: 10.24843/Jlk.2023.V11.I04.P32.
- [10] L. Rahma, H. Syaputra, A. H. Mirza, And S. D. Purnamasari, “Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma Yolo (You Only Look Once),” *J. Nas. Ilmu Komput.*, Vol. 2, No. 3, Pp. 213–232, 2021, Doi: 10.47747/Jurnalnik.V2i3.534.
- [11] T. Handayani, “Implementasi Metode You Only Look Once (Yolo) Untuk Deteksi Kesegaran Telur Ayam Berdasarkan Citra Cangkang,” Vol. 3, No. 0, Pp. 1–23, 2024, [Online]. Available: <https://Ejournal.Warunayama.Org/Index.Php/Kohesi/Article/View/4038>
- [12] A. Harun, “Implementasi Deep Learning Menggunakan Metode Youonlylookonce Untuk Mendeteksi Rokok,” *J. Ekon. Vol. 18, Nomor 1 Maret201*, Vol. 2, No. 1, Pp. 41–49, 2023, [Online]. Available: <https://Repository.Uin-Suska.Ac.Id/65514/>
- [13] L. Satya, M. R. D. S. Septian, M. W. Sarjono, M. Cahyanti, And E. R. Swedia, “Sistem Pendeteksi Plat Nomor Polisi Kendaraan Dengan Arsitektur Yolov8,” *Sebatik*, Vol. 27, No. 2, Pp. 753–761, 2023, Doi: 10.46984/Sebatik.V27i2.2374.
- [14] N. J. Hayati, D. Singasatia, And M. R. Muttaqin, “Object Tracking Menggunakan Algoritma You Only Look Once (Yolo)V8 Untuk Menghitung Kendaraan,” *Komputa J. Ilm. Komput. Dan Inform.*, Vol. 12, No. 2, Pp. 91–99, 2023, Doi: 10.34010/Komputa.V12i2.10654.
- [15] M. Ibrahim And U. Latifa, “Penerapan Algoritma Yolov8 Dalam Deteksi Waktu Panen Tanaman Pakcoy Berbasis Website,” *Jati (Jurnal Mhs. Tek. Inform.)*, Vol. 7, No. 4, Pp. 2489–2495, 2024, Doi: 10.36040/Jati.V7i4.7154.