

Implementasi Algoritma Yolo untuk Deteksi Tanaman Apotik Hidup

I Kadek Peri Arta Wijaya^{a1}, I Gede Santi Astawa^{a2}

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
¹periarta2004@gmail.com
²santi.astawa@unud.ac.id

Abstract

Living apothecary is a plant that is used to provide natural treatment or medicine. Therefore, to distinguish these plants, we carry out detection to distinguish live pharmaceutical plants. In this research we used (You Only Look Once) YOLOv8, Google Colab, Roboflow. Where we get the database from Google then we process the image using Roboflow then we create (You Only Look Once) YOLOv8, and we test it on Google Colab. The results of this research confirm that the YOLOv8 (You Only Look Once) algorithm in identifying and modeling 13 plant objects with a total of 1204 images obtained a precision percentage value of 59%, recall of 58%, and a MAP value of 60%, so the average value the accuracy is above 59%. In addition, it provides important contributions in the field of image recognition and visual data processing obtained through the application of the YOLOv8 algorithm to the problem of identifying objects in images.

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

1. Pendahuluan

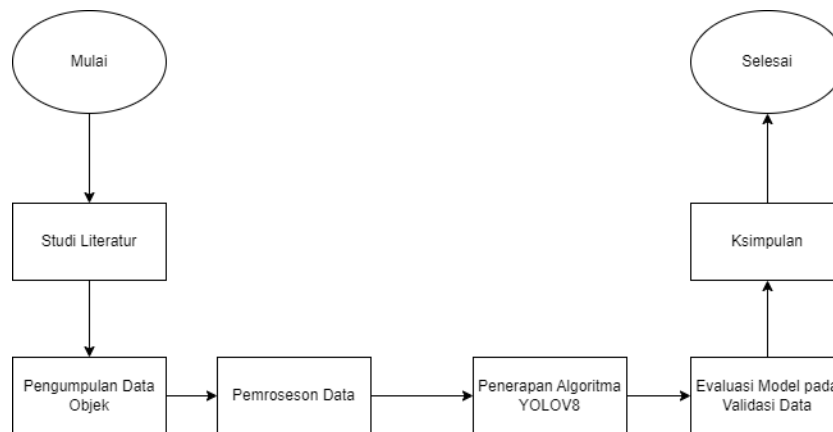
Perkembangan dunia semakin pesat, canggih dan dunia Kesehatan sangat maju dan modern. Salah satu obat-obatan yang mudah ditemui khususnya di Indonesia dan masih banyak digunakan oleh Masyarakat Indonesia adalah Apotik Hidup. Untuk mengetahui tanaman Apotik Hidup kita dapat melakukan penelitian yang terkait. Dari hasil penelitian didapatkan bahwa pemanfaatan Nguyen Widrow telah berhasil meningkatkan akurasi backpropagation sebanyak 25%. Akurasi klasifikasi dari aplikasi yang telah dikembangkan rata-rata di atas 83%. Hal ini mengindikasikan aplikasi yang telah dibangun dapat menjadi salah satu solusi untuk hal klasifikasi jenis tanaman [1]. Penelitian ini mengembangkan metode untuk melakukan deteksi jenis rambu lalu lintas menggunakan metode YOLO V5 yang merupakan salah satu pengembangan dari metode CNN. Nilai identifikasi model yang diperoleh diantaranya precision sebesar 0,923, recall sebesar 0,826, mAP50 sebesar 0,965, dan mAP50-95 sebesar 0,924 [2]. Hasil pengujian relief yang dilakukan dengan menggunakan metode Minimum Square Difference mencapai akurasi sebesar 84,31% sehingga dapat dikatakan cukup akurat untuk sistem deteksi relief Candi Borobudur. Sedangkan dengan menggunakan metode Normalized Cosine Coefficient persentase keberhasilan sistem hanya 23,52%. Hal ini menyebabkan metode Normalized Cosine Coefficient tidak dianjurkan untuk digunakan dalam sistem ini [3]. Hasil penelitian ini menegaskan bahwa algoritma YOLOv5 efektif dalam mengidentifikasi dan memodelkan variasi warna pada 1266 citra bunga *Lantana camara* melalui putaran pelatihan sebanyak 50 epoch dan membutuhkan waktu selama 0.483 jam [4]. *You Only Look Once (YOLO)* merupakan salah satu model Deep Learning yang dapat digunakan untuk pengenalan objek. Penelitian ini bertujuan untuk pengenalan objek pada citra buah menggunakan YOLO [5]. Berdasarkan pengujian yang telah dilakukan dengan data training, beserta data validasi, serta data test menggunakan algoritma YOLOv5 dapat disimpulkan bahwa metode pendeteksian ini dapat mengenali objek secara konsisten dengan tingkat akurasi yang tinggi [6]. Hasil dari penelitian ini dapat mengenali jenis bunga anggrek pada gambar. Hasil pengujian dari dataset peneliti menunjukkan akurasi sebesar 30.7% untuk color space grayscale, 37% untuk color space RGB, 34.6% untuk color space HSI, 41% untuk color space YIQ, dan 40.2% untuk color space oRGB dalam mengenali jenis bunga anggrek [7].

Penerapan deteksi 10 buah ini dilakukan pada google colaboratory melalui citra dengan dua kali pengujian. Akurasi dalam deteksi 10 buah ini bisa mencapai lebih dari 90% pada pengujian pertama dari setiap buah dan rata - rata 70% pada pengujian kedua untuk citra diluar data test [8].

2. Metode Penelitian

Penelitian ini bersifat eksperimental dengan objek penelitian yaitu tanaman Apotik Hidup (Lidah Buaya, Daun Sirih, Lengkuas, Temulawak, Jintan, Jahe, Bawang Merah, Bawang Putih, Daun Salam, Jambu Biji, Kunyit, Jeruk Nipis, Tomat) 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 mengekspor 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 tanaman apotik hidup serta penelitian - penelitian sejenis.

b. Pengumpulan Gambar

Tahap ini merupakan persiapan yang harus dilakukan sebelum melakukan penelitian. Tahap yang dimaksud adalah pengumpulan gambar tanaman apotik hidup. Data yang tersaji merupakan data primer yang diambil dari mesin Google. Gambar yang diambil terdiri dari 13 jenis tanaman yaitu Lidah Buaya, Daun Sirih, Lengkuas, Temulawak, Jintan, Jahe, Bawang Merah, Bawang Putih, Daun Salam, Jambu Biji, Kunyit, Jeruk Nipis, Tomat.

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 13 jenis tanaman yaitu

Lidah Buaya, Daun Sirih, Lengkuas, Temulawak, Jintan, Jahe, Bawang Merah, Bawang Putih, Daun Salam, Jambu Biji, Kunyit, Jeruk Nipis, Tomat.

- 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 donlowad 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.

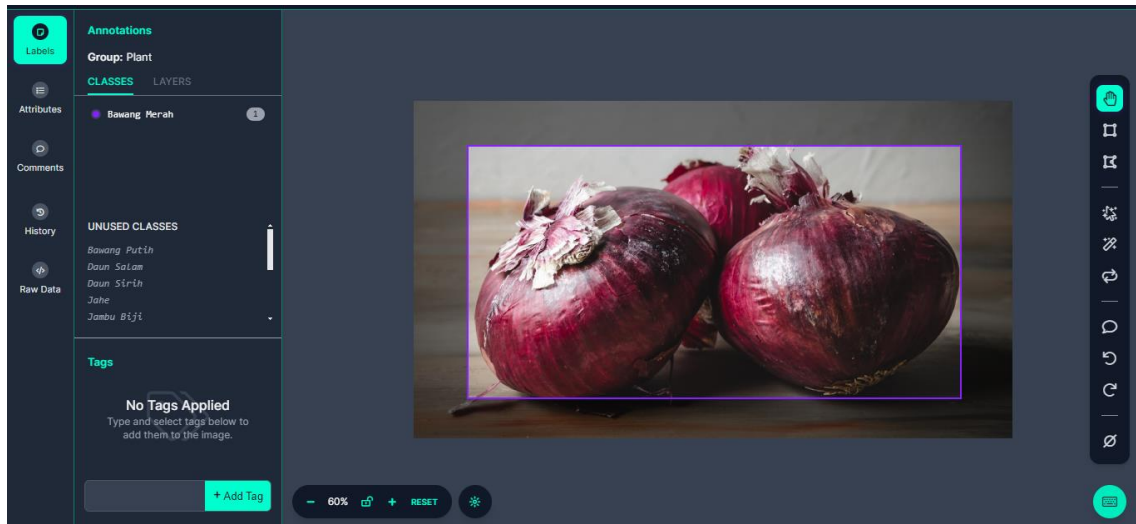
f. Kesimpulan

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 13 jenis tanaman yaitu Lidah Buaya, Daun Sirih, Lengkuas, Temulawak, Jintan, Jahe, Bawang Merah, Bawang Putih, Daun Salam, Jambu Biji, Kunyit, Jeruk Nipis, Tomat. Dengan total gambar sebanyak 1204 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

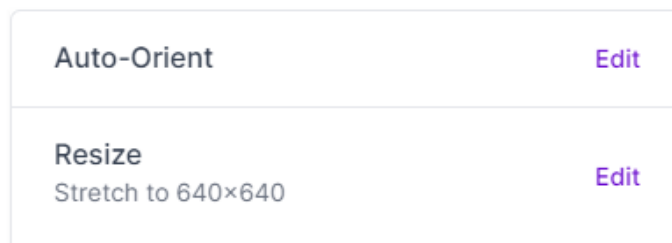


Gambar 2. Contoh Anotasi Objek

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.

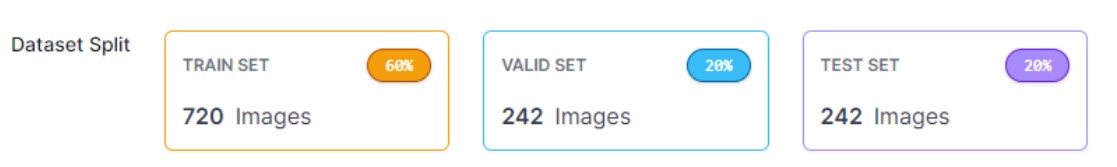
Preprocessing

Decrease training time and increase performance by applying image transformations to all images in this dataset.



Gambar 3. Tampilan Gambar *Presprocessing*

Contoh tampilan dataset yang dibagi dengan menggunakan *platform roboflow* dapat dilihat pada Gambar 4. Dataset dibagi dengan proporsi 60% untuk Train set, 20% untuk Valid set, dan sisanya 20% untuk Test set. Proporsi pembagian ini merujuk pada penelitian terdahulu yang berhasil dilakukan oleh Keraf, dkk tahun 2024[1]



Gambar 4. Tampilan Dataset Split

Setelah dataset disiapkan langkah selanjutnya masuk ke *platform* Google Collab, dan lanjut melakukan cloning, dimana proses ini menggambarkan sebagian source code yang digunakan untuk memanggil script algoritma *YOLOv8* dalam laman pekerjaan di Google Collab. Hal ini

terjadi karena sifat Google Collab yang bekerja secara runtime (sesaat) yang tidak memerlukan ruang penyimpanan dan pesifikasi laptop yang canggih langkah ini dapat dilihat pada Gambar 5.

```
# Pip install method (recommended)

!pip install ultralytics==8.0.196

from IPython import display
display.clear_output()

import ultralytics
ultralytics.checks()
```

Gambar 5. Tampilan source code untuk mengunduh *YOLOv8*

Setelah mengunduh *YOLOv8* ke dalam Google Collab, selanjutnya mengunduh Show Donlowad 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 6 dimana Show Donlowad Code.

```
!mkdir {HOME}/datasets
%cd {HOME}/datasets

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="g6ukztezdSZKZnMrTyZN")
project = rf.workspace("belajar-slj1c").project("live-pharmacy")
version = project.version(2)
dataset = version.download("yolov8")
```

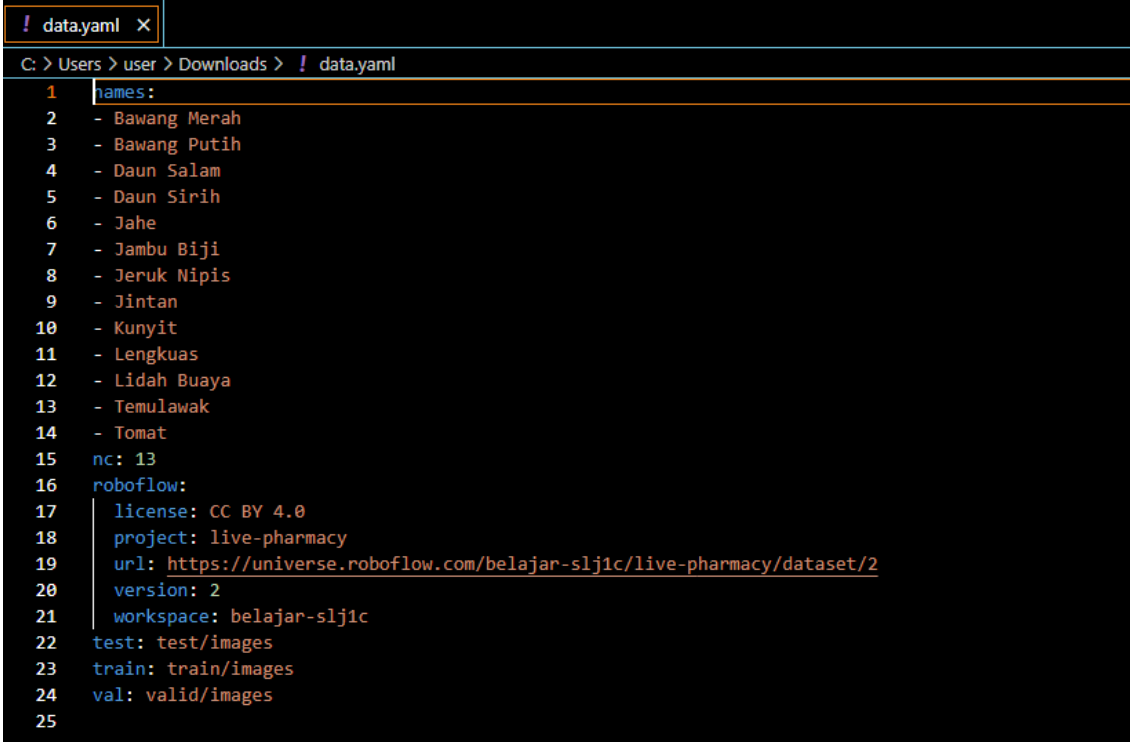
Gambar 6. Tampilan *Show Donlowad Code*

Langkah selanjutnya masuk ke tahap pelatihan data yang dapat dilihat pada Gambar 7, 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
```

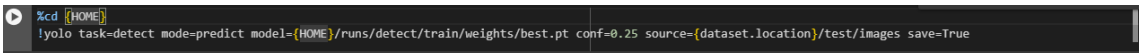
Gambar 7. Tampilan *Source Code Untuk Training Data*



```
1 names:
2   - Bawang Merah
3   - Bawang Putih
4   - Daun Salam
5   - Daun Sirih
6   - Jahe
7   - Jambu Biji
8   - Jeruk Nipis
9   - Jintan
10  - Kunyit
11  - Lengkuas
12  - Lidah Buaya
13  - Temulawak
14  - Tomat
15  nc: 13
16  roboflow:
17    license: CC BY 4.0
18    project: live-pharmacy
19    url: https://universe.roboflow.com/belajar-slj1c/live-pharmacy/dataset/2
20    version: 2
21    workspace: belajar-slj1c
22  test: test/images
23  train: train/images
24  val: valid/images
25
```

Gambar 8. Tampil Data Yaml

Setelah melakukan training data, langkah berikutnya melakukan pendeteksian objek seperti terlihat pada Gambar 9. 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 9. Tampilan Source Code Untuk Menjalankan Deteksi Objek

3.2. Performance Metric

Evaluasi pelatihan dengan epoch atau iterasi 50 dapat dilihat pada Gambar 10, dimana pada 13 kelas jenis tanaman yaitu Lidah Buaya, Daun Sirih, Lengkuas, Temulawak, Jintan, Jahe, Bawang Merah, Bawang Putih, Daun Salam, Jambu Biji, Kunyit, Jeruk Nipis, Tomat, 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 36, 22, 15, 13, 25, 42, 20, 19, 8, 12, 15, 29, 113, 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 0.27%.

```

/content
Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11130615 parameters, 0 gradients, 28.5 GFLOPs
val: Scanning /content/datasets/Live-Pharmacy-2/valid/labels.cache... 242 images, 0 backgrounds, 0 corrupt: 100% 242/242 [00:00<?, ?i
/usr/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithrea
self.pid = os.fork()
Class      Images  Instances  Box(P      R      mAP50  mAP50-95): 100% 16/16 [00:08<00:00, 1.96it/s]
all        242     369        0.596     0.585  0.608  0.273
Bawang Merah  242     36         0.54     0.361  0.399  0.198
Bawang Putih  242     22         0.811     0.682  0.7    0.382
Daun Salam   242     15         0.623     0.8    0.784  0.375
Daun Sirih   242     13         0.776     0.801  0.896  0.467
Jahe        242     25         0.545     0.64   0.604  0.266
Jambu Biji   242     42         0.706     0.686  0.73   0.367
Jeruk Nipis  242     20         0.786     0.737  0.805  0.305
Jintan      242     19         0.61     0.684  0.695  0.206
Kunyit      242     8          0.122     0.375  0.177  0.0414
Lengkuas    242     12         0.545     0.583  0.476  0.215
Lidah Buaya  242     15         0.422     0.487  0.547  0.23
Temulawak   242     29         0.67     0.276  0.555  0.267
Tomat       242     113        0.596     0.496  0.535  0.227
Speed: 2.8ms preprocess, 14.2ms inference, 0.0ms loss, 4.8ms postprocess per image
Results saved to runs/detect/val
Learn more at https://docs.ultralytics.com/modes/val
    
```

Gambar 10. Tampilan Parameter Hasil Training Dengan Google Collab

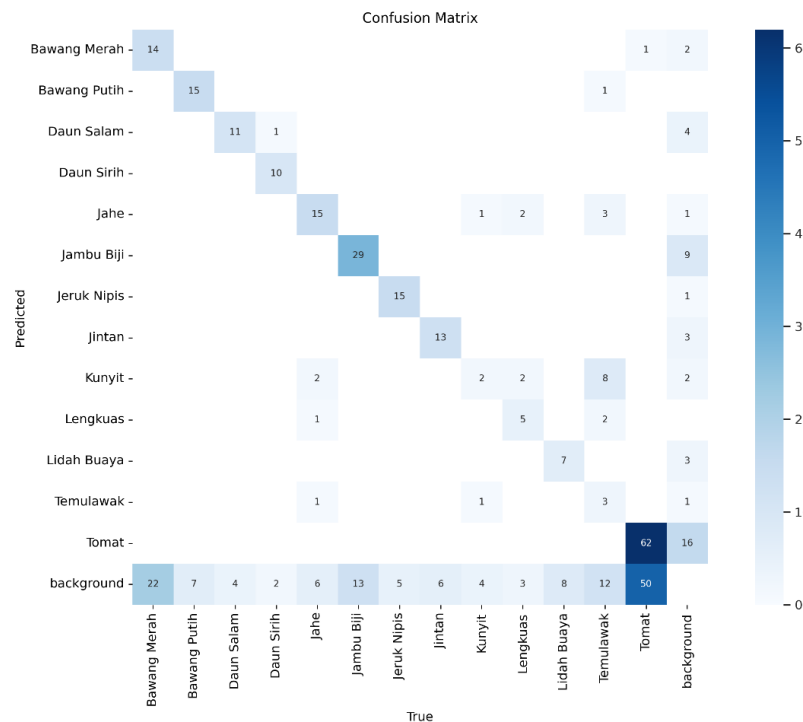
Ringkasan dari hasil ini dapat dilihat pada Tabel 1, pada kolom precision terdapat nilai masing - masing kelas 0.54, 0.811, 0.623, 0.776, 0.545, 0.706, 0.786, 0.61, 0.122, 0.545, 0.422, 0.67, 0.595. Nilai ini merupakan hasil pengukuran deteksi positif yang dilakukan oleh model adalah benar - benar relevan atau akurat. Ini memberikan informasi tentang seberapa baik model membatasi jumlah false positive, yaitu objek yang sebenarnya tidak ada tetapi model mendeteksinya.

Tabel 1. Rincian Hasil Training Model Epoch 50

Kelas	Precision	recall	mAP50
Bawang merah	0.54	0.361	0.399
Bawang putih	0.811	0.682	0.7
Daun salam	0.623	0.8	0.784
Daun sirih	0.776	0.801	0.896
Jahe	0.545	0.64	0.604
Jambu biji	0.706	0.686	0.73
Jeruk nipis	0.786	0.737	0.805
Jintan	0.61	0.684	0.695
Kunyit	0.122	0.375	0.177
Lengkuas	0.545	0.583	0.476
Lidah buaya	0.422	0.487	0.547

Kelas	Precision	recall	mAP50
Temulawak	0.67	0.276	0.555
Tomat	0.595	0.496	0.535

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 13 kelas jenis tanaman yaitu Lidah Buaya, Daun Sirih, Lengkuas, Temulawak, Jintan, Jahe, Bawang Merah, Bawang Putih, Daun Salam, Jambu Biji, Kunyit, Jeruk Nipis, Tomat.. Untuk yang bernilai true positive (TP), model membuat prediksi benar sesuai dengan kenyataan ditandai dengan Predicted dan True sama, masing-masing kelas bernilai 14, 15, 11, 10, 15, 29, 15, 13, 2, 5, 7, 3, 62. Hal ini dapat dilihat pada Gambar 11 tampilan confusion matrix.



Gambar 11. Funfusion Matrix

4. Kesimpulan

Hasil penelitian ini menegaskan bahwa algoritma YOLOv8 dalam mengidentifikasi dan memodelkan 13 objek tumbuhan dengan total 1204 gambar diperoleh berupa nilai persentase precision sebesar 59%, recall sebesar 58%, dan nilai MAP sebesar 60%, sehingga nilai rata-rata akurasi di atas 59%. Selain itu, memberikan kontribusi penting dalam bidang pengenalan citra dan pengolahan data visual yang diperoleh melalui penerapan algoritma YOLOv8 pada masalah identifikasi objek dalam citra. Agar model dapat lebih General dalam pengenalan objek, disarankan untuk memperluas dataset baik menambah objek Lidah Buaya, Daun Sirih, Lengkuas, Temulawak, Jintan, Jahe, Bawang Merah, Bawang Putih, Daun Salam, Jambu Biji, Kunyit, Jeruk Nipis, Tomat, ataupun menambah objek lain. Hal ini dapat meningkatkan kemampuan model dalam mengenali dan mengklasifikasikan objek yang lebih beragam.

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] Prayogo, D. M., Gunadi, K., & Setyati, E. (2020). Pengenalan Jenis Bunga Anggrek Menggunakan Metode Color Local Binary Pattern dan Support Vector Machine. *Jurnal Infra*, 8(1), 242-248.
 - [8] 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.

Halaman ini sengaja dibiarkan kosong