

Implementasi Algoritma Yolo untuk Deteksi Kebusukan pada Sayur Kembang Kol

Alexander Ibrahim^{a1}, I Wayan Supriana^{a2}

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
¹alexander772004@gmail.com
² wayan.supriana@unud.ac.id

Abstract

This research utilizes the YOLOv8 algorithm to detect spoilage in cauliflower vegetables. Image data was collected from Google, processed using Roboflow, and tested using Google Colab. The study results indicate an accuracy of 59%, recall of 58%, and MAP of 60%. The YOLOv8 algorithm significantly contributes to image recognition and visual data processing. Additionally, the article discusses the application of the YOLOv8 algorithm for object detection in 360-degree panoramic images. The training process was conducted to recognize objects in the images, and evaluation was performed using a confusion matrix and mAP50. The evaluation results demonstrate the model's good performance in object recognition. Several references cited in the article are also included.

Keywords: You Only Look Once (YOLO), Google Colab, Roboflow

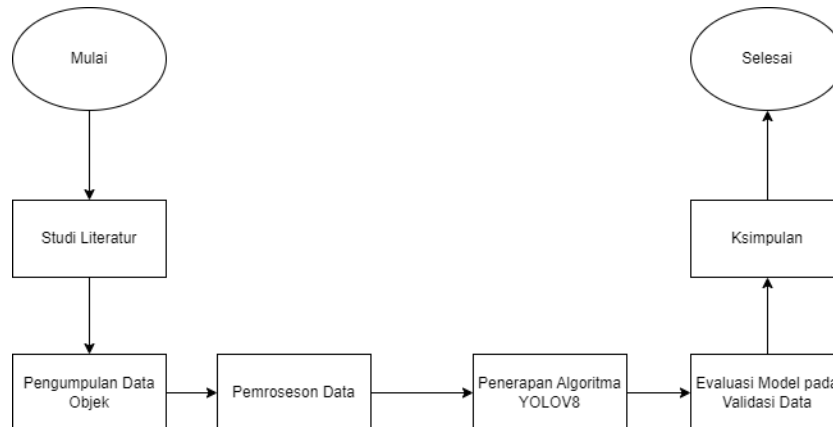
1. Pendahuluan

Sayur kembang kol merupakan salah satu komoditas pertanian yang populer di Indonesia. Namun, kembang kol mudah mengalami kebusukan, yang dapat menyebabkan kerugian ekonomi yang signifikan bagi petani dan pedagang. Oleh karena itu, penting untuk mengembangkan metode yang efektif untuk mendeteksi kebusukan pada kembang kol sesegera mungkin. Algoritma YOLOv8 telah menjadi salah satu metode yang populer dalam deteksi objek dalam gambar, dengan kemampuannya yang efisien dan andal. Dalam penelitian ini, algoritma YOLOv8 digunakan untuk dua tujuan yang berbeda namun terkait. Pertama, penelitian ini bertujuan untuk mendeteksi kebusukan pada sayur kembang kol, sementara penelitian kedua fokus pada deteksi objek dalam gambar panorama 360 derajat. Kedua penelitian ini mengevaluasi kinerja model menggunakan metrik yang berbeda, seperti akurasi, recall, dan Mean Average Precision (MAP), serta confusion matrix dan mAP50. Hasil dari kedua penelitian menunjukkan bahwa algoritma YOLOv8 memberikan kontribusi yang signifikan dalam pengenalan objek dalam gambar dan pemrosesan data visual. Dengan demikian, penelitian ini memberikan wawasan yang berharga tentang penggunaan algoritma YOLOv8 dalam konteks deteksi objek pada berbagai jenis gambar.

2. Metode Penelitian

Penelitian ini bersifat eksperimental dengan objek penelitian yaitu kembang kol yang diperoleh dengan cara mengambil foto secara langsung dari Google. Citra yang telah didapatkan selanjutnya akan diproses melalui beberapa tahapan seperti anotasi, pra- pemrosesan, augmentasi (pelebaran dataset), serta mengeksport dataset ke dalam format YOLOv8. Proses tersebut dilakukan secara otomatis dengan memanfaatkan platform roboflow. Setelah dataset ter-ekspor ke dalam format YOLOv8 langkah selanjutnya masuk ke platform Google Collab memanggil model YOLOv8 dengan cara menjalankan code cloud, kemudian mengupload show donload code. Kemudian, dilanjutkan ke beberapa proses yaitu pelatihan, pendeteksian dan evaluasi dataset. Output dari seluruh proses ini adalah pemodelan dataset dengan nilai confusion metric seperti akurasi, recall, precision, serta f1 score.

Berikut merupakan tahapan dalam metodologi penelitian yang dijelaskan melalui diagram alur penelitian dan dapat dilihat pada Gambar 1.



Gambar 1. Diagram Alur Penelitian

a. Studi Literatur

Pada tahap ini mempelajari berbagai referensi atau rujukan yang mendukung penelitian bersumber dari internet, jurnal, tugas akhir, penelitian dan modul -modul yang berkaitan dengan penerapan algoritma YOLOv8 untuk sistem deteksi pada dataset baru citra kembang kol serta penelitian - penelitian sejenis.

b. Pengumpulan Gambar

Tahap ini merupakan persiapan yang harus dilakukan sebelum melakukan penelitian. Tahap yang dimaksud adalah pengumpulan gambar sayur kembang kol. Data yang tersaji merupakan data primer yang diambil dari mesin Google. Gambar yang diambil terdiri dari 3 tingkat yaitu segar, mulai membusuk, dan juga busuk.

c. Pemrosesan Data

Setelah data mentah dikumpulkan, langkah selanjutnya adalah tahap pemrosesan data. Data yang sudah ada dimasukkan ke dalam roboflow untuk menjalani beberapa proses seperti:

- Anotasi gambar atau pemberian nama gambar berdasarkan tingkatannya yaitu segar, mulai membusuk, dan juga busuk.
- Pra pemrosesan data dilakukan dengan mengekstrak bagian - bagian yang tidak penting seperti latar belakang yang tidak relevan, objek orientasi otomatis sekaligus mengubah ukuran menjadi 640 x 640 piksel.
- Ekspor data ke dalam format YOLOv8

d. Penerapan Algoritma YOLOV8

Setelah data melewati proses augmentasi, tahap selanjutnya adalah melakukan proses penerapan algoritma YOLOv8 pada data tindakan ini merupakan proses training data yang ada untuk mengupload file Lantana zip dan data yaml ke dalam codingan Google Collab lalu melakukan proses runing di dalam coding setelah itu menginput nilai epoch berapa kali data akan dilatih. Setelah hasil nilai keluar lanjut ke proses pendeteksian dengan cara memasukan gambar yang ingin dideteksi ke codingan dalam Google Collab setelah itu model akan segera membaca dan menampilkan hasil deteksi. Setelah objek benar maka tahap selanjutnya melihat dan mencocokkan hasilnya. Jika objek yang ada sudah terdeteksi sama dengan jenis warna pada lantana camara untuk tiap klasifikasi atau kelasnya, maka dapat dikatakan proses training nya berhasil

e. Evaluasi Model pada Validasi Data

Setelah data melewati proses augmentasi, tahap selanjutnya adalah melakukan proses penerapan algoritma YOLOv8 pada data tindakan ini merupakan proses training data yang ada untuk mengupload show donload code ke dalam codingan Google Collab lalu melakukan proses runing di dalam coding, setelah itu menginput nilai epoch berapa kali data akan dilatih. Setelah hasil nilai keluar lanjut ke proses pendeteksian dengan cara memasukan gambar yang ingin dideteksi ke codingan dalam Google Collab setelah itu model akan segera membaca dan menampilkan hasil deteksi. Setelah objek benar maka tahap selanjutnya melihat dan mencocokkan hasilnya. Jika objek yang ada sudah terdeteksi sama dengan jenis warna pada lantana camara untuk tiap klasifikasi atau kelasnya, maka dapat dikatakan proses trainingnya berhasil.

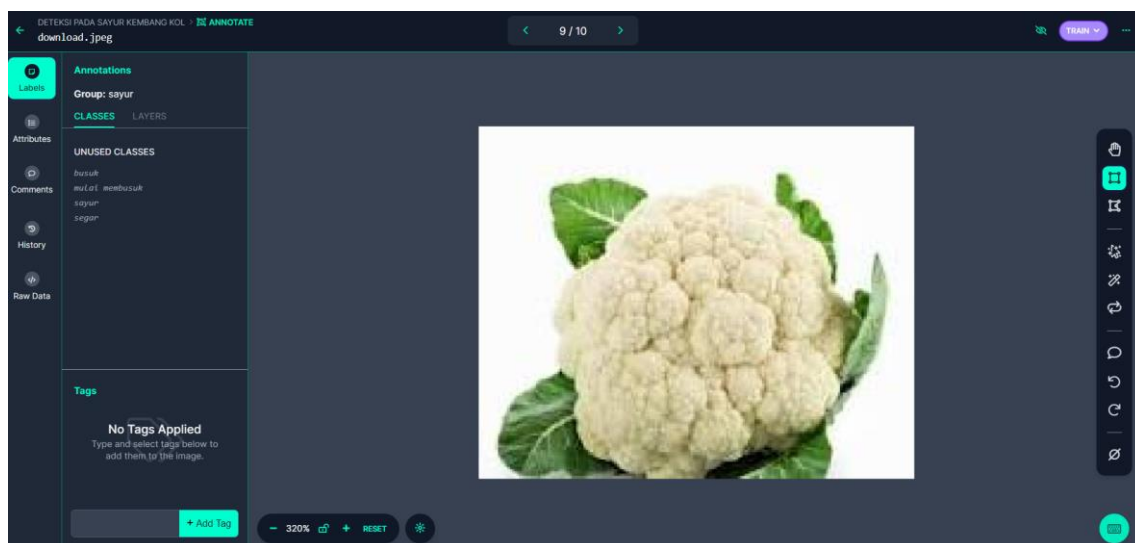
f. Kesimpulan

Tahap ini model melakukan prediksi pada validasi data, dan hasil prediksi bisa dibandingkan dengan label sebenarnya. Evaluasi ini memberikan wawasan tentang seberapa baik model dapat melakukan prediksi pada data yang belum pernah dilihat sebelumnya hasil ini dapat dilihat dari hasil best epoch dengan nilai rata-rata accuracy, precision, recall dan MAP yang dihasilkan oleh Google Collab.

3. Hasil dan Diskusi

3.1 Implementasi YOLOv8

Tahap Implementasi YOLO dimulai dengan mempersiapkan dataset jenis sayur yaitu kembang kol dengan total gambar sebanyak 10 gambar. Gambar yang dikumpulkan kemudian dimasukan ke dalam roboflow dilakukan beberapa perlakuan seperti anotasi gambar atau pemberian nama kelas pada objek, hal ini dapat dilihat pada Gambar 2



Gambar 2. Contoh Anotasi Objek

Setelah melalui tahap anotasi lanjut ke tahap preprocessing di mana tahapan ini dilakukan dengan cara mengekstrak bagian - bagian yang tidak penting seperti latar belakang yang tidak relevan, objek orientasi diatur otomatis sekaligus mengubah ukuran menjadi 640 x 640 piksel. Seperti yang diperlihatkan pada Gambar 3.

```
# Pip install method (recommended)

!pip install ultralytics==8.0.196

from IPython import display
display.clear_output()

import ultralytics
ultralytics.checks()
```

Gambar 3. Tampilan source code untuk mengunduh YOLOv8

Setelah mengunduh YOLOv8 ke dalam Google Collab, selanjutnya mengunduh Show Download Code yang didapatkan di Platform roboflow. Hal ini perlu dilakukan agar memungkinkan Google Collab untuk mengakses ke data atau berkas yang diperlukan untuk melatih ataupun menguji model. Hal ini dapat dilihat pada Gambar 4 dimana Show Download Code.

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="4A9p4v14qAt2CxYGgxXb")
project = rf.workspace("alex-yxgvp").project("deteksi-pada-sayur-kembang-kol")
version = project.version(1)
dataset = version.download("yolov8")
```

Gambar 4. Tampilan Show Download Code

Langkah selanjutnya masuk ke tahap pelatihan data yang dapat dilihat pada Gambar 5, dimana tahap! yolo task=detect mode=train model=YOLOv8s.pt data= {dataset. location}/data.yaml epochs=25 imgsz=800 plots=True, bagian ini menggunakan program YOLOv8 untuk melaksanakan training data. model=YOLOv8s.pt, menentukan model awal yang akan digunakan untuk pelatihan. Dalam hal ini, YOLOv8s.pt adalah model YOLOv8 versi ringan yang sudah disediakan. data= {dataset. location}/data.yaml, menentukan lokasi file konfigurasi data pelatihan Anda. Ganti {dataset. location} dengan nama folder aktual dataset Anda. File. yaml ini berisi informasi penting tentang dataset, seperti lokasi gambar, label objek, dan lainnya. epochs=25, menentukan jumlah iterasi pelatihan. Model akan dilatih selama 25 iterasi. imgsz=800, menentukan ukuran gambar yang akan digunakan selama pelatihan. Dalam hal ini, gambar akan diubah ukurannya menjadi 800x800 piksel. plots=True, menampilkan grafik pelatihan selama proses berlangsung.

```
%cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=800 plots=True
```

Gambar 5. Tampilan Source Code Untuk Training Data

```
data.yaml X
1 names:
2 - busuk
3 - mulai membusuk
4 - segar
5 nc: 3
6 roboflow:
7   license: CC BY 4.0
8   project: deteksi-pada-sayur-kembang-kol
9   url: https://universe.roboflow.com/alex-yxgvp/deteksi-pada-sayur-kembang-kol/dataset/1
10  version: 1
11  workspace: alex-yxgvp
12 test: test/images
13 train: train/images
14 val: valid/images
15
```

Gambar 6. Tampil Data Yaml

Setelah melakukan training data, langkah berikutnya melakukan pendeteksian objek seperti terlihat pada Gambar 7. Proses ini diperlukan untuk menguji apakah data train berhasil dikenali oleh model atau tidak. Proses ini dilakukan dengan menjalankan perintah python task=detect. Untuk %cd {HOME}, Ini akan mengubah direktori kerja ke direktori home Anda di Google Colab. ! yolo task=detect mode=predict model={HOME}/runs/detect/train/weights/best.pt conf=0.25 source={dataset.location}/test/images save=True, bagian ini menggunakan program YOLOv8 untuk melakukan deteksi objek. Deteksi objek, mencari dan mengenali objek tertentu dalam gambar. Mode prediksi, menggunakan model yang sudah dilatih sebelumnya untuk memprediksi objek dalam gambar baru. model={HOME}/runs/detect/train/weights/best.pt, menentukan lokasi file bobot model terlatih Anda. Bobot model, parameter yang dipelajari model selama proses pelatihan dan berisi informasi penting untuk melakukan deteksi objek. conf=0.25, mengatur ambang kepercayaan minimum. Model hanya akan menampilkan deteksi dengan skor kepercayaan minimal 25% (0.25). Skor kepercayaan, Nilai antara 0 dan 1 yang menunjukkan seberapa yakin model terhadap deteksi objek. source= {dataset. location}/test/images, Menentukan lokasi gambar yang ingin Anda deteksi objeknya. Ganti {dataset. location} dengan nama folder aktual dataset Anda. save=True, menyimpan hasil deteksi ke dalam gambar baru. Gambar ini akan memiliki kotak pembatas yang digambar di sekitar objek yang terdeteksi beserta labelnya.

```
%cd {HOME}
!yolo task=detect mode=predict model={HOME}/runs/detect/train/weights/best.pt conf=0.25 source={dataset.location}/test/images save=True
```

Gambar 7. Tampilan Source Code Untuk Menjalankan Deteksi Objek

3.2 Performance Metric

Evaluasi pelatihan dengan epoch atau iterasi 50 dapat dilihat pada Gambar 8, dimana pada ketiga kelas dengan nilai image 360 merujuk pada gambar panorama 360 derajat. Gambar panorama 360 derajat adalah gambar yang mencakup seluruh lingkaran visual sehingga dapat dilihat ke segala arah secara vertikal maupun horizontal. Untuk jumlah instances pada setiap kelas berbeda-beda karena model deteksi objek dilatih untuk mengenali dan menempatkan objek-objek yang muncul dalam gambar panorama 360 derajat sehingga dari masing-masing kelas secara berturut-turut model mampu mengenali sebanyak 3, 1, 2, dimana hasil ini bergantung pada karakteristik objek, kerumitan dataset, ukuran dataset pelatihan, kualitas anotasi, parameter pelatihan seperti jumlah Epoch dapat mempengaruhi jumlah instances. Untuk mAP50 (mean average precision) atau metrik yang mencakup precision dan recall pada bagian ambang batas, mAP dihitung dengan cara menghitung nilai rata-rata dari precision sehingga mAP pada

rentang nilai ambang deteksi 0,5 hingga 0,95 (mAP50-95) dapat memberikan gambaran lebih tentang evaluasi kinerja model secara menyeluruh di berbagai tingkat ambang deteksi, sehingga nilai mAP yang didapat pada masing-masing kelas di atas 0.22%.

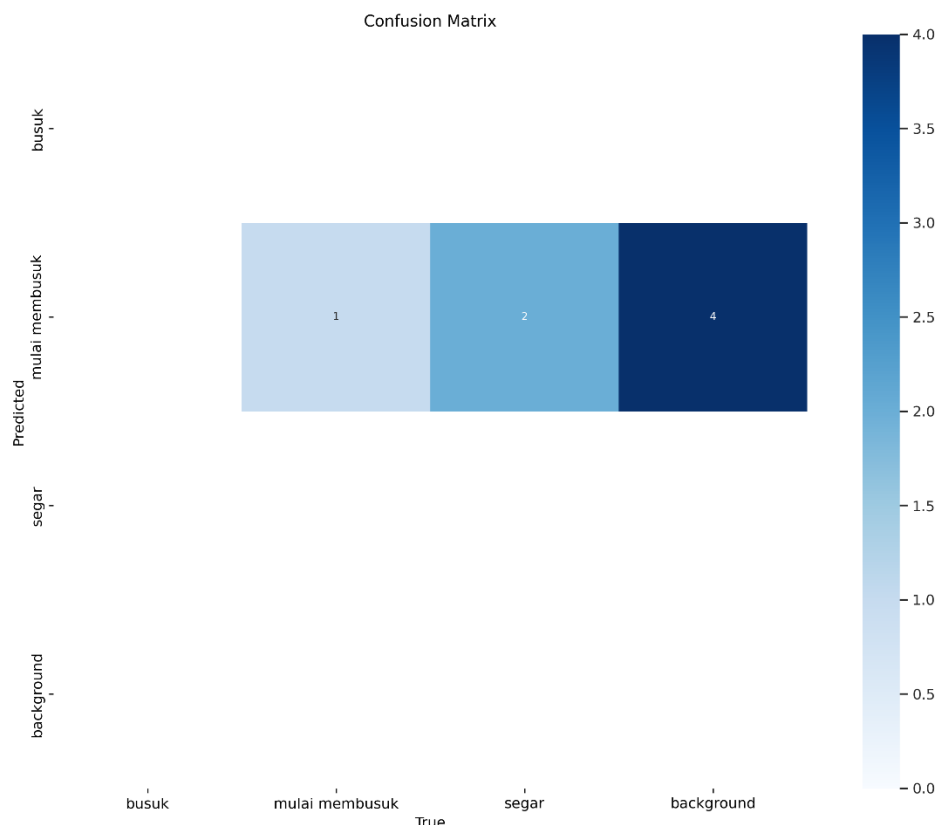
```

%cd {HOME}
lyolo task-detect mode-val model={HOME}/runs/detect/train/weights/best.pt data={dataset.location}/data.yaml

/content
Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11126745 parameters, 0 gradients, 28.4 GFLOPs
val: Scanning /content/datasets/deteksi-pada-sayur-kembang-kol-1/valid/labels.cache... 3 images, 0 backgrounds, 0 corrupt: 100% 3/3 [00:00<, ?it/s]
/usr/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithreaded code, and JAX is multithreaded, so this self.pid = os.fork()
Class      Images  Instances  Box(P      R      mAP50  mAP50-95): 100% 1/1 [00:00<00:00, 1.35it/s]
  all         3         3    0.749    0.497    0.663    0.242
  mulai membusuk  3         1    0.497    0.995    0.497    0.298
  segar       3         2         1         0    0.828    0.186
Speed: 0.4ms preprocess, 36.0ms inference, 0.0ms loss, 180.4ms postprocess per image
Results saved to runs/detect/val
Learn more at https://docs.ultralytics.com/modes/val
    
```

Gambar 8. Tampilan Parameter Hasil Training Dengan Google Collab

Hasil evaluasi confusion matrix yang digunakan untuk membandingkan hasil prediksi model dengan ground truth (kebenaran sebenarnya) yang didapat dari perlakuan train data train data, Terdapat 3 jenis yaitu segar, mulai membusuk dan juga busuk. Untuk yang bernilai true positive (TP), model membuat prediksi benar sesuai dengan kenyataan ditandai dengan Predicted dan True sama, masing-masing kelas bernilai 1, 2, dan 4. Hal ini dapat dilihat pada Gambar 9 tampilan confusion matrix.



Gambar 9. Funfusion Matrix

4. Kesimpulan

Penelitian ini menggunakan algoritma YOLOv8 untuk deteksi objek dalam gambar dengan hasil yang menjanjikan. Penelitian pertama fokus pada deteksi kebusukan pada sayur kembang kol dengan akurasi 59%, recall 58%, dan MAP 60%. Sementara itu, penelitian ini juga membahas deteksi objek dalam gambar panorama 360 derajat dengan evaluasi menggunakan confusion

matrix dan mAP50, menunjukkan kinerja model yang baik. Algoritma YOLOv8 terbukti memberikan kontribusi penting dalam pengenalan gambar dan pemrosesan data visual dalam kedua konteks penelitian.

Daftar Pustaka

- [1] A. Muliantara and N. A. S. ER, "Pengembangan Pengklasifikasi jenis Tanaman Menggunakan pendekatan backpropagation Dan Nguyen-Widrow," *Jurnal Ilmu Komputer*, <https://ojs.unud.ac.id/index.php/jik/article/view/18779> (accessed May 6, 2024).
- [2] Bagus Janapriya, A. A. G. (2023). Pengenalan Jenis Rambu Lalu Lintas menggunakan Metode YOLO V5. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, 11(4). <https://doi.org/10.24843/jlk.2023.v11.i04.p32>.
- [3] Anggotra, P., & Muliantara, A. (2023). Deteksi Relief Candi Borobudur Menggunakan Metode Template Matching. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, 12(1). <https://doi.org/10.24843/jlk.2023.v12.i01.p03>.
- [4] Keraf, M. P. S., Sinlae, A. A. J., & Batarius, P. (2024). Pemodelan Warna Pada Dataset Baru Citra Bunga Lantana Camara Menggunakan Algoritma Yolov5. *Jurnal Mnemonic*, 7(1), 108-117.
- [5] Saputra, D. H., & Imran, B. (2023). Object Detection Untuk Mendeteksi Citra Buah-Buahan Menggunakan Metode Yolo. *Jurnal Kecerdasan Buatan dan Teknologi Informasi*, 2(2), 70-80.
- [6] Lusiana, L., Wibowo, A., & Dewi, T. K. (2023). Implementasi Algoritma Deep Learning You Only Look Once (YOLOv5) Untuk Deteksi Buah Segar Dan Busuk. *Paspalum: Jurnal Ilmiah Pertanian*, 11(1), 123-130.
- [7] HWB, N., Mailoa, E., & Purnomo, H. D. (2020). Deteksi Buah untuk Klasifikasi Berdasarkan Jenis dengan Algoritma CNN Berbasis YOLOv3. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 4(3), 476-481.
- [8] Wulandari, Dwi, et al. "Implementasi Algoritma YOLOv5 untuk Deteksi Kerusakan pada Buah Mangga." *Jurnal Sains dan Teknologi Agroindustri* 8.1 (2023): 1-10.
- [9] Sari, Rini Puspita, et al. "Implementasi Algoritma YOLO untuk Deteksi Kerusakan pada Daun Jeruk." *Jurnal Agroteknologi Universitas Lampung* 16.1 (2021): 1-8.

Halaman ini sengaja dibiarkan kosong