

**IDENTIFIKASI PENYAKIT DAUN TEH  
MENGUNAKAN METODE *YOU ONLY LOOK ONCE*  
VERSI 8 (YOLOv8)**

**TUGAS AKHIR**

Diajukan sebagai syarat menyelesaikan jenjang strata Satu  
(S-1) di Program Studi Teknik Informatika, Fakultas  
Teknologi Industri, Institut Teknologi Sumatera

Oleh:

**WELLA AMANDA**

**120140057**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SUMATERA  
LAMPUNG SELATAN  
2024**

## LEMBAR PENGESAHAN

Tugas Akhir dengan judul “IDENTIFIKASI PENYAKIT DAUN TEH MENGGUNAKAN METODE *YOU ONLY LOOK ONCE* VERSI 8 (YOLOv8)” adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya, baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Institut Teknologi Sumatera maupun di institusi pendidikan lainnya.

Lampung Selatan, 28 Agustus 2024

Penulis,

PHOTO  
BERWARNA

Wella Amanda

NIM. 120140057

Diperiksa dan disetujui oleh,

Pembimbing

Tanda Tangan

1. Winda Yulita S.Pd., M.Cs.

NIP. 19930727 2022 03 2 022

.....

2. Ilham Firman Ashari, S.Kom., M.T.

NIP. 19930314 201903 1 018

.....

Penguji

Tanda Tangan

1. Meida Cahyo Untoro, S.Kom., M.Kom.

NIP. 19890518 201903 1 011

.....

2. Miranti Verdiana, S.Si., M.Si.

NIP. 19920905 202203 2 008

.....

Disahkan oleh,

Koordinator Program Studi Teknik Informatika

Fakultas Teknologi Industri

Institut Teknologi Sumatera

Andika Setiawan, S.Kom., M.Cs.

NIP. 19911127 2022 03 1 007

## HALAMAN PERNYATAAN ORISINALITAS

**Tugas Akhir dengan judul “Identifikasi Penyakit Daun Teh Menggunakan Metode *You Only Look Once* Versi 8 (Yolov8)” adalah karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan benar.**

**Nama : Wella Amanda**

**NIM : 120140057**

**Tanda Tangan :**

**Tanggal : 28 Agustus 2024**

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Institut Teknologi Sumatera, saya yang bertanda tangan di bawah ini:

Nama : Wella Amanda  
NIM : 120140057  
Program Studi : Teknik Informatika  
Fakultas : Fakultas Teknologi, Produksi dan Industri  
Jenis Karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Sumatera **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

IDENTIFIKASI PENYAKIT DAUN TEH MENGGUNAKAN METODE  
*YOU ONLY LOOK ONCE* VERSI 8 (YOLOv8)

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini, Institut Teknologi Sumatera berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Lampung Selatan  
Pada tanggal 28 Agustus 2024

Yang menyatakan,

Wella Amanda

## **KATA PENGANTAR**

Puji syukur kehadiran Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga penyusunan tugas akhir ini telah terselesaikan dengan baik. Dalam penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. I Nyoman Pugeg Aryantha., selaku Rektor Institut Teknologi Sumatera
2. Bapak Hadi Teguh Yudistira, S.T., Ph.D., selaku Dekan Fakultas Teknologi dan Industri
3. Bapak Andika Setiawan, S.Kom., M.Cs., selaku Koordinator Program Studi Teknik Informatika Institut Teknologi Sumatera
4. Ibu Winda Yulita, M.Cs., selaku Pembimbing Utama yang telah membimbing dan mengarahkan penulis untuk menyelesaikan tugas akhir ini. Penulis juga berterimakasih atas bantuan, saran, inspirasi dan dukungan yang diberikan Ibu Winda yang juga merupakan dosen wali penulis selama menempuh perkuliahan di Institut Teknologi Sumatera.
5. Bapak Ilham Firman Ashari, S.Kom., M.T., selaku Pembimbing Kedua penulis yang telah membantu penulis menyelesaikan tugas akhir.
6. Bapak Meida Cahyo Untoro, S.Kom., M.Kom., dan Ibu Miranti Verdiana, S.Si., M.Si., selaku dosen penguji 1 dan 2 yang telah memberi saran dan masukan untuk perbaikan tugas akhir penulis.
7. Seluruh dosen Program Studi Teknik Informatika yang telah memberikan ilmu, pengetahuan, motivasi dan dukungan kepada penulis selama menempuh pendidikan di Institut Teknologi Sumatera.
8. Kedua orang tua, terutama ayah, inspirator dan motivator utama dan seumur hidup bagi penulis. Orang yang paling menjunjung tinggi pendidikan, selalu memberi dukungan untuk pendidikan putri-putrinya. Terimakasih atas petuah dan nasihat pendidikan yang telah membuat penulis selalu termotivasi untuk terus belajar. Juga kepada ibu yang tidak pernah berhenti mendoakan dan percaya bahwa penulis bisa. Terimakasih kepada dua orang

paling berharga yang telah mendukung dan menghantarkan penulis hingga sampai di ujung pendidikan sarjana ini.

9. Widya Ferary dan Serin, saudara-saudari terbaik yang selalu bisa menjadi tempat berbagi cerita dan mendukung penulis. Semua tingkah dan cerita *random* yang selalu berhasil menghibur dan membuat penulis bertahan. Terimakasih telah menjadi sahabat, teman sekaligus saudara bagi penulis.
10. Lasriama Dabukke S.Si., selaku sahabat terbaik yang telah banyak menginspirasi penulis. Terimakasih juga tetap menjadi teman dari ribuan orang yang telah ditemui di perkuliahan ini.
11. Teman-teman grup Bocil, Salsa, Maulani, Okta, Tegar, Taki, Farhan, Reza, Elang, Winda, Debora yang sudah memberikan suasana keluarga di perantauan yang jauh dari rumah. Terimakasih karena telah menjadi keluarga baru bagi penulis, terimakasih karena sudah menginspirasi, terimakasih karena sudah banyak membantu.
12. Teman-teman HMIF, terutama departemen keprofesian dan predisium Nawasena yang telah memberikan kesempatan kepada penulis untuk lebih mengembangkan diri menjadi pribadi yang lebih baik dan mempercayakan penulis untuk berpartisipasi dan belajar untuk lebih bertanggung jawab.
13. Teman-teman yang tidak bisa saya sebutkan satu persatu, terutama teman-teman Informatika angkatan 20 yang telah menjadi teman penulis selama menempuh pendidikan di perkuliahan.
14. Kepada Institut Teknologi Sumatera yang telah memberikan kesempatan kepada penulis untuk melanjutkan pendidikan sarjana, serta membuka peluang yang amat sangat berharga untuk penulis bisa memperdalam ilmu pengetahuan dan mengembangkan diri.
15. Terakhir untuk saya sendiri, Wella Amanda yang telah berhasil menyelesaikan pendidikan sarjananya. Terimakasih telah berjuang hingga akhir.

Peneliti menyadari terdapat banyak kekurangan pada tugas akhir ini. Oleh karena itu, penulis berharap adanya saran dan kritik untuk membuat tugas akhir ini menjadi lebih baik. Akhir kata penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi siapapun.

## RINGKASAN

Identifikasi Penyakit Daun Teh Menggunakan Metode *You Only Look*

*Once* Versi 8 (Yolov8)

Wella Amanda

Pertanian adalah sektor penting di Indonesia. Sebagai negara tropis, Indonesia memiliki berbagai hasil produksi pertanian lokal seperti padi, kedelai, jagung, kacang tanah, ketela pohon, ubi jalar, kopi, kelapa, cengkeh, tebu, karet, dan teh. Salah satu komoditas unggulan Indonesia adalah teh (*Camellia Sinensis*), dengan produksi nasional mencapai 136.800 ton pada tahun 2022. Sumatera Utara adalah salah satu daerah penghasil daun teh yang dikelola oleh Perseroan Terbatas Perkebunan Nusantara IV (PTPN). Perusahaan ini memiliki luas perkebunan 1018,28 Ha di unit Bah-Butong, Kabupaten Simalungun, dengan rata-rata hasil panen 50-55 ton/hari. Namun, tanaman teh rentan terhadap berbagai penyakit daun yang dapat menurunkan kualitas dan rasa teh dan membutuhkan pestisida untuk melindungi tanaman sejak masih di perkebunan. Proses identifikasi penyakit daun teh selain dapat dilakukan secara manual, dapat juga memanfaatkan penggunaan teknologi modern, seperti model deteksi objek *You Only Look Once* (YOLO).

Model terbaru, YOLOv8, yang dikembangkan oleh Ultralytics, menawarkan akurasi tinggi dalam mendeteksi objek. Penelitian ini bertujuan untuk mengetahui kemampuan model YOLOv8 dalam melakukan identifikasi penyakit daun teh serta mengevaluasi performanya. Data yang digunakan dalam penelitian ini adalah data yang diambil langsung di PTPN IV Unit Bah-Butong yang kemudian akan divalidasi oleh pakar / ahli dari BRIN. Model yang dibangun menggunakan arsitektur YOLOv8 asli dengan hanya perubahan pada konfigurasi nilai *hyperparameter* dalam upaya peningkatan akurasi. *Hyperparameter* yang digunakan mencakup pengaturan epoch, *batch*, *confidence*, IoU dan *learning rate*. Setelah beberapa kali percobaan dengan konfigurasi nilai *hyperparameter* yang berbeda, diperoleh hasil terbaik dari model dengan konfigurasi epoch 35, *batch* 16 *confidence* dan IoU 0,7 serta nilai *learning rate* sebesar 0,001. Evaluasi model

dilakukan dengan metode *Confusion Matrix* untuk mengukur akurasi, presisi, *recall*, dan F1-score.

Hasil pengujian menunjukkan bahwa model memiliki akurasi sebesar 95%, *recall* sebesar 92%, presisi sebesar 85%, dan F1-Score sebesar 88%. Ini berarti 95% dari prediksi model benar, baik untuk data positif maupun negatif. Namun, akurasi saja tidak cukup untuk menilai kinerja model secara keseluruhan, sehingga dihitung juga presisi, *recall*, dan F1-Score. *Recall* sebesar 92% menunjukkan model berhasil mendeteksi 92% dari total *instance* positif pada dataset. Presisi sebesar 85% berarti 85% dari prediksi positif benar-benar positif. Nilai F1-Score sebesar 88% memberikan gambaran keseimbangan antara presisi dan *recall*.

Beberapa faktor mempengaruhi kinerja model, termasuk kualitas dan kuantitas data, nilai hyperparameter, serta teknik augmentasi yang digunakan. Kualitas data sangat mempengaruhi performa model deteksi objek seperti YOLOv8. Resolusi gambar yang rendah dapat menghilangkan detail penting, sehingga menyulitkan model dalam mendeteksi objek kecil atau objek yang berdekatan. Kuantitas data juga berperan penting. Semakin banyak data yang tersedia, semakin luas pola yang dapat dikenali oleh model, sehingga meningkatkan kemampuan generalisasi. Dataset yang kecil tidak menyediakan cukup contoh untuk belajar dengan baik, sementara dataset yang besar dan bervariasi memungkinkan model memahami karakteristik objek dengan lebih baik, terutama jika dilatih menggunakan hyperparameter yang tepat.

Konfigurasi hyperparameter juga memiliki dampak signifikan terhadap kinerja model. Nilai *epoch* yang terlalu tinggi dapat menyebabkan overfitting, sementara nilai yang terlalu rendah dapat menyebabkan underfitting. Dalam penelitian ini, penggunaan 35 *epoch* dinilai optimal karena menyeimbangkan penurunan *loss* dan peningkatan presisi-*recall*. Penggunaan batch size 16 dengan 35 *epoch* memberikan keseimbangan yang baik antara stabilitas gradien dan kemampuan generalisasi. Dengan *learning rate* sebesar 0.001, model menghasilkan akurasi 0.96, presisi 1.00, dan *recall* 0.83. Teknik augmentasi seperti rotasi dan *mosaic* juga membantu memperkaya dataset yang terbatas, meningkatkan variasi data yang dapat dipelajari oleh model.



Untuk meningkatkan kinerja model, disarankan agar kualitas dan kuantitas dataset ditingkatkan dengan menggunakan peralatan akuisisi yang lebih baik dan pencahayaan yang optimal. Penyesuaian nilai hyperparameter juga diperlukan untuk menghindari overfitting atau underfitting. Selain itu, penggunaan teknik augmentasi yang lebih bervariasi, seperti rotasi dan *mosaic*, dapat membantu model mendeteksi objek dengan lebih efektif.

## ABSTRAK

Identifikasi Penyakit Daun Teh Menggunakan Metode *You Only Look*

*Once* Versi 8 (Yolov8)

Wella Amanda

Pertanian adalah sektor penting di Indonesia, dengan teh sebagai salah satu komoditas unggulan. Sumatera Utara, khususnya PTPN IV di Bah-Butong, adalah daerah penghasil teh utama dengan rata-rata hasil panen 50-55 ton/hari. Tanaman teh rentan terhadap penyakit daun yang mempengaruhi kualitas dan rasa, sehingga identifikasi penyakit daun teh menjadi penting. Penelitian ini menggunakan model YOLOv8 untuk mendeteksi penyakit daun teh, dengan data yang diambil dari PTPN IV dan divalidasi oleh pakar BRIN. Model YOLOv8 asli digunakan dengan perubahan pada *hyperparameter* untuk meningkatkan akurasi. *Hyperparameter* yang diuji mencakup *epoch*, *batch size*, *confidence*, *IoU*, dan *learning rate*. Konfigurasi nilai *hyperparameter* terbaik yang diperoleh adalah *epoch* 35, *batch size* 16, *confidence* dan *IoU* 0,7, dan *learning rate* 0,001. Dengan menggunakan evaluasi model *Confusion Matrix* diperoleh nilai akurasi, presisi, *recall* dan F1-Score berturut-turut adalah 95%, 85%, 92% dan 88%. Hasil ini menunjukkan bahwa model YOLOv8 efektif dalam mendeteksi penyakit daun teh, dengan keseimbangan yang baik antara presisi dan *recall*. Faktor yang mempengaruhi kinerja model meliputi kualitas dan kuantitas data, konfigurasi *hyperparameter*, dan teknik augmentasi.

**Kata Kunci :** YOLO, Daun Teh, *Hyperparameter*, *Confusion Matrix*, Deteksi

## ABSTRACT

*Identification Of Tea Leaf Disease Using You Only See Once Version*

*8 (Yolov8) Methods*

Wella Amanda

*Agriculture is a crucial sector in Indonesia, with tea being one of its prominent commodities. North Sumatra, specifically PTPN IV in Bah-Butong, is a major tea-producing region with an average yield of 50-55 tons per day. Tea plants are susceptible to leaf diseases that affect quality and flavor, making the identification of tea leaf diseases essential. This study employs the YOLOv8 model to detect tea leaf diseases, using data collected from PTPN IV and validated by experts from BRIN. The original YOLOv8 model was used with adjustments to the hyperparameters to improve accuracy. The hyperparameters tested include epochs, batch size, confidence, IoU, and learning rate. The optimal hyperparameter configuration obtained was 35 epochs, batch size of 16, confidence and IoU of 0.7, and a learning rate of 0.001. Using the Confusion Matrix model evaluation, the accuracy, precision, recall, and F1-Score were found to be 95%, 85%, 92%, and 88%, respectively. These results indicate that the YOLOv8 model is effective in detecting tea leaf diseases, with a good balance between precision and recall. Factors influencing model performance include the quality and quantity of data, hyperparameter configuration, and augmentation techniques.*

***Keywords: YOLO, Tea Leaf, Hyperparameters, Confusion Matrix, Detection***

## DAFTAR ISI

LEMBAR PENGESAHAN .....	ii
HALAMAN PERNYATAAN ORISINALITAS .....	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS .....	iv
KATA PENGANTAR .....	v
RINGKASAN .....	vii
ABSTRAK .....	x
ABSTRACT .....	xi
DAFTAR ISI .....	xii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvi
DAFTAR RUMUS .....	xvii
DAFTAR KODE.....	xviii
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah .....	3
1.3. Tujuan Penelitian.....	4
1.4. Batasan Masalah .....	4
1.5. Manfaat Penelitian.....	4
1.6. Sistematika Penulisan.....	5
1.6.1. Bab I Pendahuluan.....	5
1.6.2. Bab II Tinjauan Pustaka .....	5
1.6.3. Bab III Metode Penelitian .....	5
1.6.4. Bab IV Hasil dan Pembahasan.....	5
1.6.5. Bab V Penutup .....	5

BAB II TINJAUAN PUSTAKA.....	6
2.1    Tinjauan Pustaka.....	6
2.2    Dasar Teori.....	10
2.2.1    Pertanian .....	10
2.2.2    Perkebunan Daun Teh .....	11
2.2.3    Penyakit Daun Teh.....	11
2.2.4    Perseroan Terbatas Perkebunan Nusantara IV .....	13
2.2.5 <i>Deep Learning</i> .....	14
2.2.6    Pengolahan Citra Digital .....	14
2.2.7    Deteksi Objek.....	14
2.2.8 <i>You Only Look Once</i> .....	15
2.2.9 <i>Hyperparameter</i> .....	27
2.2.10    Roboflow .....	29
2.2.11    Dataset .....	29
2.2.12 <i>Confusion Matrix</i> .....	30
BAB III METODE PENELITIAN .....	33
3.1    Alur Penelitian .....	33
3.2    Penjabaran Alur Penelitian .....	34
3.2.1    Identifikasi Masalah.....	34
3.2.2    Studi Literatur .....	34
3.2.3    Pengumpulan Data.....	34
3.2.4 <i>Pre-processing</i> dan Augmentasi Data .....	35
3.2.5    Perancangan Model .....	39
3.2.6    Evaluasi .....	50
3.2.7    Pembahasan.....	53
3.3    Alat dan Bahan .....	53

3.3.1	Alat.....	53
3.3.2	Bahan .....	53
BAB IV HASIL DAN PEMBAHASAN .....		54
4.1	Hasil.....	54
4.1.1	Pengumpulan Data .....	54
4.1.2	<i>Pre-processing</i> dan Augmentasi Data.....	55
4.1.3	Perancangan Model .....	56
4.1.4.	Evaluasi .....	64
4.2	Pembahasan .....	66
BAB V KESIMPULAN DAN SARAN .....		70
5.1	Kesimpulan .....	70
5.2	Saran.....	71
DAFTAR PUSTAKA .....		72
LAMPIRAN .....		76

## DAFTAR GAMBAR

Gambar 2. 1. Brown Blight pada Daun Teh .....	12
Gambar 2. 2. Brown Blight pada Daun Teh .....	12
Gambar 2. 3. Gray Blight pada Daun Teh .....	13
Gambar 2.4. Gray Blight pada Daun Teh .....	13
Gambar 2. 5. Arsitektur Model YOLOv8.....	17
Gambar 2. 6. Gambar Blok Modul C2f .....	20
Gambar 2. 7. Modul SPPF .....	21
Gambar 2. 8. <i>Detection Path</i> .....	22
Gambar 2. 9. Logo Roboflow.....	29
Gambar 3. 1 Alur Penelitian .....	33
Gambar 3. 2 Alur Perancangan Model .....	39
Gambar 3. 3 Ilustrasi Pemrosesan Gambar pada Yolov8.....	40
Gambar 3. 1 Alur Penelitian .....	33
Gambar 3. 2 Alur Perancangan Model .....	39
Gambar 3. 3 Ilustrasi Pemrosesan Gambar pada Yolov8.....	40
Gambar 3. 4 Hasil Pelatihan 10 Epoch .....	47
Gambar 3. 5 Hasil Pelatihan 20 Epoch .....	47
Gambar 3. 6 Hasil Pelatihan 35 Epoch.....	48
Gambar 4. 1 Hasil dari <i>Train</i> 1 .....	59
Gambar 4. 2 Hasil dari <i>Train</i> 2 .....	60
Gambar 4. 3 Hasil dari <i>Train</i> 3 .....	61
Gambar 4. 4 Hasil dari <i>Train</i> 4 .....	62
Gambar 4. 5 Hasil <i>Confusion Matrix</i> .....	64
Gambar 4. 6 Kesalahan Hasil Deteksi (FN) .....	65

## DAFTAR TABEL

Tabel 2. 1 Perbandingan Hasil <i>Literatur Review</i> .....	6
Tabel 2. 2 Tabel Ukuran Filter .....	19
Tabel 2. 3 Tabel Jenis <i>Hyperparameter</i> .....	28
Tabel 3. 1. Sebelum dan Sesudah <i>Pre-processing</i> .....	35
Tabel 3. 2 Hasil Anotasi.....	36
Tabel 3. 3 Hasil Labeling.....	37
Tabel 3. 4 Hasil Rotasi .....	38
Tabel 3. 5 Hasil Rotasi .....	38
Tabel 3. 6 Kombinasi Nilai <i>Hyperparameter</i> .....	49
Tabel 3. 7 Contoh Hasil Pengujian.....	50
Tabel 3. 8 Contoh Akumulasi Hasil Klasifikasi .....	52
Tabel 3. 9 Contoh Hasil Perhitungan.....	53
Tabel 4. 1. Komposisi Dataset.....	54
Tabel 4. 2 Hasil Data Akuisisi Lapangan .....	54
Tabel 4. 3 Hasil <i>Pre-processing</i> dan Augmentasi Data .....	55
Tabel 4. 4 Tabel Hasil Percobaan.....	58
Tabel 4. 5 Tabel Hasil Pengujian .....	66



## DAFTAR RUMUS

Rumus 2. 1 Rumus <i>Output</i> .....	19
Rumus 2. 2 Rumus <i>xmin</i> .....	23
Rumus 2. 3 Rumus <i>ymin</i> .....	23
Rumus 2. 4 Rumus <i>xmax</i> .....	24
Rumus 2. 5 Rumus <i>ymax</i> .....	24
Rumus 2. 6 Rumus <i>Intersection Width</i> .....	24
Rumus 2. 7 Rumus <i>Intersection Height</i> .....	25
Rumus 2. 8 Rumus <i>Intersection</i> .....	25
Rumus 2. 9 Rumus Area Prediksi .....	26
Rumus 2. 10 Rumus <i>Ground Truth</i> .....	26
Rumus 2. 11 Rumus <i>Union Area</i> .....	26
Rumus 2. 12 Rumus IoU .....	27
Rumus 2. 13 Rumus Akurasi.....	31
Rumus 2. 14 Rumus Presisi.....	31
Rumus 2. 15 Rumus <i>Recall</i> .....	31
Rumus 2. 16 Rumus F1-Score.....	32

**DAFTAR KODE**

Kode 4. 1 Baris Kode Instalasi Package Ultralytics .....	57
Kode 4. 2 Baris Kode Instalasi Model .....	57
Kode 4. 3 Baris Kode Instalasi Roboflow dan Akses Dataset .....	57
Kode 4. 4 Baris Kode Pelatihan dan Validasi Model .....	58
Kode 4. 7 Baris Kode Pengujian Model .....	63

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Pertanian merupakan salah satu sektor yang memanfaatkan sumber daya alam untuk memenuhi kebutuhan seperti pangan, bahan baku industri, dan kebutuhan lainnya. Indonesia adalah negara tropis dengan hasil produksi pertanian yang beragam, termasuk padi, kedelai, jagung, kacang tanah, ketela pohon, ubi jalar, kopi, kelapa, cengkeh, tebu, karet, dan teh[1]. Salah satu tanaman subtropis yang ada di Indonesia adalah Teh (*Camellia Sinensis*), yang berasal dari dataran tinggi Cina. Tanaman ini tumbuh subur di beberapa wilayah Indonesia dan menjadi salah satu komoditas unggulan dengan total produksi nasional pada 2022 mencapai 136.800 ton [2]. Selain produk dijual dan diekspor, komoditas teh juga digunakan sebagai minuman herbal sehari-hari masyarakat Indonesia [3], mencakup teh putih, teh hitam, teh hijau, teh kuning dan teh oolong [4].

Sumatera Utara merupakan salah satu daerah penghasil daun teh yang dikelola oleh Perseroan Terbatas Perkebunan Nusantara (PTPN) IV, dengan 30 unit usaha yang tersebar di 9 kabupaten, termasuk Unit Bah-Butong di Kabupaten Simalungun[5]. Perkebunan teh yang dikelola oleh Unit Bah-Butong memiliki luas 1.018,28 hektar, dengan rata-rata hasil panen mencapai 50 hingga 55 ton per hari. Namun, dalam proses pemeliharannya, tanaman teh rentan terhadap serangan penyakit, terutama pada bagian daun, seperti *Anthraxnose*, *Alga Leaf*, *Gray Light*, *Bird Eye Spot*, *Brown Light*, *Red Leaf Spot*, dan *White Spot*[6]. Penyakit-penyakit tersebut tidak hanya dapat menurunkan kualitas daun teh, tetapi juga mempengaruhi rasa dan aroma teh yang dihasilkan, sehingga perlu ditangani sejak dini.

Dalam upaya menjaga kualitas daun teh sebelum di produksi, PTPN IV memberikan pestisida tanaman sejak masih di perkebunan. Pestisida adalah zat kimia yang digunakan oleh petani untuk membasmi hama hewan dan tumbuhan di sekitar tanaman, serta mencegah hama pada tanaman teh, sehingga tidak menurunkan kualitas daun yang dihasilkan[7]. Di perkebunan teh PTPN IV Unit Bah-Butong, pemberian pestisida dilakukan secara serentak 8 hingga 10 hari setelah panen dengan teknik penyemprotan (*spraying*) ke seluruh permukaan tanaman teh. Jika masih terdapat daun teh yang terkena penyakit setelah penyemprotan,

pemberian pestisida tambahan akan dilakukan sesuai dengan jenis penyakit yang menyerang tanaman. Menurut Bapak Purwanto, pihak perkebunan masih melakukan identifikasi penyakit secara manual dengan memeriksa kondisi daun teh yang terkena penyakit secara langsung. Hal ini menimbulkan kekhawatiran akan kemungkinan terjadinya kesalahan dalam identifikasi penyakit oleh pihak perkebunan. Untuk mengurangi risiko kesalahan ini, peneliti menawarkan solusi dengan menggunakan teknologi modern untuk membantu identifikasi jenis penyakit pada daun teh.

Salah satu teknologi yang sering dipakai dalam identifikasi dan deteksi objek adalah *You Only Look Once* (YOLO). YOLO merupakan salah satu model algoritma *real time* tercepat yang dapat memproses seluruh gambar sekaligus dan memberikan lokasi serta kelas dari objek yang ada pada gambar tersebut[8]. Hingga saat penelitian ini dilakukan, algoritma YOLO telah berkembang hingga versi 10 (YOLOv10). Namun, pada penelitian ini, peneliti tetap menggunakan versi YOLO yang 8 dengan mempertimbangkan hasil kerja model pada kasus *real-time*. Pemilihan versi YOLO dalam penelitian ini didasarkan pada jurnal-jurnal penelitian lain yang membandingkan kinerja YOLOv8, YOLOv9, dan YOLOv10. Penelitian oleh Ranjan Sapkota dkk. dalam jurnal *Comprehensive Performance Evaluation of YOLOv10, YOLOv9, and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments*[9] menunjukkan bahwa kecepatan kinerja YOLOv8 pada tahap preprocessing dan inference adalah yang tertinggi dibandingkan dua versi lainnya, dengan hasil masing-masing 0,9 dan 4,1 ms. Penelitian lain yang membandingkan kinerja YOLOv8 dan YOLOv9 telah dilakukan oleh Mochamad Althaf Pramasetya dkk[10], dengan objek deteksi pada permainan sepak bola. Hasil penelitian menunjukkan bahwa tingkat akurasi keseluruhan yang diperoleh YOLOv8 lebih tinggi dibandingkan model lainnya, yaitu mencapai 87,4%. Penelitian yang dilakukan oleh Athulya Sundaresan Geetha dkk[11] dengan judul *Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection: Performance Metrics and Model Efficacy* menunjukkan perbandingan kinerja antara YOLOv8 dan YOLOv10. Hasil penelitian tersebut menunjukkan bahwa YOLOv8 unggul dalam deteksi kendaraan, dengan tingkat akurasi 82%, yang 3% lebih tinggi dibandingkan YOLOv10.

Berdasarkan literatur di atas, metode yang digunakan dalam penelitian ini adalah YOLOv8. YOLOv8 memanfaatkan penambahan modul *Convolution to Fusion* (C2f), yang berfungsi untuk melakukan operasi konvolusi dan menggabungkan hasil ekstraksi fitur menjadi representasi yang lebih lengkap. Oleh karena itu, peneliti menggunakan YOLOv8 sebagai dasar pengembangan model untuk klasifikasi penyakit pada daun teh. Dataset untuk model ini akan diambil langsung dari perkebunan menggunakan kamera ponsel, kemudian divalidasi oleh ahli penyakit tanaman daun teh. Dataset tersebut akan diproses melalui tahapan *pre-processing* guna meningkatkan variasi dataset yang terbatas. Proses *pre-processing* ini akan dilakukan secara *online* menggunakan platform Roboflow. Berdasarkan penelitian Ardha Prakash D.[12], yang membandingkan beberapa platform untuk pelabelan data, disimpulkan bahwa Roboflow memiliki keunggulan, termasuk akses gratis dengan fitur augmentasi terbatas, antarmuka pengguna yang ramah, serta kemudahan penggunaan dalam mempersiapkan model. Keunggulan Roboflow juga dijelaskan oleh Qinjie Lin dkk[13], yang menggunakan platform ini dalam pengembangan model, mulai dari *pre-processing* data hingga pengujian. Penelitian tersebut menunjukkan bahwa alur kerja (*workflow*) dari Roboflow dapat diterapkan dalam pembuatan robot pintar dan pengembangan kecerdasan buatan. Platform ini membantu pengembang mengatur modul dan komponen yang diperlukan secara berurutan dan paralel. Oleh karena itu, platform Roboflow dipilih untuk *pre-processing* data selama pengembangan model ini.

Model dievaluasi menggunakan metode *Confusion Matrix*, yang mencakup metrik akurasi, presisi, *recall*, dan F1-score. Hasil dari perhitungan ini akan menjadi acuan untuk penilaian dan analisis model, serta digunakan sebagai dasar perbaikan pada penelitian serupa di masa mendatang.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dibuat, ada beberapa rumusan masalah dalam penelitian ini, di antaranya:

1. Apakah model YOLOv8 dapat digunakan untuk membangun model identifikasi penyakit pada daun teh?
2. Bagaimana hasil evaluasi model YOLO8 yang dibangun dalam identifikasi penyakit daun teh?

### 1.3. Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan utama yang ingin dicapai antara lain sebagai berikut:

1. Membangun model klasifikasi jenis penyakit daun teh menggunakan model YOLOv8.
2. Melakukan evaluasi untuk mengetahui hasil performa dari model yang dibangun.

### 1.4. Batasan Masalah

Batasan masalah dalam penelitian ini dibuat untuk menghindari penelitian keluar dari fokus dan tujuan utamanya. Adapun batasan masalah pada penelitian ini yaitu sebagai berikut:

1. Dataset yang digunakan diperoleh hanya dari Perseroan Terbatas Perkebunan Teh Nusantara IV Unit Bah-Butong yang berlokasi di Kabupaten Simalungun, Sumatera Utara.
2. Citra daun teh yang digunakan ber-ekstensi \*.jpg.
3. Jenis penyakit daun teh yang digunakan sebanyak 2 jenis penyakit yang paling sering muncul pada saat data citra daun teh diambil diambil yaitu *Brown Blight* dan *Gray Blight*.
4. Kelas yang digunakan disesuaikan dengan jenis penyakit daun teh yaitu berjumlah dua jenis penyakit.

### 1.5. Manfaat Penelitian

Beberapa manfaat yang diperoleh setelah melakukan penelitian adalah sebagai berikut:

1. Mengetahui jenis penyakit daun teh menggunakan model YOLOv8.
2. Mengetahui performa model YOLOv8 dalam identifikasi jenis penyakit pada daun teh yaitu *Brown Blight* dan *Gray Blight*.

## **1.6. Sistematika Penulisan**

### **1.6.1. Bab I Pendahuluan**

Bab I ini menjabarkan membahas tentang latar belakang, rumusan masalah-masalah, tujuan masalah, batasan masalah, manfaat dari penelitian yang dilakukan serta sistematika penulisan.

### **1.6.2. Bab II Tinjauan Pustaka**

Pada Bab II berisi hasil tinjauan pustaka dari beberapa penelitian terdahulu yang berkaitan dengan penelitian yang akan dilakukan serta landasan teori yang menunjang keberlangsungan penelitian yang sedang dilakukan

### **1.6.3. Bab III Metode Penelitian**

Pada Bab III terdapat penjelasan mengenai metode, perhitungan, rancangan pengujian, alat dan bahan yang digunakan, dalam penelitian.

### **1.6.4. Bab IV Hasil dan Pembahasan**

Bab IV ini berisi penjelasan terkait hasil implementasi dari seluruh tahapan yang terdapat pada alur penelitian dan pengujian terhadap pemodelan yang akan dibuat.

### **1.6.5. Bab V Penutup**

Bab V akan memberikan simpulan, yang mencakup hasil akhir evaluasi, serta saran dan rekomendasi untuk penelitian lanjutan dengan judul yang berkaitan.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka

Pada tahapan ini dilakukan tinjauan pustaka dari berbagai sumber termasuk jurnal-jurnal, buku serta internet sebagai referensi. Tinjauan pustaka ini memberikan gambaran serta informasi yang diperlukan selama melakukan penelitian. Penelitian-penelitian terkait identifikasi penyakit pada daun tanaman yang dijadikan sebagai referensi oleh penulis dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Perbandingan Hasil *Literatur Review*

NO	Peneliti [Tahun]	Judul	Metode	Hasil	Perbedaan
1	Md. Janibul Alam Soeb dkk [2023] [14]	<i>Tea Leaf Disease Detection and Identification based on YOLOv7 (YOLO-T)</i>	<i>(Improved YOLOv7)</i> atau YOLO-T	Model berhasil mencapai tingkat akurasi 97.3% dengan masing-masing nilai presisi, <i>recall</i> , mAP dan F1-Score adalah 96.7%, 96.4%, 98/2% dan 0.965.	Model yang dibangun menggunakan metode YOLO versi 7 sementara pada penelitian ini akan menggunakan YOLO versi 8
2	Mochamad Althaf Pramasetya Perkasa dkk [2024] [10]	<i>Visual Entity Object Detection System In Soccer Matches Based On Various YOLO architechture</i>	<i>You Only Look Once</i> (YOLOv7, YOLOv8, YOLOv9)	YOLOv8 berhasil melakukan deteksi objek lebih cepat dibandingkan YOLOv7 dan YOLOv9 dengan hasil 87,4%.	Penelitian Mochamad Althaf Pramasetya dkk menggunakan model YOLOv7, YOLOv8, dan YOLOv9, sedangkan peneliti lainnya hanya menggunakan YOLOv8.



3	Ranjan Saptoka dkk [2024] [9]	<i>Comprehensive Performance Evaluation of YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments</i>	<i>You Only Look Once</i> (YOLOv7, YOLOv8, YOLOv9)	YOLOv8 memiliki kecepatan tertinggi dalam <i>pre-processing</i> dan <i>inference</i> (0,9 ms dan 4,1 ms), sementara YOLOv10 unggul dalam <i>post-processing</i> .	Penelitian Ranjan Saptoka dkk menggunakan YOLOv8, YOLOv9, dan YOLOv10, sedangkan peneliti lainnya hanya menggunakan YOLOv8.
4	Athulya Sundaresan Geetha [2024] [11]	<i>Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection: Performance Metrics and Model Efficacy</i>	<i>You Only Look Once</i> (YOLOv8, YOLOv10)	Pada penelitian ini YOLOv8 berhasil mendeteksi objek mobil lebih cepat daripada YOLOv10.	Penelitian oleh Athulya menggunakan model YOLOv10, sedangkan penelitian ini menggunakan model YOLOv8.
5	Ardra Prakash D [2024] [12]	<i>A Comparative Analysis of Object Identification Labelling Platforms: Basketball Perspective</i>	Roboflow, Makesense.ai, SentiSight.ai, Labelbox and SuperAnnotate	Dari hasil penelitian diperoleh bahwa Roboflow merupakan platform yang terbaik diantara dataset <i>tools</i> yang lain.	Penelitian ini tidak membandingkan platform pengolahan dataset, hanya menggunakan platform Roboflow dalam pengembangan model.
6	Qinjie Lin [2021] [13]	<i>RoboFlow: a Data-centric Workflow Management System for Developing AI-enhanced Robots</i>	Platform Roboflow	Penelitian ini berhasil menunjukkan kinerja Roboflow yang baik dan bisa digunakan untuk pengembangan kecerdasan buatan.	Pada penelitian ini tidak dibahas Roboflow secara mendalam namun hanya digunakan untuk <i>pre-processing</i> dan

					augmentasi data.
--	--	--	--	--	------------------

Berdasarkan tabel di atas, dapat dilihat bahwa penelitian mengenai klasifikasi jenis penyakit pada daun sudah pernah dilakukan. Penelitian yang dilakukan oleh Md. Janibul Alam Soeb[14] menggunakan data citra daun teh yang diambil di beberapa wilayah di negara Bangladesh berjumlah 4000 data. Pada penelitian ini model yang dibangun berbasis pada model YOLOv7 dimana model ini masih menggunakan *backbone* E-ELAN. Pengembangan metode YOLOv7 pada penelitian ini adalah dengan menambahkan *planned re-parameterized convolution* yang merupakan improvisasi dan variasi arsitektur jaringan konvolusi pada suatu model. Konsep ini dibuat untuk mengurangi jumlah parameter yang perlu dipelajari dalam proses pelatihan model dan berfungsi untuk meningkatkan hasil akhir namun memperlambat proses pelatihan model. Selain itu, langkah-langkah normalisasi termasuk penggunaan *batch normalization* yang dikombinasikan dengan lapisan konvolusi juga digunakan pada penelitian ini. Hal ini untuk menjaga stabilitas distribusi data dalam setiap *batch*. Hasil dari penelitian ini mendapatkan tingkat akurasi sebesar 97,3% dan berhasil unggul diantara metode-metode klasifikasi, identifikasi dan deteksi lain seperti *Convolutional Neural Network* (CNN), *Deep Convolutional Neural Network* (CNN), DNN, AX-Retina Net, *improved DCNN*, YOLOv5 dan *Multi-Objective Image Segmentation*.

Dalam penelitian dengan judul *Visual Entity Object Detection System In Soccer Matches Based On Various Yolo Architecture* [10], dilakukan perbandingan kinerja antara model YOLOv7, YOLOv8, dan YOLOv9 dalam mengidentifikasi objek pada pertandingan sepak bola. Penyesuaian parameter berdasarkan kapasitas penyimpanan GPU juga dievaluasi. Hasilnya menunjukkan bahwa YOLOv8 memiliki kinerja lebih baik dengan nilai presisi, *recall*, dan F1-score yang lebih tinggi, terutama pada kelas "Bola", serta akurasi keseluruhan (mAP@0.5) sebesar 87,4%. YOLOv9 juga menunjukkan kinerja yang mirip dengan YOLOv8, namun nilai mAP@0.5 yang lebih tinggi pada YOLOv8 menunjukkan keunggulannya dalam mendeteksi objek dengan tingkat kepercayaan yang bervariasi. Kedua model tersebut menunjukkan peningkatan signifikan dibandingkan YOLOv7 dalam kinerja deteksi objek secara keseluruhan. Oleh karena itu, berdasarkan hasil ini,

YOLOv8 dapat dianggap sebagai model dengan kinerja terbaik dalam mendeteksi objek pada dataset yang digunakan. Penelitian ini tidak hanya memberikan wawasan tentang kinerja dan karakteristik model YOLOv7, YOLOv8, dan YOLOv9 dalam konteks deteksi objek pada pertandingan sepak bola, tetapi juga menghasilkan dataset yang siap untuk analisis tambahan atau pelatihan model pembelajaran mendalam.

Penelitian yang dilakukan oleh Ranjan Saptoka dkk[9] mengevaluasi kinerja YOLOv8, YOLOv9, dan YOLOv10 dalam mendeteksi buah kecil di kebun apel dengan mengukur presisi, *recall*, mAP@50, dan kecepatan komputasi. YOLOv9, terutama konfigurasi YOLOv9 Gelan-e, menunjukkan kinerja terbaik dengan mAP@50 tertinggi (0.935) dan akurasi paling tinggi dalam menghitung buah kecil (RMSE 3.11 dan MAE 4.58). Meskipun YOLOv10x unggul dalam presisi (0.908) dan *recall*, YOLOv9 tetap unggul berkat akurasi yang setara dengan kecepatan inferensi yang lebih tinggi, menjadikannya lebih efisien untuk otomatisasi pertanian di kebun yang kompleks.

Deteksi kendaraan yang akurat sangat penting untuk pengembangan sistem transportasi cerdas seperti mengemudi otonom dan pemantauan lalu lintas. Makalah ini membandingkan kinerja dua model *deep learning*, YOLOv8 dan YOLOv10[11], dalam mendeteksi kendaraan seperti sepeda, bus, mobil, motor, dan truk. Berdasarkan berbagai metrik kinerja, hasil menunjukkan YOLOv10 umumnya lebih baik, terutama dalam mendeteksi kendaraan kecil dan kompleks seperti sepeda dan truk, berkat peningkatan arsitektur. Namun, YOLOv8 sedikit unggul dalam deteksi mobil. Kedua model menunjukkan kinerja yang sebanding dalam mendeteksi bus dan motor. Penelitian ini memberikan wawasan tentang kekuatan dan kelemahan masing-masing model serta aplikasinya dalam dunia nyata, membantu mengoptimalkan arsitektur YOLO untuk tugas deteksi kendaraan tertentu.

Pelabelan manual objek sangat melelahkan, memakan waktu, dan rentan terhadap inkonsistensi yang dapat menghambat kemajuan tugas-tugas visi komputer. Inkonsistensi ini dapat menghasilkan model yang tidak akurat dan berkinerja buruk, sehingga penting untuk mengatasi tantangan pelabelan demi pengembangan AI yang etis dan bertanggung jawab. Studi ini mengevaluasi lima

platform pelabelan data populer—Roboflow, Makesense.ai, SentiSight.ai, Labelbox, dan SuperAnnotate—untuk menilai kesesuaiannya dalam menghadapi tantangan ini[12]. Analisis dilakukan berdasarkan fitur, kemudahan penggunaan, harga, dan dukungan untuk anotasi gambar, deteksi objek, serta integrasi dengan YOLOv8. Setelah mempertimbangkan kelebihan dan kekurangan setiap platform, rekomendasi akhir diberikan mengenai platform yang menawarkan keseimbangan terbaik antara fitur, efisiensi, dan efektivitas biaya untuk tugas deteksi basket menggunakan YOLOv8.

Pada penelitian dengan judul *RoboFlow: a Data-centric Workflow Management System for Developing AI-enhanced Robots*[13] dibahas secara mendalam tentang platform Roboflow. RoboFlow memodelkan proses pengembangan robot menjadi 4 modul utama (pemrosesan data, pengembangan algoritma, pengujian kembali, dan adaptasi aplikasi) yang berinteraksi dengan mesin data terpusat. Semua modul ini dikontainerisasi dan diorkestrasikan di bawah kerangka antarmuka terpadu, meningkatkan pemeliharaan dan penggunaan kembali modul, serta memungkinkan pengembangan secara paralel. Kami mendemonstrasikan efektivitas RoboFlow dengan dua prototipe, "Egomobility" untuk navigasi robot *mobile* dan "Egoplan" untuk pemecahan masalah perencanaan jalur pada lengan robot. Hasilnya menunjukkan bahwa RoboFlow dapat mempercepat siklus pengembangan dan dapat diterapkan pada berbagai aplikasi robotik cerdas.

Berdasarkan studi literatur yang telah dilakukan maka metode yang akan digunakan pada penelitian ini adalah menggunakan model YOLOv8 sebagai *base* model dan platform Roboflow untuk mengolah datasetnya. Model kemudian akan dievaluasi menggunakan metode evaluasi *confusion matrix* untuk melihat kinerja dan hasil performa dari model.

## 2.2 Dasar Teori

### 2.2.1 Pertanian

Pertanian adalah kegiatan memanfaatkan sumber daya hayati untuk memenuhi kebutuhan hidup manusia dengan mengelola lingkungan sekitar. Pertanian dilakukan dengan mengelola sumber daya alam yang tersedia termasuk memelihara

tanaman-tanaman yang dianggap akan mampu memenuhi kebutuhan hidup[15]. Selain sebagai sektor penopang kebutuhan hidup, pertanian juga merupakan dasar pertumbuhan ekonomi, penghasil devisa, penyedia lapangan pekerjaan serta menjadi wadah penyediaan pangan masyarakat[1].

### **2.2.2 Perkebunan Daun Teh**

Perkebunan teh merupakan area perkebunan yang dikelola dengan menanam dan melestarikan tanaman teh sebagai upaya pemenuhan kebutuhan masyarakat terhadap produk berbahan dasar teh (Indonesia *investment*). Budidaya perkebunan teh dapat dikelola secara perorangan dan persero menjadikan Indonesia sebagai salah satu produsen daun teh terbesar di dunia.

### **2.2.3 Penyakit Daun Teh**

Teh merupakan salah satu komoditas unggulan Indonesia yang tersebar luas di Jawa Barat, Sumatera Utara dan beberapa daerah perkebunan teh lain[1]. Namun dalam proses pemeliharannya, tanaman teh dapat terserang penyakit terutama pada bagian daun. Terdapat beberapa penyakit yang paling umum menyerang daun teh seperti *Anthrachnose*, *Alga Leaf*, *Gray Light*, *Bird Eye Spot*, *Brown Light*, *Red Leaf Spot* dan *White Spot*[6]. Namun pada penelitian ini, penulis akan menggunakan dua jenis penyakit, yaitu, *Brown Blight* dan *Grey Blight*.

#### **2.2.3.1 *Brown Blight***

*Brown Blight* adalah salah satu penyakit umum pada daun teh, yang banyak menyebabkan kerusakan produksi dan penurunan kualitas daun teh jika tidak ditangani dengan cepat[16]. Penyakit ini data diidentifikasi dari ciri morfologis yang muncul seperti terdapat titik-titik berwarna gelap dan dikelilingi garis pembatas berwarna merah gelap dan sedikit kuning[17]. Penyakit ini muncul jika daun terkena infeksi jamur *Exobasidium Vexans*[18] seperti pada Gambar 2.1 dan Gambar 2.2.



Gambar 2. 1. *Brown Blight* pada Daun Teh



Gambar 2. 2. *Brown Blight* pada Daun Teh

#### **2.2.3.2 *Gray Blight***

*Gray Blight* merupakan salah satu penyakit yang paling banyak menyerang perkebunan daun teh di dunia. Penyakit ini disebabkan oleh pertumbuhan jamur *Pestilatiopsis Theae* pada daun. Ciri atau gejala umum kemunculan penyakit *Gray Blight* ini ditandai dengan munculnya bintik coklat ditengah-tengah daun, yang kemudian berubah menjadi bintik abu dengan pinggiran berwarna coklat dan semakin lama akan menyebar ke seluruh permukaan daun[19]seperti pada Gambar 2.3 dan Gambar 2.4.



Gambar 2. 3. *Gray Blight* pada Daun Teh



Gambar 2.4. *Gray Blight* pada Daun Teh

#### **2.2.4 Perseroan Terbatas Perkebunan Nusantara IV**

Perseroan Terbatas Perkebunan Nusantara (PTPN) IV adalah perusahaan yang bergerak dalam bidang industri pertanian dengan mengolah komoditas kelapa sawit dan daun teh. Perusahaan ini memiliki 30 unit usaha yang tersebar di 9 kabupaten di Sumatera Utara, yaitu Kabupaten Langkat, Deli Serdang, Serdang Bedagai, Asahan, Simalungun, Labuhan Batu, Padang Lawas, Batu Bara dan Mandailing Natal. Selain melakukan pemeliharaan perkebunan daun teh dan kelapa sawit, PTPN IV juga mengolah komoditas yang dihasilkan menjadi berbagai bahan baku industri dan melakukan pemasaran hasil[5]. Dalam penelitian yang dilakukan, peneliti memilih unit usaha yang berada di Simalungun tepatnya di unit usaha Bah-

Butong sebagai lokasi penelitian untuk pengambilan data (berupa daun teh) yang akan digunakan untuk pengembangan model.

### **2.2.5 Deep Learning**

*Deep learning* merupakan cabang ilmu komputer sains yang mengambil peran dalam bidang pengembangan robot, *Artificial Intelligence* dan teknologi lanjut. *Deep learning* sering diartikan sebagai sebuah pendekatan baru pada pembelajaran mesin (*machine learning*) yang berfokus pada pembelajaran representasi data menggunakan arsitektur-arsitektur jaringan syaraf (*neural network*). Pada prosesnya, *deep learning* memiliki beberapa lapisan yang akan melakukan proses pengolahan data untuk mengambil informasi, kesimpulan, pola atau keputusan sebagai hasil. Namun untuk mendapatkan hasil yang lebih aktual, diperlukan adanya proses pelatihan (*Training*) pada model *deep learning* untuk mengenali data-data yang nantinya akan digunakan pada model. Jika tingkat akurasi yang didapatkan selama pelatihan sudah didapatkan, maka model akan digunakan untuk mengolah data asli (data uji) untuk mengetahui kinerja model yang telah dibuat[20]

### **2.2.6 Pengolahan Citra Digital**

Pengolahan citra digital (*Digital Image Processing*) merupakan cabang ilmu yang mempelajari tentang citra atau gambar secara mendalam untuk mendapatkan informasi yang dapat dipahami manusia. Berdasarkan jenis sinyal penyusunnya, citra terbagi menjadi dua yaitu citra analog dan citra digital. Citra analog merupakan citra yang terbentuk dari sinyal analog seperti mata manusia atau kamera analog. Sedangkan citra digital adalah citra yang terbentuk dari susunan sinyal-sinyal digital yang bersifat diskrit dan bisa diolah menggunakan komputer digital. Secara matematis, citra yang dapat diolah menggunakan komputer digital harus disajikan dalam bentuk numerik dengan nilai diskrit. Pengolahan citra digital dilakukan sebagai upaya penyelesaian masalah serta perluasan pengetahuan mengenai konsep tipografi, citra *bitmap* dan citra *vector* dengan melakukan pemrosesan gambar (memperbaiki, melakukan analisa dan melakukan perubahan pada gambar)[21].

### **2.2.7 Deteksi Objek**

Deteksi objek adalah proses mengidentifikasi dan pengenalan objek spesifik dari data video atau gambar untuk mencari informasi penting dari data yang diolah



dengan menggunakan kemampuan pengenalan objek seperti manusia. Pendeteksian objek ini dapat dilakukan dengan metode tradisional (*Viola-Jones Algorithm*, *Scale-Invariant Transform* (SIFT), *Histogram of Oriented Gradients* (HOG)) hingga metode terbaru yang dikembangkan (CNNs, *Fast R-CNN*, *Faster R-CNN*, *Single Shot Multibox Detector* (SSD), *You Only Look Once* (YOLO), *RetinaNet*). Deteksi objek berperan dalam pengembangan pengenalan wajah, pengembangan teknologi robot dan manufaktur, pengembangan kendaraan otomatis (*self-driving*), identifikasi dalam dunia medis hingga pengembangan *augmented reality* (AR)[22].

### 2.2.8 *You Only Look Once*

*You Only Look Once* (YOLO) merupakan metode deteksi objek *real-time* yang dirilis pertama kali oleh Joseph Redmon pada tahun 2015 sekaligus merupakan YOLO versi 1 (YOLOv1). YOLO termasuk dalam *one-stage object detection*, yaitu model atau metode yang mampu melakukan deteksi secara cepat dalam satu kali proses. Implementasi metode YOLO mencakup tugas pelacakan objek, perhitungan jumlah atau volume suatu objek, identifikasi jenis objek, klasifikasi dan tugas deteksi objek-objek lain[23], [24]. Pada prosesnya, model akan membagi gambar *input* ke dalam grid berukuran  $S \times S$  dan memberi *bounding box* berikut dengan nilai *confidence* yang menunjukkan seberapa yakin model dengan objek yang dideteksi. Arsitektur dasar yang digunakan dalam model YOLO ini adalah *Cross Stage Partial Darknet-53* (CSPDarknet53) yang menerapkan pemisahan peta fitur lapisan dasar menjadi dua dan menggabungkannya kembali melalui hirarki *cross-stage*. Teknik ini membantu memperlancar aliran gradien pada jaringan model dan meningkatkan hasil deteksi objek.

Versi YOLO yang pertama kali dirilis dapat melakukan deteksi objek dan segmentasi dengan menyertakan lokasi objek berupa kotak pembatas (*bounding box*) pada objek yang sedang dideteksi tanpa mengurangi tingkat akurasi hasil. Namun, meskipun merupakan model deteksi objek tercepat pada masanya, YOLOv1 sering melakukan kesalahan dalam memprediksi lokasi objek dan memiliki keterbatasan dalam mendeteksi lebih dari satu objek dengan kelas yang sama pada satu gambar masukan. Sehingga pada 2017 muncul pengembangan baru dari YOLOv1, yaitu YOLOv2.

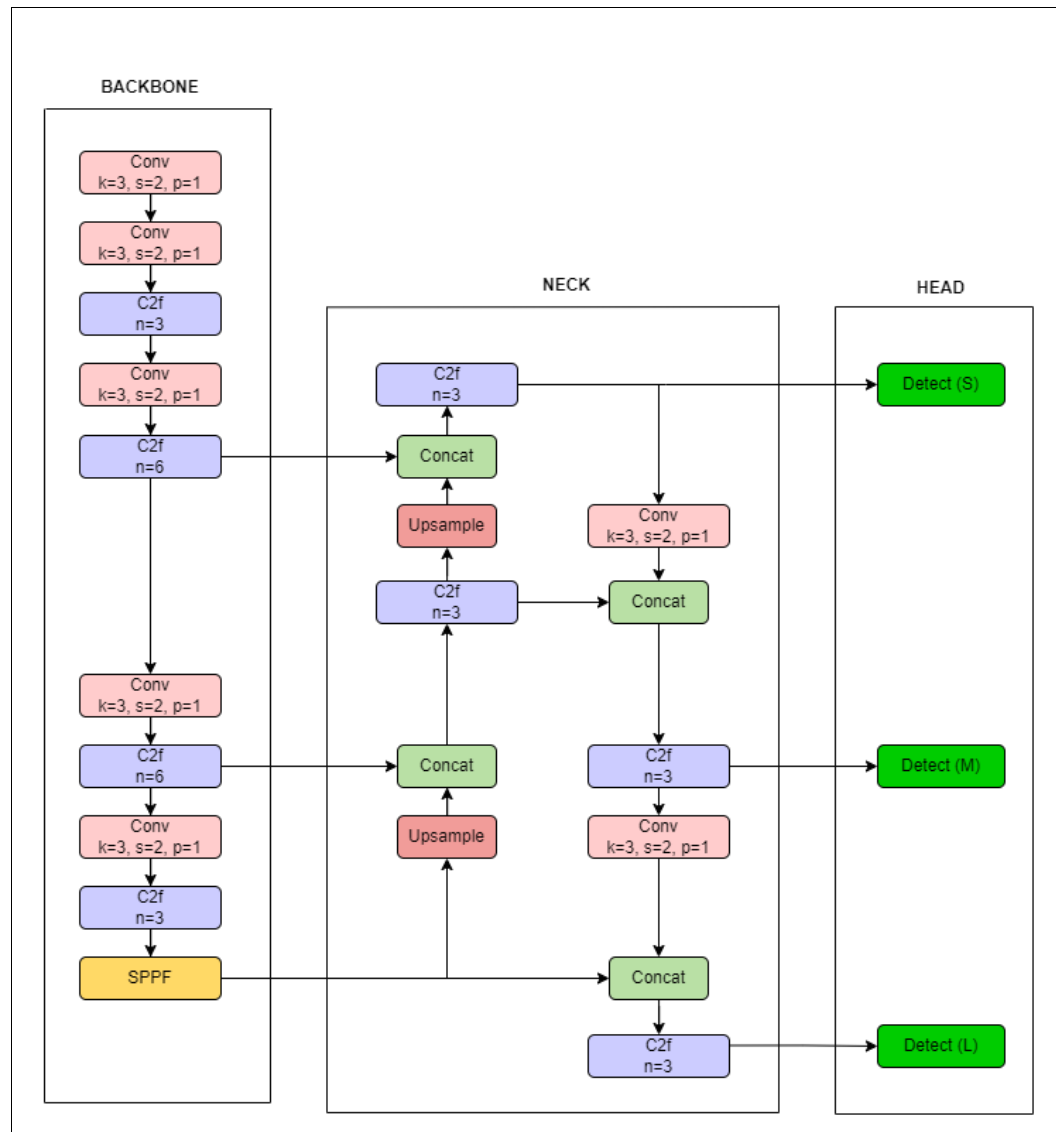
YOLOv2 dikembangkan dengan kecepatan dan ketepatan yang lebih tinggi dan dapat digunakan pada ukuran gambar yang lebih variatif. Arsitektur dari YOLOv2 mencakup sembilan belas *convolutional layer* dan lima *max-pooling layer* dengan penggunaan *Anchor Box* untuk meningkatkan prediksi *bounding box* pada objek. Pengembangan seri selanjutnya dari YOLO adalah YOLOv3 pada 2018 oleh Joseph Redmon dan Ali Farhadi.

Pada YOLOv3, *backbone* pada arsitektur sebelumnya dikembangkan menjadi lima puluh tiga *convolutional layer* dan penggunaan *Spatial Pyramid Pooling* (SPP) untuk meningkatkan nilai akurasi model. Selain itu, pengembangan terbesar pada YOLOv3 adalah *multi-scale predictions* yaitu kemampuan model dalam melakukan deteksi pada lebih dari satu objek dengan ukuran yang berbeda. Hal ini membantu tugas deteksi objek berukuran kecil, yang merupakan kekurangan dari model YOLO seri sebelum-sebelumnya. Pengembangan selanjutnya dari model YOLO yaitu YOLOv4 yang memperkenalkan arsitektur YOLO baru dengan penambahan integrasi *bag-of-freebies* (BoF) dan *bag-of-specials* (BoS). BoF adalah teknik meningkatkan kinerja model dengan melakukan peningkatan termasuk augmentasi dan normalisasi pada dataset. Sedangkan BoS adalah penggunaan komponen-komponen pada model termasuk penambahan *layer-layer* yang dapat mendukung peningkatan kinerja model dalam melakukan tugas yang akan dilakukan.

Pengembangan model YOLOv5 dilakukan oleh sebuah Perusahaan Ultralytics pada tahun 2020, hanya berselang beberapa bulan setelah perilisannya YOLOv4. Model YOLO versi kelima ini menggunakan *backbone* dari EfficientNet atau disebut juga CSPNet untuk membantuk meningkatkan kecepatan deteksi tanpa mengurangi tingkat akurasi. Pengembangan terbaru dari seri YOLO yaitu YOLOv6 menggunakan jenis *backbone* yang berbeda yaitu RepVGG yang menggunakan paralelisme yang lebih tinggi untuk meningkatkan efisiensi dan kinerja deteksi. Sedangkan pada YOLOv7 digunakan jaringan *Efficient Long-range Aggregation Network* (E-ELAN) untuk melakukan optimalisasi gradien dalam model. Terakhir, seri terbaru dari YOLO dirilis pada awal 2023 lalu oleh Perusahaan Ultralytics yang juga merupakan pengembang YOLOv5 dan YOLOv3.

### 2.2.8.1. YOLOv8

YOLOv8 merupakan bentuk pengembangan terbaru dari seri YOLO dan juga merupakan model deteksi objek terbaik yang dikembangkan oleh Perusahaan Ultralytics pada Januari 2023 lalu. Arsitektur YOLOv8 terdiri dari beberapa blok utama sebagai penyusun dan dapat dilihat pada gambar Gambar 2.5.



Gambar 2. 5. Arsitektur Model YOLOv8

Terlihat pada gambar, arsitektur YOLOv8 terdiri dari delapan lapisan *Convolutional*, delapan blok modul C2f, satu blok *Spatial Pyramid Pooling Fast* (SPPF), dua lapisan Upsample dan empat lapisan Concat. YOLOv8 terbagi menjadi tiga bagian yaitu *backbone*, *neck* dan *head*. *Backbone* merupakan bagian *deep learning* arsitektur yang melakukan fungsi ekstraksi fitur dari gambar *input*.

Sementara bagian selanjutnya banyak diartikan sebagai bagian *neck* arsitektur yaitu bagian yang mengolah fitur-fitur hasil ekstraksi dari *backbone* sebelumnya. Dan bagian *head* melakukan fungsi deteksi pada model. Pada bagian *head* ini, terdapat dua *path* atau jalur tahapan yang akan melakukan prediksi *bounding box* dan prediksi kelas dari objek. Model YOLOv8 mampu mendeteksi objek dalam berbagai variasi mulai dari *nano*, *small*, *medium*, *large* dan *extra-large*.

### 1. *Backbone*

Pada awal pemrosesan, gambar tiga *channel* dengan ukuran  $S \times S$  diinput ke dalam model. Gambar awalnya akan diproses melalui *backbone* yang terdiri dari sejumlah lapisan konvolusi, beberapa blok modul C2f dan lapisan *Spatial Pyramid Pooling Fusion* (SPPF). Lapisan konvolusi pada *backbone* atau disebut *convolutional layer* ini memuat *layer* Conv2d, BatchNorm2d dan fungsi aktivasi SILU. Ketiga *layer* ini digunakan sebagai satu kesatuan dalam memproses dan melakukan ekstraksi fitur-fitur dari suatu gambar secara mendalam

Lapisan konvolusi ini memiliki tiga parameter utama yaitu *kernel*, *stride* dan *padding*. *Kernel* merupakan filter utama yang akan mengekstraksi blok piksel pada gambar yang diinput. Filter *kernel* ini mengekstraksi keseluruhan gambar dengan bergeser ke arah kanan dan ke bawah. Hasil dari ekstraksi filter *kernel* ini sering disebut *Feature Map* atau peta fitur. *Stride* merupakan jarak perpindahan filter *kernel* dalam melakukan ekstraksi gambar. Dengan kata lain, *kernel* akan bergeser sejauh *stride*. Sementara *padding* adalah nilai tambahan yang ditambahkan pada blok piksel suatu gambar untuk menghasilkan peta fitur yang lebih akurat dan tanpa mengurangi informasi penting dari gambar. Lapisan konvolusi pada YOLOv8 memiliki ukuran *kernel* tiga, *stride* dua dan *padding* satu. Pada *backbone*, *stride* berukuran dua berperan dalam mengurangi resolusi gambar sehingga *output* gambar akan berubah. Untuk masing masing gambar, dikarenakan setiap lapisan konvolusi pada arsitektur YOLOv8 menggunakan *stride* 2, maka ukuran *output* gambar dari tiap-tiap proses konvolusi adalah setengah dari ukuran *input* sebelum proses konvolusi. Hasil *output* gambar memuat ukuran baru dari gambar serta jumlah *channel* yang baru. Untuk menghitung ukuran keluaran dari gambar yang diproses menggunakan *kernel*, *stride* dan *padding* dapat dihitung menggunakan Rumus 2.1.

$\text{Keluaran} = \frac{\text{Ukuran } Input + 2 \times P - K}{S} + 1$	(Rumus 2. 1 Rumus <i>Output</i> )
---	-----------------------------------

Keterangan:

Keluaran = Ukuran keluaran gambar yang diproses

P = Ukuran *Padding*

K = Ukuran *Kernel*

S = Ukuran *Stride*

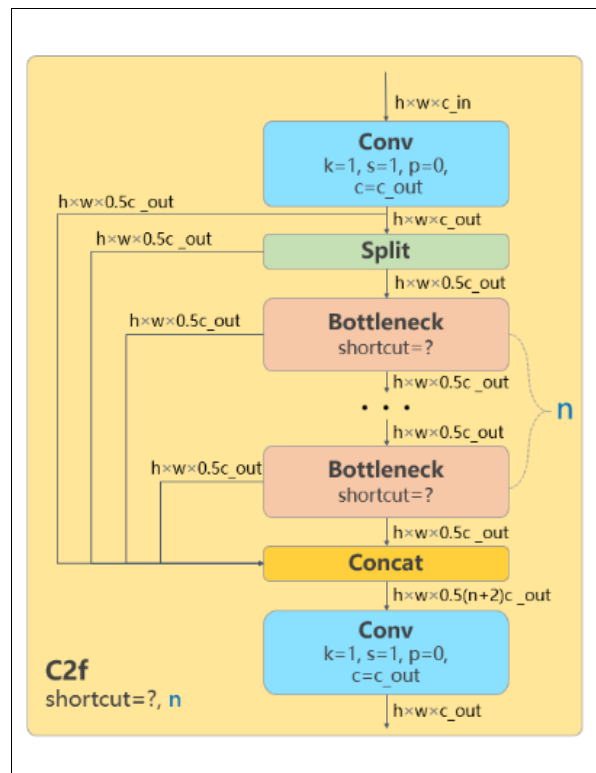
Sementara ukuran saluran sesuai dengan jumlah filter yang digunakan pada lapisan konvolusi tersebut seperti tertera pada Tabel 3.1.

Tabel 2. 2 Tabel Ukuran Filter

Lapisan ( <i>Layer</i> )	<i>Input</i>	<i>Filter</i>	<i>Output</i>
Conv	3	48	48
Conv	48	96	96
C2f	96	96	96
Conv	96	192	192
C2f	192	192	192
Conv	192	384	384
C2f	384	384	384
Conv	384	576	576
C2f	576	576	576
Sppf	576	576	576
Upsample	576	576	576
Concat			960
C2f	960	384	384
Upsample	384	384	384
Concat			576
C2f	576	192	192
Conv	192	192	192
Concat			576
C2f	576	384	384

Conv	384	384	384
Concat			960
C2f	960	576	576

Selanjutnya adalah modul *Convolution to Fusion* (C2f), yang merupakan pengembangan utama pada bagian *backbone* model. Penggunaan modul C2f pada bagian *backbone* model bertujuan untuk mengombinasikan dan meningkatkan representasi fitur untuk membantu pemahaman jaringan terhadap gambar yang dalam hal ini dapat membantu peningkatan nilai akurasi deteksi. Blok C2f digambarkan pada Gambar 2.6.

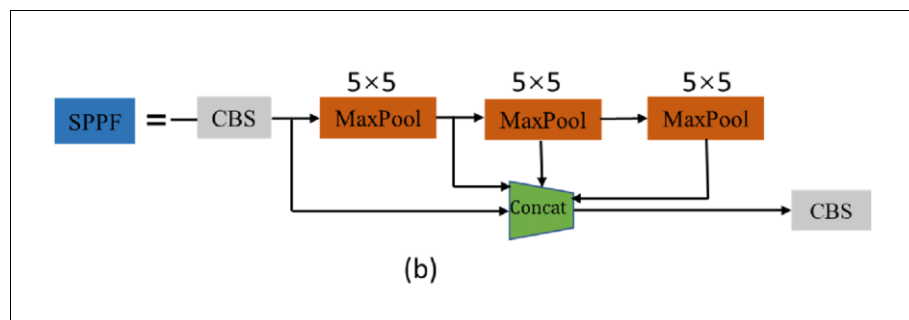


Gambar 2. 6. Gambar Blok Modul C2f

Pada modul C2f, terdapat empat lapisan penyusun utama yaitu lapisan konvolusi, lapisan *split*, lapisan *bottleneck* dan lapisan *concat*[25]. Pada blok C2f ini, gambar akan kembali diekstraksi dan melalui proses konvolusi untuk mengetahui fitur-fitur penting pada gambar. Selanjutnya hasil fitur-fitur akan dibagi (*split*) dalam dua sambungan dimana jalur pertama tersambung ke lapisan *concat* pada bagian dan tersambung juga ke *bottleneck*. Fitur-fitur ini kemudian akan

melalui lapisan *bottleneck* untuk mengurangi dimensi fitur pada tiap gambar. *Output* dari proses pada *bottleneck* ini memiliki ukuran lebih kecil dari gambar *input* awal. Selanjutnya akan dijalankan fungsi pada *concat* untuk menggabungkan fitur-fitur dari berbagai jalur dalam jaringan. Setelah penggabungan fitur dilakukan, terdapat operasi konvolusi terakhir pada blok modul untuk meningkatkan kompleksitas representasi fitur.

Pada bagian *backbone*, dilakukan beberapa kali operasi lapisan konvolusi dan C2f yang *output*nya tersambung ke dua jalur yaitu *concat* pada bagian *neck* dan blok *Spatial Pyramid Pooling Fast* (SPPF) pada *backbone*. Pada akhir bagian *backbone*, terdapat blok SPPF yang berfungsi untuk meningkatkan representasi fitur dan menghasilkan representasi akhir fitur dari berbagai ukuran objek pada gambar tanpa mengubah ukuran spasial dari *input*. Blok SPPF ini akan melakukan operasi *pooling* maksimum 5x5 kemudian menggabungkan semua hasil *pooling* dari skala berbeda untuk membantu meningkatkan kemampuan model dalam melakukan tugas tugas kompleks. Gambar modul SPPF dapat dilihat pada Gambar 2.8.



Gambar 2. 7. Modul SPPF

Sumber : ASFF-YOLOv5: *Multielement Detection Method for Road Traffic in UAV Images Based on Multiscale Feature Fusion*[26]

## 2. Neck

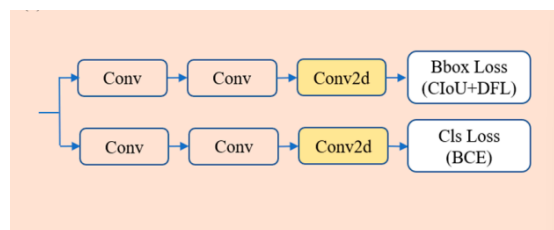
Selanjutnya pada bagian *neck* terdapat lapisan Upsample yang berfungsi untuk meningkatkan dimensi dari peta fitur gambar yang dihasilkan agar sama dengan nilai resolusi gambar yang akan disatukan dengan Concat. Upsample umumnya menggunakan interpolasi *bilinear* yang merupakan interpolasi umum dalam Upsample. Lapisan ini menambahkan nilai piksel berupa baris dan kolom di antara tiap-tiap pasangan piksel yang ada untuk meningkatkan ukuran spasial gambar asli.

Penggabungan peta fitur gambar ini dilakukan menggunakan lapisan Concat. Pada YOLOv8 teknik yang digunakan dalam lapisan upsample adalah *nearest neighbor interpolation*, dimana setiap ukuran spasial pada peta fitur *input* diperbanyak menjadi beberapa piksel peta fitur dengan cara menyalin nilai piksel yang sama. Teknik ini memungkinkan model untuk memperbesar ukuran peta fitur dengan cepat dan tanpa menambahkan informasi baru pada model.

Gambar dengan resolusi yang berbeda digabungkan melalui suatu lapisan yang disebut Concat. Concat bertanggung jawab untuk menyatukan fitur *maps* dari blok-blok yang berbeda dalam jaringan model. Kombinasi fitur dari tingkatan berbeda ini memungkinkan model menangkap informasi dengan lebih bervariasi dan meningkatkan kemampuan model untuk melakukan deteksi objek secara akurat di berbagai skala dan resolusi. Dalam proses operasi konkatenasi, kombinasi fitur dilakukan dengan menyatukan peta fitur dari setiap blok untuk membentuk peta fitur yang lebih besar. Pada implementasinya, lapisan Concat ini akan melakukan operasi untuk penyesuaian ukuran peta fitur dari masing-masing gambar yang akan disatukan seperti melakukan penambahan satu dimensi pada peta fitur.

### 3. *Head*

Bagian *head* merupakan bagian akhir dari metode YOLO yang bertanggung jawab atas pemberian *bounding box* dan kelas dari objek yang dideteksi. Pada YOLOv8, jalur perolehan *bounding box* dan klasifikasi kelas objek dibagi menjadi dua sehingga memungkinkan model untuk melakukan deteksi dengan lebih akurat.



Gambar 2. 8. *Detection Path*

Pada bagian *head* ini, lapisan Conv2d digunakan untuk melakukan operasi konvolusi dua dimensi pada data masukan dan menghasilkan peta fitur baru. Terdapat dua jalur pada *head detection of YOLOv8*, satu *path* untuk menghitung *boundong box loss* dan satu lagi untuk menghitung *class loss*. *Bounding box loss*



menghitung keakuratan model dalam memprediksi lokasi dan ukuran dari kotak pembatas untuk objek pada gambar. Perhitungan ini dilakukan dengan membandingkan nilai *ground truth* dengan hasil prediksi menggunakan CIOU dan DFL. Sementara *class loss* digunakan untuk mengukur kinerja model dalam melakukan klasifikasi objek pada kotak pembatas hasil prediksi dengan menghitung nilai *Binary Cross-Entropy* (BCE). Dalam penerapannya, *bounding box loss* dan *class loss* bukan merupakan nilai yang digunakan untuk menentukan kotak pembatas dan kelas. Tetapi digunakan untuk mengukur kinerja model dalam melakukan tugas identifikasi dengan menghitung nilai *loss* (kesalahan).

Untuk menentukan kotak pembatas, digunakan rumus regresi. Setelah gambar diproses oleh model, model akan menghasilkan nilai prediksi yang meliputi nilai titik koordinat pusat ( $x_{center}, y_{center}$ ), lebar kotak pembatas ( $w$ ), tinggi kotak pembatas ( $h$ ), dan skor kepercayaan (*confidence score*). Nilai-nilai ini selanjutnya akan digunakan untuk menghitung koordinat dari kotak pembatas yang terdiri dari nilai  $x_{min}, y_{min}, x_{max}, y_{max}$  yang menunjukkan nilai sudut kanan dan kiri kotak pembatas. Untuk menghitung nilai  $x_{min}, y_{min}, x_{max}, y_{max}$  digunakan rumus berikut.

$x_{min} = x_{center} - \frac{w}{2}$	Rumus 2. 2 Rumus $x_{min}$
--------------------------------------	----------------------------

Keterangan:

$x_{min}$  = nilai x pada sumbu horizontal yang menunjukkan posisi paling kiri dari kotak pembatas

$x_{center}$  = nilai x yang menunjukkan posisi horizontal dari pusat kotak pembatas

$w$  = lebar dari kotak pembatas

$y_{min} = y_{center} - \frac{h}{2}$	Rumus 2. 3 Rumus $y_{min}$
--------------------------------------	----------------------------

Keterangan:

$y_{min}$  = nilai y pada sumbu vertikal yang menunjukkan posisi paling atas dari kotak pembatas

$y_{center}$  = nilai y yang menunjukkan posisi vertikal dari pusat kotak pembatas

$h$  = tinggi dari kotak pembatas

$x_{max} = x_{center} + \frac{w}{2}$	Rumus 2. 4 Rumus $x_{max}$
--------------------------------------	----------------------------

Keterangan:

$x_{max}$  = nilai x pada sumbu horizontal yang menunjukkan posisi paling kanan dari kotak pembatas

$x_{center}$  = nilai x yang menunjukkan posisi horizontal dari pusat kotak pembatas

$w$  = lebar dari kotak pembatas

$y_{max} = y_{center} + \frac{h}{2}$	Rumus 2. 5 Rumus $y_{max}$
--------------------------------------	----------------------------

Keterangan:

$y_{max}$  = nilai y pada sumbu vertikal yang menunjukkan posisi paling bawah dari kotak pembatas

$y_{center}$  = nilai y yang menunjukkan posisi vertikal dari pusat kotak pembatas

$h$  = tinggi dari kotak pembatas

Setelah mendapatkan nilai-nilai tersebut, dihitung nilai *Intersection Width* untuk menghitung lebar dan tinggi dari area *overlap* antara kotak pembatas hasil prediksi dan *ground truth*. *Intersection width* dihitung menggunakan Rumus 2.6. Sementara *intersection height* dihitung menggunakan Rumus 2.7.

$Intersection\ Width = \min(x_{max}, X_{max}) - \max(x_{min}, X_{min})$	Rumus 2. 6 <i>Intersection Width</i>
---	---

Keterangan:

*Intersection Width* = lebar area *overlap* antara kotak pembatas dan *ground truth*

$x_{max}$  = nilai x pada sumbu horizontal yang menunjukkan posisi paling kanan dari *bounding box*

$X_{max}$  = nilai x pada sumbu horizontal yang menunjukkan posisi paling kanan dari *ground truth*

$x_{min}$  = nilai x pada sumbu horizontal yang menunjukkan posisi paling kiri dari *bounding box*

$X_{min}$  = nilai x pada sumbu horizontal yang menunjukkan posisi paling kiri dari *ground truth*

$Intersection\ Height = \min(y_{max}, Y_{max}) - \max(y_{min}, Y_{min})$	Rumus 2. 7 <i>Intersection Height</i>
--	--

Keterangan:

*Intersection Height* = tinggi area *overlap* antara kotak pembatas dan *ground truth*

$y_{max}$  = nilai y pada sumbu vertikal yang menunjukkan posisi paling bawah dari *bounding box*

$Y_{max}$  = nilai y pada sumbu vertikal yang menunjukkan posisi paling bawah dari *ground truth*

$y_{min}$  = nilai y pada sumbu vertikal yang menunjukkan posisi paling atas dari *bounding box*

$Y_{min}$  = nilai y pada sumbu vertikal yang menunjukkan posisi paling atas dari *ground truth*

Nilai *Intersection Area* diperoleh dari hasil perkalian antara *intersection width* dan *intersection height* seperti yang ditunjukkan pada Rumus 2.8.

$Intersection = Intersection\ Width \times Intersection\ Height$	Rumus 2. 8 <i>Intersection</i>
--	-----------------------------------

Keterangan:

*Intersection* = area *overlap* antara kotak pembatas dan *ground truth*

*Intersection Width* = lebar area *overlap* antara kotak pembatas dan *ground truth*

*Intersection Height* = tinggi area *overlap* antara kotak pembatas dan *ground truth*

Selanjutnya dihitung area prediksi dan *ground truth* menggunakan nilai  $x_{min}, y_{min}, x_{max}, y_{max}$ . Untuk menghitung area prediksi dan area *ground truth* digunakan Rumus 2.9 dan Rumus 2.10.

$Area\ of\ Prediction = (x_{max} - x_{min}) \times (y_{max} - y_{min})$	Rumus 2. 9 Area Prediksi
---	-----------------------------

Keterangan:

$x_{max}$  = nilai x pada sumbu horizontal yang menunjukkan posisi paling kanan dari *bounding box*

$x_{min}$  = nilai x pada sumbu horizontal yang menunjukkan posisi paling kiri dari *bounding box*

$y_{max}$  = nilai y pada sumbu vertikal yang menunjukkan posisi paling bawah dari *bounding box*

$y_{min}$  = nilai y pada sumbu vertikal yang menunjukkan posisi paling atas dari *bounding box*

$Area\ of\ Ground\ Truth = (X_{max} - X_{min}) \times (Y_{max} - Y_{min})$	Rumus 2. 10 Ground Truth
--	-----------------------------

$X_{max}$  = nilai x pada sumbu horizontal yang menunjukkan posisi paling kanan dari *ground truth*

$X_{min}$  = nilai x pada sumbu horizontal yang menunjukkan posisi paling kiri dari *ground truth*

$Y_{max}$  = nilai y pada sumbu vertikal yang menunjukkan posisi paling bawah dari *ground truth*

$Y_{min}$  = nilai y pada sumbu vertikal yang menunjukkan posisi paling atas dari *ground truth*

Selanjutnya dilakukan perhitungan total area yang mencakup prediksi kotak pembatas dengan *ground truth* dengan menjumlahkan area kotak pembatas hasil prediksi dengan *ground truth* dikurangi dengan nilai *intersection*. Rumus menghitung *Union Area* dapat dilihat pada Rumus 2.11.

$Union\ Area = Area\ of\ Prediction\ Box + Area\ of\ Ground\ Truth\ Box - Intersection\ Area$	Rumus 2. 11 Union Area
---	---------------------------

Keterangan:

*Union Area* = total area prediksi kotak pembatas dengan *ground truth*

*Area of Prediction Box* = area kotak pembatas hasil prediksi model

*Area of Ground Truth* = area yang sebenarnya dari kotak pembatas

*Intersection Area* = area tumpang tindih dari kotak pembatas hasil prediksi dengan *ground truth*.

Selanjutnya dilakukan perhitungan nilai *Intersection over Union* (IoU) yang menunjukkan *overlap* antara hasil prediksi dan *ground truth*. Nilai ini biasa digunakan untuk menunjukkan seberapa baik hasil prediksi model jika dibandingkan dengan *ground truth* nya. Rumus untuk menghitung nilai IoU adalah Rumus 2.12.

$\text{IoU} = \frac{\text{Intersection Area}}{\text{Union Area}}$	Rumus 2. 12 IoU
---	--------------------

Keterangan:

*IoU* = *overlap* antara hasil prediksi dan *ground truth*

*Union Area* = total area prediksi kotak pembatas dengan *ground truth*

*Intersection Area* = area tumpang tindih dari kotak pembatas hasil prediksi dengan *ground truth*

Penentuan kelas untuk kasus multi-kelas memerlukan nilai probabilitas untuk masing-masing kelas, di mana jika probabilitas yang dihasilkan model lebih tinggi dari probabilitas suatu kelas, maka hasil prediksi dihitung sebagai kelas yang bersangkutan. Pada kasus multi-kelas untuk kelas yang hanya terdiri dari dua kelas, nilai ambang batas probabilitas yang dibutuhkan cukup satu, artinya jika nilai melampaui ambang batas maka akan termasuk kelas positif dan jika tidak melampaui ambang batas maka termasuk kelas negatif.

### 2.2.9 Hyperparameter

*Hyperparameter* adalah konfigurasi dari variabel-variabel diatur sebelum proses pelatihan untuk memperbaiki performa pada suatu model. Nilai *hyperparameter* ini mempengaruhi hasil kinerja model seperti akurasi, generalisasi

dan matriks evaluasi lainnya[27]. Nilai-nilai ini ditetapkan sebelum proses latihan dan tidak akan berubah selama pelatihan. Tahapan menentukan nilai optimal dari *hyperparameter* disebut juga dengan *hyperparameter tuning* atau proses untuk mencari nilai paling optimal untuk digunakan dalam pelatihan model pembelajaran mesin [28]. Ada banyak jenis *hyperparameter* namun yang digunakan dalam penelitian ini diantaranya *epoch*, *intersection over union (IoU) threshold*, *confidence threshold*, *learning rate* dan *batch size*. Penggunaan *hyperparameter* *epoch* dan *batch* bertujuan untuk membagi data menjadi lebih kecil agar lebih mudah diproses oleh model, juga sebagai upaya antisipasi keterbatasan komputer dalam mengolah data dengan jumlah yang terlalu besar[29]. Jenis-jenis *hyperparameter* yang digunakan pada penelitian ini tercantum dalam Tabel 2.4.

Tabel 2. 3 Tabel Jenis *Hyperparameter*

Jenis <i>Hyperparameter</i>	Deskripsi
Epoch	Jumlah iterasi pelatihan model yang memberikan kesempatan kepada model untuk mempelajari seluruh dataset berkali kali selama pelatihan. Penentuan nilai epoch merupakan hal yang penting karena nilai epoch yang terlalu besar dapat menimbulkan situasi <i>overfitting</i> pada model sedangkan nilai epoch yang terlalu kecil dapat menghambat model mencapai kinerja yang kurang maksimal[29]
<i>Batch size</i>	Jumlah data yang digunakan selama pelatihan model dalam satu kali iterasi. Jumlah <i>batch</i> yang terlalu besar dapat membuat proses data menjadi tidak efektif dan <i>batch</i> yang terlalu kecil dapat memperlambat proses pelatihan[29].
<i>IoU threshold</i>	<i>IoU threshold</i> adalah nilai minimum antara hasil prediksi dan nilai sebenarnya untuk dapat dianggap sebagai prediksi positif. Nilai ini juga digunakan sebagai nilai ambang batas untuk mencegah terjadinya tumpang tindih kotak pembatas pada suatu objek[30].

<i>Confidence threshold</i>	<i>Confidence threshold</i> adalah nilai yang menjadi ambang batas untuk menentukan apakah suatu prediksi dianggap valid atau tidak. Nilai <i>confidence</i> dibawah ambang batas ini dianggap tidak valid[31].
<i>Learning rate</i>	<i>Learning rate</i> , adalah parameter dalam proses pelatihan yang menunjukkan seberapa besar langkah model dalam memperbarui bobotnya. Nilai <i>learning rate</i> ini berada pada <i>range</i> nol (0) sampai (1)[32].

#### 2.2.10 Roboflow

Roboflow merupakan alat pengolahan data gambar termasuk membangun dataset, melakukan pelatihan model hingga *deploy*. Alat ini umumnya digunakan untuk membangun *computer vision models* seperti melakukan anotasi gambar, membuat *bounding box*, memberikan label pada data gambar serta menyusun data data visual yang telah diolah [33]. Platform ini juga mampu melakukan pelatihan model dengan berbagai jenis arsitektur dan melakukan penyempurnaan model. Logo resmi Roboflow seperti pada Gambar 2.10.



Gambar 2. 9. Logo Roboflow

#### 2.2.11 Dataset

Dataset merupakan sekumpulan data berisi informasi yang dikumpulkan melalui penelitian, observasi, pengamatan atau analisa suatu objek spesifik. Data pada umumnya berupa pernyataan, angka-angka, nama atau deskripsi yang disajikan dalam bentuk diagram, *chart* atau tabel. Dataset banyak digunakan di beberapa bidang mencakup data *science*, *machine learning* hingga dalam bidang

penelitian untuk membantu analisa, pengenalan pola data, pemodelan hingga pengambilan keputusan [34].

#### **2.2.11.1 Data Latih**

Data latih merupakan kumpulan data aktual (data sebenarnya) yang akan diolah dan dipelajari oleh model untuk menghasilkan *output* yang ingin diprediksi. Pelatihan model menggunakan data latih ini bertujuan untuk melatih model melakukan pengambilan keputusan yang tepat. Pembagian jumlah data untuk dijadikan data latih pada umumnya lebih dari 60% dari total keseluruhan data pada dataset. Kualitas data selama pelatihan model akan mempengaruhi kinerja model dalam pengambilan keputusan, sehingga data latih yang digunakan harus memiliki kualitas yang baik[34]. Pada penelitian ini, dataset untuk pelatihan adalah sebanyak 70% dari total seluruh dataset yang telah diambil oleh peneliti.

#### **2.2.11.2 Data Validasi**

Data validasi masih termasuk dalam pelatihan model dan digunakan untuk mengevaluasi kinerja model selama pelatihan. Jika akurasi model pada validasi lebih besar daripada akurasi data latih, maka kinerja model yang dibangun dianggap berhasil. Jumlah data validasi dalam suatu model umumnya berkisar 10-15% dari total keseluruhan dataset yang digunakan[34]. Pada penelitian ini, dataset untuk validasi adalah sebanyak 20% dari total seluruh dataset yang telah diambil oleh peneliti.

#### **2.2.11.3 Data Uji**

Data uji merupakan sekumpulan data yang memuat seluruh skenario yang akan dihadapi model ketika diimplementasikan. Sehingga hasil *output* dari pengujian menggunakan data uji ini akan memberikan gambaran hasil kinerja model yang sebenarnya[34]. Pada penelitian ini, dataset untuk pengujian adalah sebanyak 10% dari total seluruh dataset yang telah diambil oleh peneliti.

### **2.2.12 Confusion Matrix**

Evaluasi model dilakukan untuk mengukur kinerja dan kualitas model yang telah dibangun. Pada penelitian kali ini, digunakan metode evaluasi menggunakan *Confusion Matrix*, yang terdiri dari 4 elemen yaitu *True Positive* (TP), *False positive* (FP), *True Negative* (TN) dan *False negative* (FN). TP merupakan jumlah



data positif yang diprediksi benar, FP merupakan jumlah data *negative* yang diprediksi salah, TN jumlah data *negative* yang diprediksi benar dan FN yaitu jumlah data positif yang diprediksi salah. Model evaluasi *Confussion Matrix* ini dapat digunakan untuk menghitung kinerja model mencakup nilai akurasi, presisi, *recall* dan F1-Score.

Akurasi merupakan parameter untuk mengukur keakuratan model dalam mengolah data. Adapun rumus untuk menghitung nilai akurasi yaitu Rumus 2.5.

$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$	Rumus 2. 13 Rumus Akurasi
--	---------------------------

Keterangan:

TP = *True Positive*

TN = *True Negative*

FP = *False positive*

FN = *False negative*

Presisi adalah parameter yang menunjukkan rasio data positif benar dari hasil prediksi model. Rumus untuk menghitung presisi digunakan Rumus 2.6.

$Precision = \frac{TP}{TP+FP}$	Rumus 2. 14 Rumus Presisi
--------------------------------	---------------------------

Keterangan:

TP = *True Positive*

TN = *False positive*

*Recall* adalah parameter untuk menunjukkan tingkat keberhasilan model dalam mendeteksi data positif dari seluruh data positif yang ada pada dataset. Perhitungan *recall* dapat dilakukan dengan Rumus 2.7.

$Recall = \frac{TP}{TP+FN}$	Rumus 2. 15 Rumus <i>Recall</i>
-----------------------------	---------------------------------

Keterangan:

TP = *True Positive*

FN = *False negative*

F1-Score adalah *harmonic mean* dari presisi (positif) dan *recall* (negative). Presisi adalah jumlah hasil positif sejati dibagi dengan jumlah semua hasil positif (semua yang diprediksi positif dan yang sebenarnya positif), sementara *recall* adalah jumlah hasil positif sejati dibagi dengan jumlah semua sampel yang sebenarnya positif. F1-Score dapat diperoleh dengan melakukan perhitungan menggunakan Rumus 2.8.

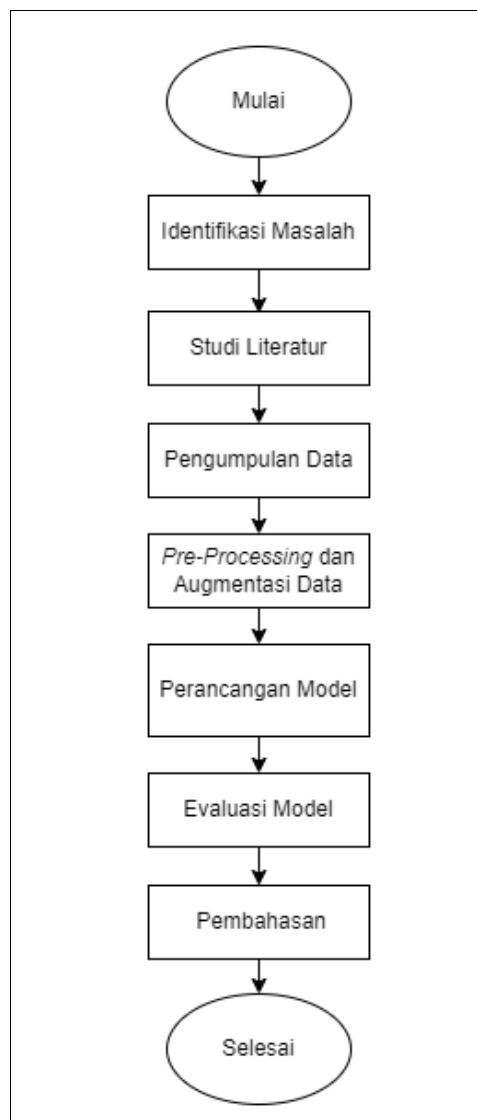
$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$	Rumus 2. 16 Rumus F1-Score
--	----------------------------

## BAB III

### METODE PENELITIAN

#### 3.1 Alur Penelitian

Dalam melakukan penelitian, terdapat beberapa tahapan penting yang dilalui dalam proses pembuatan model. Tahapan-tahapan ini melibatkan identifikasi masalah dan studi literatur, pengumpulan dan pemrosesan data gambar, perencanaan perancangan model, evaluasi serta pembahasan. Detail dari alur penelitian dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Alur Penelitian

## 3.2 Penjabaran Alur Penelitian

### 3.2.1 Identifikasi Masalah

Tahap awal dari penelitian ini peneliti melakukan identifikasi masalah yang terjadi di PTPN IV Unit Bah-Butong dengan menghubungi langsung pihak Perkebunan teh PTPN IV Unit Bah-Butong. Pada tahap ini, terdapat permasalahan yaitu kesalahan deteksi penyakit pada daun sehingga pemberian pestisida yang tidak sesuai dengan penyakit akan membuat daun membusuk dan tidak dapat dipanen. Dari permasalahan tersebut, maka dibuatlah sebuah model untuk melakukan deteksi penyakit pada taun teh secara otomatis dengan memanfaatkan model deteksi objek YOLO.

### 3.2.2 Studi Literatur

Dari permasalahan yang ada, peneliti kemudian akan melakukan tinjauan studi terkait topik klasifikasi penyakit pada tanaman menggunakan beberapa sumber seperti jurnal, buku, artikel dan sumber-sumber lain. Langkah ini bertujuan untuk memperoleh gambaran yang komprehensif tentang pendekatan dan metodologi terkini dalam klasifikasi penyakit tanaman.

### 3.2.3 Pengumpulan Data

Pada tahap ini, dilakukan pengumpulan data citra daun teh dari Perkebunan teh PTPN IV Unit Bah-Butong sebagai data latih, data uji dan data validasi. Data yang dikumpulkan terdiri dari data citra daun teh yang terkena penyakit *Brown Blight* dan *Gray Blight*. Pengambilan data dilakukan dengan menggunakan kamera *Smart phone* OPPO A78 5G dengan spesifikasi penyimpanan 128 GB, RAM 8 GB, 2 kamera belakang yaitu kamera utama 50 MP dan kamera hitam putih 2MP. Jarak pengambilan gambar tidak lebih dari 10cm antara kamera dan daun sehingga objek penyakit akan terlihat jelas. Jumlah data yang diambil adalah sebanyak 600 data gambar yang terdiri dari daun berpenyakit *Gray Blight* dan *Brown Blight* masing-masing berjumlah 300 data. Sebelum digunakan, nantinya data yang telah dikumpulkan akan divalidasi terlebih dahulu oleh ahli untuk memastikan kesesuaian dan kelayakan data sebelum digunakan. Pihak yang akan melakukan validasi adalah ahli atau pakar di bidang tanaman. Hanya data yang lolos proses validasi yang akan digunakan untuk membangun model.

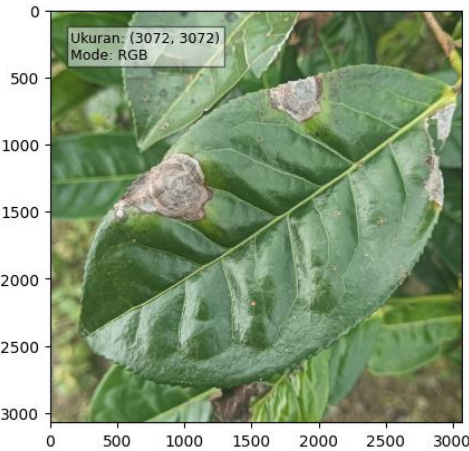
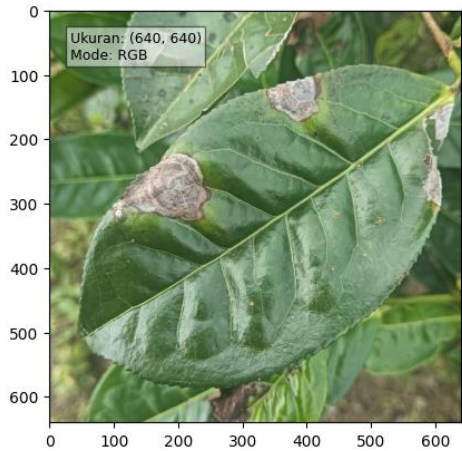
### 3.2.4 *Pre-processing* dan Augmentasi Data

Tahapan ini merupakan tahapan persiapan data citra yang telah diambil sebelum diproses oleh model. Tahapan *pre-processing* dilakukan untuk menyamaratakan ukuran seluruh dataset agar lebih mudah dipahami model sementara augmentasi dilakukan untuk meningkatkan kualitas dan kuantitas data agar model dapat melakukan tugasnya dengan lebih baik.

#### 3.2.4.1 *Pre-processing*

Pada tahapan ini, data diproses untuk menghasilkan kumpulan data dengan ukuran sama yaitu 640x640. Ukuran data dipilih berdasarkan beberapa pertimbangan, termasuk kemampuan model dalam mengolah data dengan ukuran besar. Pada dokumen resmi YOLO dari Ultralytics tertera bahwa ukuran yang ditargetkan dalam proses pelatihan model adalah 640 sebagai *default*. Penentuan ukuran ini berpengaruh pada hasil akurasi akhir dan kompleksitas model yang dibuat. Ukuran data yang terlalu besar dapat menyebabkan model memiliki waktu komputasi yang lebih lama sementara ukuran yang terlalu kecil membuat model kesulitan membaca informasi penting dari gambar. Oleh karena itu, sebelum diproses ke dalam model, dataset yang telah dikumpulkan akan diubah menjadi ukuran 640x640 seperti pada Tabel 3.1.



Tabel 3. 1. Sebelum dan Sesudah *Pre-processing*

Sebelum <i>Resizing</i>	Setelah <i>Resizing</i>
 <p>Ukuran: (3072, 3072) Mode: RGB</p>	 <p>Ukuran: (640, 640) Mode: RGB</p>

### 3.2.4.2 Anotasi Data

Anotasi pada penelitian dilakukan menggunakan jenis anotasi poligon agar mengikuti bentuk objek yang dideteksi pada gambar dan hasil anotasi menjadi lebih akurat. Anotasi poligon memungkinkan pengguna untuk menggambar garis-garis yang mengelilingi objek dengan tepat melalui titik-titik koordinat, memberikan hasil yang lebih akurat dibandingkan dengan anotasi *bounding box*. Teknik ini sangat cocok untuk objek dengan bentuk kompleks atau tidak beraturan seperti objek penyakit pada daun teh pada penelitian ini. Contoh gambar dengan anotasi poligon dapat dilihat pada Tabel 3.2.



Tabel 3. 2 Hasil Anotasi

Sebelum Anotasi	Setelah Anotasi
	

### 3.2.4.3 Labeling Data

Pada penelitian ini labeling dilakukan secara manual bersamaan dengan proses anotasi pada gambar. Seluruh dataset yang sudah dianotasi terbagi menjadi dua kelas yaitu kelas penyakit *Gray Blight* yang disingkat menjadi kelas 'G' dan kelas *Brown Blight* yang disingkat menjadi kelas 'B'. Keakuratan dan konsistensi dalam anotasi dan labeling sangat penting karena data akan digunakan untuk melatih model deteksi objek. Contoh data yang telah diberi label kelas dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Hasil Labeling



Sebelum <i>Labeling</i>	Setelah <i>Labeling</i>
	

#### 3.2.4.4 Rotasi Data

Pada penelitian ini, teknik augmentasi data rotasi digunakan untuk meningkatkan kuantitas dataset. Dua jenis rotasi yang diterapkan adalah rotasi searah jarum jam (*clockwise*) dan rotasi berlawanan arah jarum jam (*counter-clockwise*). Rotasi *clockwise* adalah rotasi gambar sejauh 90 derajat searah jarum jam, sementara rotasi *counter-clockwise* adalah rotasi gambar sejauh 90 derajat melawan arah jarum jam.

Penggunaan rotasi ini bertujuan untuk menghasilkan variasi dalam posisi objek pada gambar, sehingga model dapat mempelajari fitur-fitur objek dalam berbagai orientasi. Dengan demikian, model akan lebih mampu mengenali objek dengan lebih baik meskipun dalam posisi yang berbeda dari yang ada di data asli. Hal ini sangat penting dalam meningkatkan kemampuan generalisasi model pada situasi nyata, di mana objek bisa muncul dalam berbagai orientasi. Gambar hasil proses augmentasi rotasi dapat dilihat pada Tabel 3.4.



Tabel 3. 4 Hasil Rotasi

Hasil Rotasi <i>Clockwise</i>	Hasil Rotasi <i>Counter Clockwise</i>
	

### 3.2.4.5 *Mosaic Data*

*Mosaic data* merupakan teknik penggabungan 4 gambar menjadi satu gambar tunggal dalam bentuk *grid*. Dalam satu gambar hasil *mosaic data*, ada kemungkinan objek yang dimuat memiliki kelas yang berbeda. Penerapan teknik *mosaic data* ini dilakukan untuk meningkatkan variasi dataset sehingga model memiliki gambaran yang lebih banyak tentang data yang dipelajari sebelum diuji dengan data uji. Contoh hasil proses *mozaic data* dapat dilihat pada Tabel 3.5.

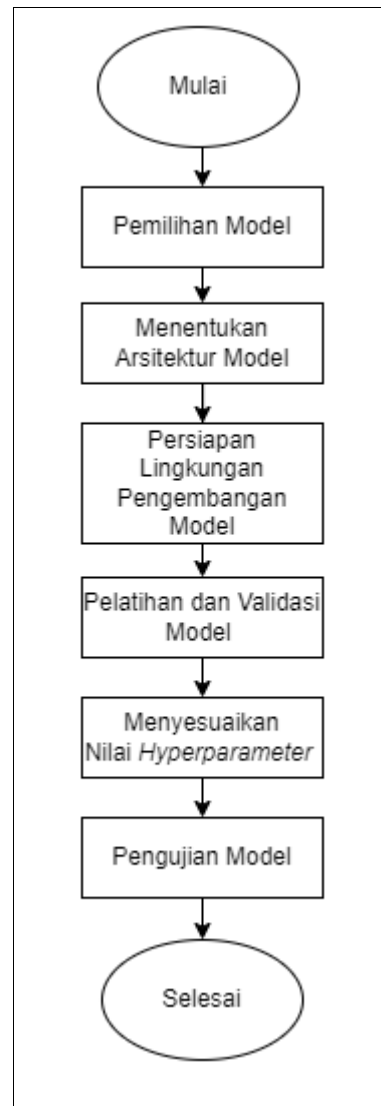
Tabel 3. 5 Hasil Rotasi

Sebelum	Setelah Mozaic
	



### 3.2.5 Perancangan Model

Pada tahap pengembangan model, peneliti menggunakan YOLOv8 sebagai basis dari model. Perancangan model dilakukan dalam beberapa tahap sesuai dengan alur pada Gambar 3.2.



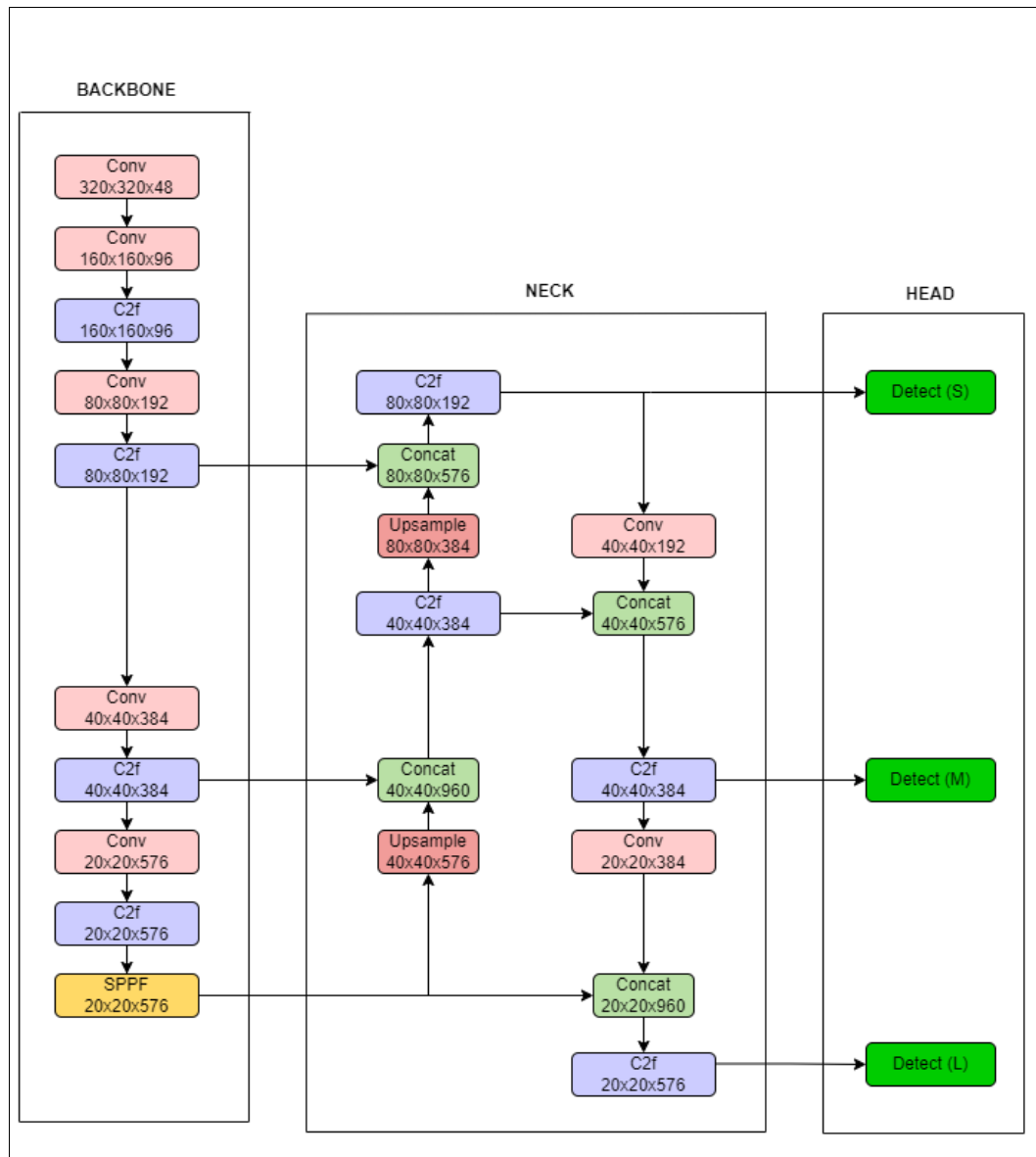
Gambar 3. 2 Alur Perancangan Model

#### 3.2.5.1 Pemilihan Model

Pada tahap ini dilakukan pemilihan model YOLO yang akan digunakan. Dan model yang dipilih adalah model YOLOv8 yang merupakan hasil pengembangan terbaru dari versi YOLO. Penjelasan dan keunggulan YOLOv8 dapat dilihat pada bab sebelumnya.

### 3.2.5.2 Menentukan Arsitektur Model

Pada penelitian ini, terdapat tiga bagian utama pada model yaitu *backbone*, *neck* dan *head*. Selama memproses data, ada beberapa lapisan model yang digunakan yaitu 7 lapisan konvolusi, 8 lapisan C2f, 1 lapisan SPPF, 4 lapisan Concat dan 2 lapisan Upsample. Arsitektur model YOLOv8 dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Ilustrasi Pemrosesan Gambar pada YOLOv8

Lapisan konvolusi pada YOLOv8 terdiri dari ukuran *kernel*, *stride* dan *padding* yang sama yaitu *kernel* 3, *stride* 2 dan *padding* 1. *Kernel* 3 artinya ukuran filter yang digunakan selama melintasi gambar adalah 3x3. Sementara *stride* berukuran 2 menunjukkan jumlah piksel yang akan dilewati filter pada setiap pergerakan. Dan

*padding* 1 menunjukkan jumlah piksel yang akan ditambahkan di sekitar tepi gambar sebelum menerapkan filter.

Dari Gambar 3.3 dapat dilihat ukuran keluaran dari masing-masing lapisan pada model. Gambar *input* yang digunakan memiliki ukuran 640x640x3, di mana 640 adalah tinggi dan lebar gambar, sedangkan 3 adalah jumlah *channel* (merah, hijau, dan biru). Untuk memproses gambar ini, lapisan konvolusi dengan *kernel* berukuran 3x3, *stride* 2, dan *padding* 1 digunakan. Dengan memasukkan nilai-nilai ini dalam rumus, diperoleh ukuran keluaran:

$$\begin{aligned}
 \text{Keluaran} &= \frac{\text{Ukuran Input} + 2 \times P - K}{S} + 1 \\
 &= \frac{640 + 2 \times 1 - 3}{2} + 1 \\
 &= \frac{640 + 2 - 3}{2} + 1 \\
 &= \frac{640 + 2 - 3}{2} + 1 \\
 &= \frac{639}{2} + 1 \\
 &= 319,5 + 1 \\
 &= 320,5
 \end{aligned}$$

Dalam konteks ukuran gambar, hasil ini biasanya dibulatkan menjadi bilangan bulat, sehingga ukuran keluaran setelah proses konvolusi menjadi 320x320. Oleh karena itu, tinggi dan lebar *output* dari lapisan konvolusi ini adalah 320x320. Selain itu, jumlah *channel output* bergantung pada jumlah filter yang digunakan dalam lapisan konvolusi. Ukuran filter dari masing-masing lapisan dapat dilihat pada Tabel 3.1. Pada lapisan konvolusi pertama, jumlah filter adalah 48 yang artinya jumlah *channel output* adalah 48. Dengan demikian, gambar berukuran 640x640x3 setelah diproses oleh lapisan konvolusi pertama akan menghasilkan *output* berukuran 320x320x48, di mana setiap filter menghasilkan satu *channel* dalam *output*. Perhitungan ini berlaku untuk seluruh lapisan konvolusi yang terdapat pada model. Menggunakan rumus dan perhitungan yang sama, dihasilkan gambar

dengan ukuran 160x160x96 pada lapisan konvolusi kedua. Hasil ini akan diproses kembali di lapisan selanjutnya yaitu lapisan C2f.

Layer C2f dalam model YOLO terdiri dari beberapa komponen utama yang bekerja bersama untuk menghasilkan fitur yang lebih kaya dan mendalam. Layer pertama adalah lapisan konvolusi (cv1) yang menerima *input* dengan 96 *channels* dan menghasilkan *output* dengan jumlah *channels* yang sama, yaitu 96. *Output* dari cv1 kemudian dibagi menjadi dua bagian, masing-masing dengan 48 *channels*, dan diproses lebih lanjut melalui dua blok *Bottleneck*. Setiap blok *Bottleneck* terdiri dari dua lapisan konvolusi, di mana masing-masing lapisan menerima *input* 48 *channels* dan juga menghasilkan *output* 48 *channels*. Dengan demikian, total *output* dari kedua blok *Bottleneck* adalah 48 *channels* dari blok pertama ditambah 48 *channels* dari blok kedua, sehingga totalnya menjadi 96 *channels*. *Output* dari blok *Bottleneck* ini kemudian dikonkatenasi dengan *output* awal dari cv1 yang berjumlah 96 *channels*. Hasil konkatenasi ini menghasilkan total 192 *channels* (96 *channels* dari cv1 ditambah 96 *channels* dari *Bottleneck*). Lapisan konvolusi kedua (cv2) kemudian menerima 192 *channels* ini sebagai *input* dan memprosesnya untuk menghasilkan *output* dengan 96 *channels*. Hasil *output* 640x640x96 ini kemudian akan digunakan untuk proses pada lapisan selanjutnya.

Pada akhir bagian *backbone* model, terdapat lapisan SPPF yang terdiri dari beberapa blok seperti pada Gambar 2.7. Dimulai dengan lapisan konvolusi pertama (cv1) yang menerima *input* dengan 576 *channels* dan menghasilkan *output* dengan 288 *channels*. Kemudian, dilakukan operasi Max Pooling dengan *kernel* 5x5 dan *stride* 1, yang mempertahankan ukuran spasial tetapi memperkaya fitur. Pooling ini dilakukan sebanyak tiga kali untuk menangkap informasi kontekstual dari berbagai skala. Hasil dari *pooling* ini kemudian digabungkan (*concatenate*) dengan *output* asli dari cv1 yang memiliki 288 *channels*. Oleh karena itu, hasil gabungan ini mencakup *output* asli cv1 (288 *channels*) ditambah hasil dari tiga *pooling* (masing-masing 288 *channels*), sehingga totalnya menjadi 1152 *channels*. Lapisan konvolusi kedua (cv2) kemudian memproses 1152 *channels* ini dan menghasilkan *output* dengan 576 *channels*. Proses ini memungkinkan model untuk menangkap dan mengintegrasikan informasi dari berbagai skala spasial, yang sangat berguna untuk deteksi objek dengan berbagai ukuran

Layer SPPF dimulai dengan lapisan konvolusi (cv1) yang mengurangi jumlah *channels* dari 576 menjadi 288 sambil mempertahankan informasi penting. Selanjutnya, operasi Max Pooling dilakukan dengan *kernel* 5x5 dan *stride* 1, yang mempertahankan ukuran spasial tetapi memperkaya fitur dengan informasi kontekstual dari berbagai skala. Hasil dari *pooling* ini kemudian digabungkan (*concatenate*) dengan *input* asli serta dua hasil *pooling* lainnya, yang menghasilkan total 1152 *channels*. Akhirnya, lapisan konvolusi kedua (cv2) digunakan untuk menggabungkan dan mengurangi jumlah *channels* menjadi 576. Ukuran spasial gambar tidak berubah, sehingga hasil *output* dari lapisan SPPF ini adalah 20x20x576.

Perulangan proses perhitungan pada lapisan yang sama terjadi pada bagian *neck* dan *head* pada model. Pada akhir dari proses, ada bagian *head* yang merupakan bagian yang bertanggung jawab untuk memberikan *bounding box* dan prediksi kelas pada gambar. Untuk prediksi kelas, YOLOv8 menggunakan *Binary Cross-Entropy* (BCE) *Loss* dengan perhitungan Rumus 2.1. Untuk melakukan perhitungan, diperlukan nilai *Threshold* untuk menentukan kelas dari gambar yang diproses model. Pada penelitian ini, nilai *Threshold* yang digunakan adalah 0,7 dimana jika nilai *confidence* dan IoU yang diperoleh >0,7 maka *bounding box* akan ditampilkan pada gambar dan hasil prediksi dianggap valid.

Pada penelitian ini, model melakukan prediksi dengan hasil *bounding box* yaitu nilai  $x_{center}$ ,  $y_{center}$ ,  $w$  dan  $h$  berturut turut adalah 50,50,40 dan 20. Nilai-nilai ini selanjutnya akan digunakan untuk menghitung koordinat dari kotak pembatas yang terdiri dari nilai  $x_{min}$ ,  $y_{min}$ ,  $x_{max}$ ,  $y_{max}$ . Hasil perhitungan nilai-nilai dapat dilihat pada Tabel 3.6.

Tabel 3. 6 Hasil Perhitungan  $x_{min}$ ,  $y_{min}$ ,  $x_{max}$ ,  $y_{max}$

Parameter	Rumus	Hasil
$x_{min}$	$x_{min} = x_{center} - \frac{w}{2}$	30
$y_{min}$	$y_{min} = y_{center} - \frac{h}{2}$	40

$x_{max}$	$x_{max} = x_{center} + \frac{w}{2}$	70
$y_{max}$	$y_{max} = y_{center} + \frac{h}{2}$	60

Setelah mendapatkan nilai-nilai tersebut, dihitung nilai *Intersection Width* dan *Intersection Height* menggunakan Rumus 2.6 dan 2.7.

$$\begin{aligned}
 \text{Intersection Width} &= \min(x_{max}, X_{max}) - \max(x_{min}, X_{min}) \\
 &= \min(70, 75) - \max(30, 35) \\
 &= 70 - 35 \\
 &= 35
 \end{aligned}$$

$$\begin{aligned}
 \text{Intersection Height} &= \min(y_{max}, Y_{max}) - \max(y_{min}, Y_{min}) \\
 &= \min(60, 65) - \max(40, 45) \\
 &= 60 - 45 \\
 &= 15
 \end{aligned}$$

Nilai *Intersection Area* diperoleh dari hasil perkalian antara *intersection width* dan *intersection height* seperti yang ditunjukkan pada Rumus 2.8. Hasil perkalian dua nilai ini adalah 35x15 menjadi 525. Jadi nilai *Intersection Area* yang diperoleh adalah 525. Selanjutnya dihitung area prediksi dan area *ground truth* menggunakan Rumus 2.9 dan Rumus 2.10.

$$\begin{aligned}
 \text{Area of Prediction} &= (x_{max} - x_{min}) \times (y_{max} - y_{min}) \\
 &= (70 - 30) \times (60 - 40) \\
 &= 40 \times 20 \\
 &= 800
 \end{aligned}$$

$$\begin{aligned}
\text{Area of Ground Truth} &= (X_{max} - X_{min}) \times (Y_{max} - Y_{min}) \\
&= (75 - 35) \times (65 - 45) \\
&= 40 \times 20 \\
&= 800
\end{aligned}$$

Selanjutnya dilakukan perhitungan total area yang mencakup prediksi kotak pembatas dengan *ground truth* dengan menjumlahkan area kotak pembatas hasil prediksi dengan *ground truth* dikurangi dengan nilai *intersection*..

$$\begin{aligned}
\text{Union Area} &= 800 + 800 - 525 \\
&= 1075
\end{aligned}$$

Terakhir, dilakukan perhitungan nilai *Intersection over Union* (IoU) yang menunjukkan *overlap* antara hasil prediksi dan *ground truth*.

$$\begin{aligned}
\text{IoU} &= \frac{\text{Intersection Area}}{\text{Union Area}} \\
&= \frac{525}{1075} \\
&= 0,48
\end{aligned}$$

Pada kasus ini, diperoleh nilai *intersection* sebesar 525 dan union 1075, menghasilkan nilai IoU sebesar 0,48 atau 48%. Sesuai dengan nilai yang telah ditentukan, nilai ambang batas *confidence* dan IoU kotak pembatas yang dianggap valid adalah 0,7. Sehingga nilai IoU sebesar 0,48 menunjukkan bahwa kotak pembatas dianggap tidak valid pada hasil prediksi model.

Sementara untuk penentuan kelas, dipengaruhi oleh nilai ambang batas yang ditentukan saat validasi model yang dalam penelitian ini adalah 0,7. Jika hasil probabilitasnya lebih dari nilai ambang batas yang ditentukan maka objek akan terdeteksi sebagai kelas positif (kelas B), namun jika lebih rendah dari nilai ambang batas maka prediksi objek termasuk kelas G. Pada ilustrasi perhitungan model, nilai

probabilitas yang dihasilkan untuk suatu prediksi objek adalah 0,8. Artinya objek yang diidentifikasi termasuk ke dalam kelas B.

### 3.2.5.3 Persiapan Lingkungan Pengembangan Model

Pada tahap ini dilakukan persiapan lingkungan untuk membangun model mulai dari melakukan menghubungkan media ke *Graphic Processing Unit* (GPU), melakukan instalasi dan *cloning* model dan melakukan instalasi *pre-Trained* model untuk akses data dan melakukan instalasi *package-package* yang diperlukan selama membangun model.

### 3.2.5.4 Pelatihan dan Validasi Model

Pada pelatihan model, akan digunakan data citra yang diambil langsung dari perkebunan PTPN IV berupa data mentah yang akan divalidasi oleh ahli terlebih dahulu untuk kemudian dibagi menjadi data latih, data validasi dan data uji sebelum diproses oleh model. Data citra pada pelatihan menggunakan data citra daun teh yang terkena penyakit *Brown Blight* dan daun teh yang terkena penyakit *Gray Blight*. Pada tahapan ini, model akan mempelajari dataset gambar yang sudah menjalani proses *pre-processing* dan augmentasi untuk menghasilkan model yang dapat melakukan identifikasi jenis penyakit pada daun teh.

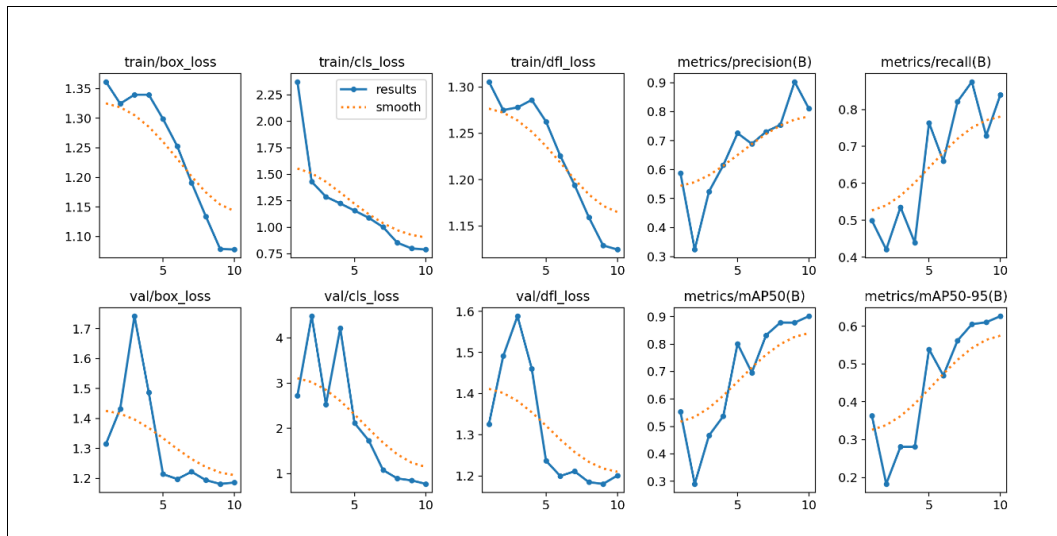
Setelah dilakukan pelatihan, dilakukan validasi model untuk melihat apakah model yang dibangun tidak mengalami *overfitting* atau *underfitting* sebelum dilakukan pengujian. Validasi dilakukan menggunakan dataset validasi yang tidak digunakan dalam pelatihan model. Hasil dari matriks validasi ini akan digunakan sebagai acuan penentuan nilai *hyperparameter* di tahapan selanjutnya.

### 3.2.5.5 Menyesuaikan Nilai *Hyperparameter*

Pada tahapan ini, ada beberapa perubahan nilai yang dilakukan dengan tujuan optimalisasi kinerja model termasuk perubahan nilai *epoch*, *batch*, *confidence threshold*, *IoU threshold*, dan *learning rate*. Pemilihan nilai epoch dilakukan dengan mempertimbangkan sumber daya komputasi dan *resource* yang digunakan dalam penelitian ini. Jumlah dataset pelatihan yang digunakan dalam membangun model hanya sebanyak 588 data (sesudah di augmentasi). Standar jumlah epoch dalam pelatihan model *deep learning* adalah 100 epoch. Namun pada penelitian ini

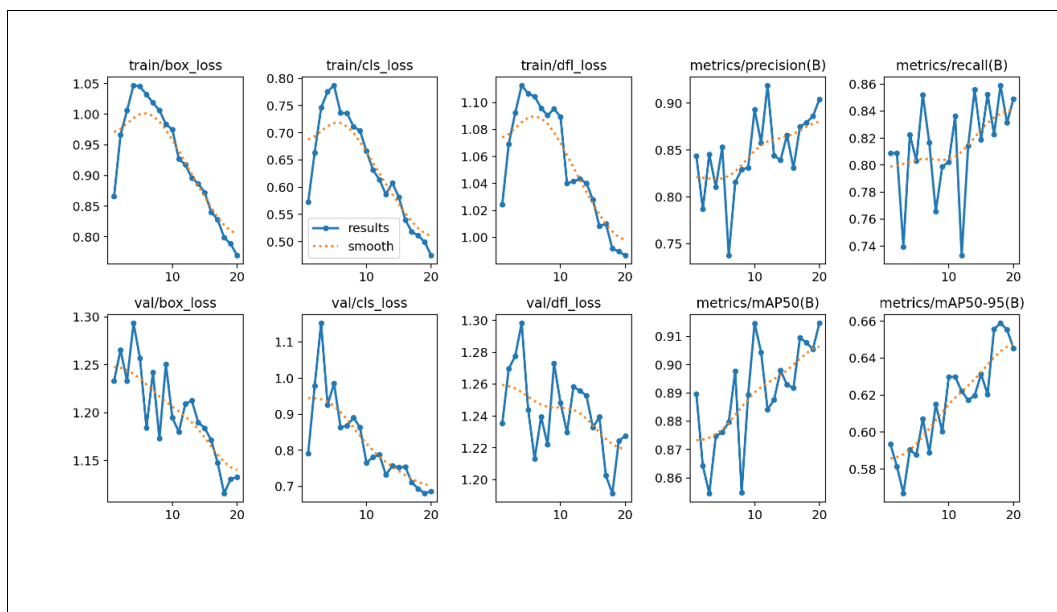


digunakan sebanyak 10, 20 dan 35 epoch dengan mempertimbangkan terjadinya *overfitting* selama pelatihan. Dari ketiga jumlah epoch berbeda sebelumnya akan dilakukan perbandingan untuk mengetahui hasil pelatihan terbaik.



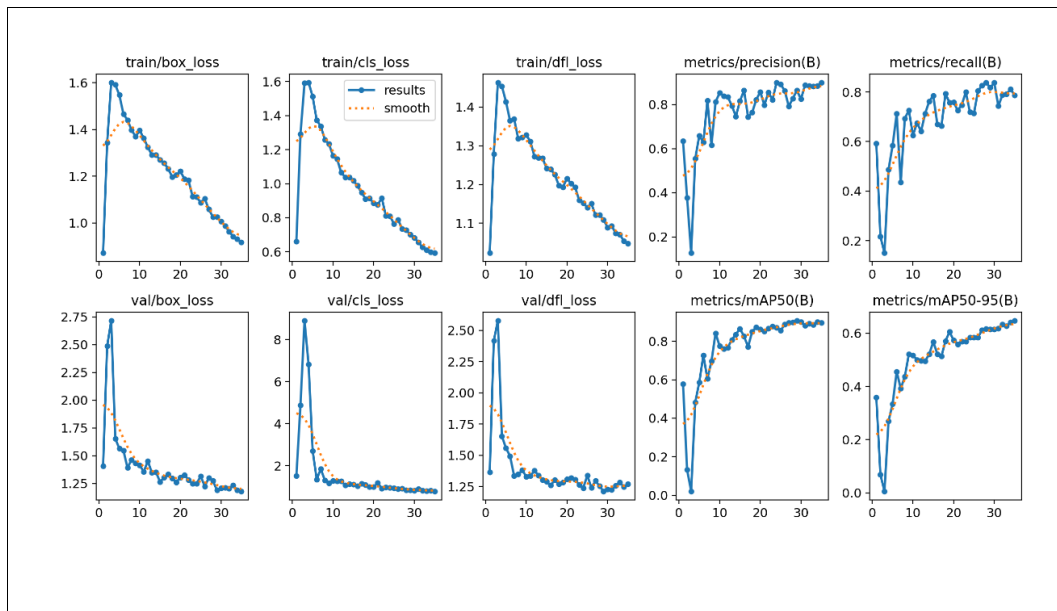
Gambar 3. 4 Hasil Pelatihan 10 Epoch

Gambar 3.4 di atas menunjukkan bahwa model mengalami peningkatan kinerja secara keseluruhan, baik pada data pelatihan maupun validasi. Semua metrik performa (*precision*, *recall*, mAP) menunjukkan tren peningkatan, sementara nilai loss menurun, yang merupakan indikasi bahwa model belajar dengan baik.



Gambar 3. 5 Hasil Pelatihan 20 Epoch

Gambar 3.5 menunjukkan model yang lebih matang dengan stabilitas yang lebih baik setelah lebih banyak epoch pelatihan (20 epoch). Ada peningkatan yang lebih konsisten pada metrik, menunjukkan model yang lebih baik dibandingkan dengan yang ada pada gambar pertama. Dibandingkan Gambar 3.4 (hasil 10 epoch), pada pelatihan dengan 20 epoch diperoleh hasil *loss* yang lebih stabil dan menurun secara konsisten serta *precision*, *recall*, dan mAP yang lebih stabil dan terus meningkat.



Gambar 3. 6 Hasil Pelatihan 35 Epoch

Gambar 3.6 menunjukkan bahwa setelah 35 epoch, model telah mencapai stabilitas dalam hal *loss* dan metrik. Dari grafik dapat dilihat *precision*, *recall*, dan mAP mencapai nilai yang tinggi dan stabil, menunjukkan bahwa model telah belajar secara efektif dan menghasilkan prediksi yang andal.

Dari tiga pelatihan yang dilakukan dengan nilai epoch yang berbeda, terlihat bahwa peningkatan jumlah epoch memberikan hasil stabilitas dan kinerja model yang lebih baik. Fluktuasi pada awal pelatihan berkurang secara signifikan setelah digunakan lebih banyak epoch, menunjukkan bahwa model semakin akurat dalam deteksi objek dan klasifikasi setelah lebih banyak waktu pelatihan. Namun, peningkatan nilai epoch yang berlebihan juga dapat menghasilkan kondisi *overfitting* sehingga pada penelitian ini, di mana model telah mencapai kondisi konvergensi maka epoch yang akan digunakan dalam penelitian ini adalah 35 epoch.

Sementara nilai *batch* menggunakan nilai 8, namun dalam upaya mempercepat pencapaian konvergensi model selama pelatihan dilakukan juga pelatihan model menggunakan *batch* berukuran 16. Pemilihan nilai *IoU Threshold* berdasarkan dari *range* ketepatan suatu prediksi dimana standar ketepatan suatu prediksi dikatakan baik adalah diatas 70%. Oleh karena itu, dalam pemodelan untuk mengidentifikasi jenis penyakit pada daun teh ini nilai untuk *IoU threshold* diatur sebesar 0.7 yang artinya kotak pembatas dianggap valid jika lebih dari 70% atau lebih. Sama halnya dengan nilai *confidence score*, kotak pembatas yang ditampilkan adalah kotak pembatas dengan nilai lebih tinggi atau sama dengan 0.7. Sementara pada *learning rate*, pemilihan nilai dimulai dengan menggunakan nilai *learning rate default* model yaitu 0,01 dan menggunakan nilai 0,001 yang disesuaikan dengan nilai *batch* yang digunakan. Penggunaan nilai *batch* yang cenderung kecil pada model juga sejalan dengan langkah pembaruan bobot model selama pelatihan untuk menjaga stabilitas pelatihan. Nilai-nilai *hyperparameter* ini nantinya akan dikombinasikan dengan *hyperparameter* lain untuk mencapai hasil akhir terbaik.

Dari hasil penjabaran sebelumnya, nilai yang digunakan yaitu epoch sebesar 35, batch dengan nilai sebesar 8 dan 16, nilai *learning rate* sebesar 0,01 dan 0,001 serta *IoU* dan *confidence threshold* sebesar 0,7. Dari nilai-nilai tersebut diperoleh 4 konfigurasi berbeda untuk tiap pelatihan dan dapat dilihat pada Tabel 3.6 yang terdiri dari *Train 1*, *Train 2*, *Train 3* dan *Train 4*. Pelatihan dilakukan untuk mendapatkan nilai dengan hasil paling optimal dari setiap *hyperparameter*, dilakukan percobaan dengan konfigurasi nilai-nilai pada Tabel 3.6.

Tabel 3. 7 Kombinasi Nilai *Hyperparameter*

Konfigurasi Nilai	Nilai <i>Hyperparameter</i>				<i>Learning rate</i>
	<i>Epoch</i>	<i>Batch</i>	<i>Threshold</i>		
			<i>Confidence</i>	<b>IoU</b>	
<i>Train 1</i>	35	8	0.7	0.7	0.01
<i>Train 2</i>	35	16	0.7	0.7	0.01
<i>Train 3</i>	35	8	0.7	0.7	0.001
<i>Train 4</i>	35	16	0.7	0.7	0.001



### 3.2.5.6 Pengujian Model

Pada penelitian ini, pengujian dilakukan untuk mengetahui apakah model dapat melakukan identifikasi penyakit pada daun teh. Pengujian dilakukan menggunakan data uji yang terdiri dari dua jenis penyakit yang diakuisisi di lapangan secara langsung oleh peneliti.




### 3.2.6 Evaluasi

Tahap evaluasi dilakukan untuk mengukur kemampuan model dalam melakukan tugas identifikasi dengan membandingkan hasil prediksi dengan nilai sebenarnya yang digunakan saat pelatihan model. Evaluasi model dilakukan menggunakan *Confusion Matrix* untuk menghitung nilai *Accuracy*, *Precision*, *Recall* dan nilai *F1-Score*. Hasil evaluasi akan digunakan sebagai pertimbangan perubahan nilai *hyperparameter* model jika diperlukan. Gambaran perhitungan evaluasi menggunakan *Confusion Matrix* dapat dilihat pada Tabel 3.7.

Tabel 3. 8 Contoh Hasil Pengujian

No	Gambar <i>Input</i>	Hasil Prediksi Model	Data Yang Sebenarnya	Nilai Komponen <i>Confusion Matrix</i>			
				TP	FP	TN	FN
1		<i>Brown Blight</i>	<i>Brown Blight</i>	1	0	0	0
2		<i>Brown Blight</i>	<i>Brown Blight</i>	1	0	0	0

3		<i>Brown Blight</i>	<i>Brown Blight</i>	1	0	0	0
4		<i>Brown Blight</i>	<i>Brown Blight</i>	1	0	0	0
5		<i>Gray Blight</i>	<i>Brown Blight</i>	0	0	0	1
6		<i>Gray Blight</i>	<i>Gray Blight</i>	<b>0</b>	<b>0</b>	<b>1</b>	0
7		<i>Gray Blight</i>	<i>Gray Blight</i>	<b>0</b>	<b>0</b>	<b>1</b>	0

8		<i>Gray Blight</i>	<i>Gray Blight</i>	0	0	1	0
9		<i>Brown Blight</i>	<i>Gray Blight</i>	0	1	0	0
10		<i>Gray Blight</i>	<i>Gray Blight</i>	0	0	1	0

Berdasarkan tabel di atas, TP (True Positive) merupakan jumlah data positif yang diprediksi benar, dengan nilai sebesar 4. TN (True Negative) adalah jumlah data negatif yang diprediksi benar, juga dengan nilai sebesar 4. FP (False Positive) merujuk pada jumlah data negatif yang diprediksi salah, yaitu sebanyak 1. Sedangkan FN (False Negative) adalah jumlah data positif yang diprediksi salah, juga sebesar 1.

Tabel 3. 9 Contoh Akumulasi Hasil Klasifikasi

Hasil Klasifikasi Model		Nilai Komponen <i>Confusion Matrix</i>			
Terdeteksi	Tidak Terdeteksi	TP	FP	TN	FN
10	0	4	1	4	1

Tabel 3. 10 Contoh Hasil Perhitungan

<b>Akurasi</b>	<b>Hasil Akurasi</b>	<b>Presisi</b>	<b>Hasil Presisi</b>	<b>Recall</b>	<b>Hasil Recall</b>	<b>F1- Score</b>
$\frac{4 + 0}{4 + 1 + 1 + 4}$	80%	$\frac{4}{4 + 1}$	80%	$\frac{4}{4 + 1}$	80%	0.8

### 3.2.7 Pembahasan

Pada tahap ini dijabarkan hasil analisa terkait hasil kerja model setelah dilakukan pengujian dan evaluasi. Analisa dapat memuat penjabaran dan pembahasan dari hasil identifikasi kekurangan atau kelebihan dari model, sehingga selain dapat memberikan gambaran dari kinerja model juga dapat digunakan sebagai bahan pertimbangan dalam perbaikan dan pengembangan hingga acuan pengambilan keputusan untuk penelitian selanjutnya.

## 3.3 Alat dan Bahan

### 3.3.1 Alat

Alat yang digunakan dalam pengerjaan penelitian ini, yaitu:

1. Laptop dengan spesifikasi sistem operasi Windows 11, *processor* Intel® Celeron®, memori 4 GB, SSD 240 GB.
2. Kamera *Smartphone* OPPO A78 5G dengan spesifikasi penyimpanan 128 GB, RAM 8 GB, 2 kamera belakang yaitu kamera utama 50 MP dan kamera hitam putih 2MP.

### 3.3.2 Bahan

Adapun bahan yang diperlukan dalam penelitian ini adalah:

1. Dataset diperoleh dari Perkebunan teh PTPN IV Unit Bah-Butong yang disusun sendiri oleh peneliti, terdiri dari daun yang terkena penyakit *Brown Blight* dan *Gray Blight*.
2. Dokumen-dokumen yang menjadi panduan pengerjaan penelitian, seperti jurnal, buku, atau artikel yang relevan.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Hasil

##### 4.1.1 Pengumpulan Data

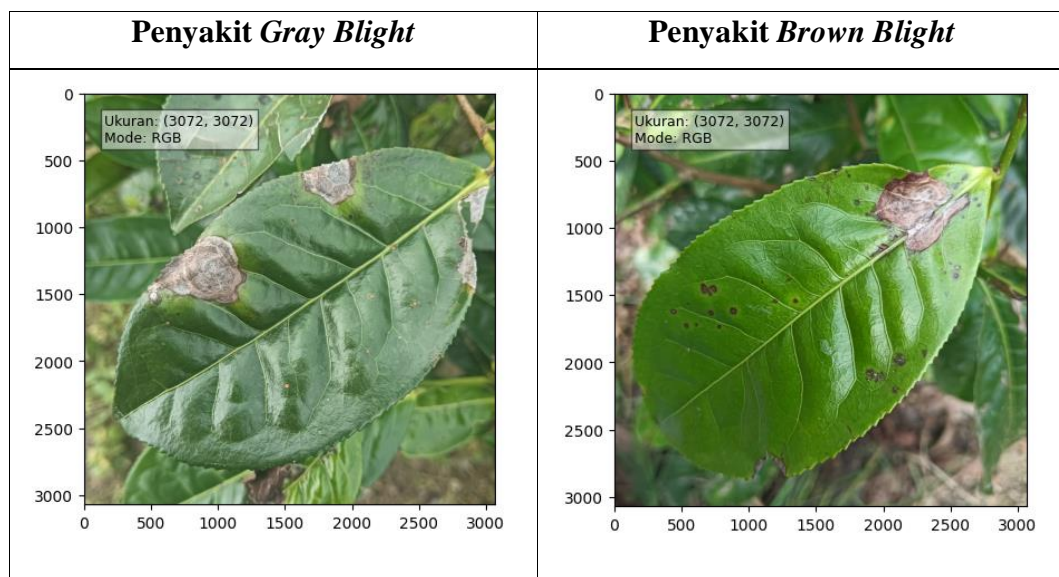
Pada penelitian ini, peneliti melakukan akuisisi data (pengumpulan data) gambar daun teh dengan total 600 gambar atau 300 gambar untuk masing-masing penyakit. Data citra yang telah dikumpulkan kemudian divalidasi oleh peneliti dari Badan Riset dan Nasional (BRIN), ibu Dr. Danarsi Diptaningsari, S.P., M. Si. Dari hasil validasi, hanya terdapat 280 dari 600 data gambar yang layak digunakan untuk membangun model, terdiri dari 182 gambar penyakit *Gray Blight* dan 98 penyakit *Brown Blight* yang akan dibagi menjadi data latih, data validasi dan data uji dengan perbandingan 70 : 20 : 10. Komposisi dataset selama pembuatan model dapat dilihat pada Tabel 4.1.

Tabel 4. 1. Komposisi Dataset

Data Latih	Data Validasi	Data Uji
196	56	28

Data hasil akuisisi lapangan memiliki ukuran yang 3072x3072 untuk semua gambar seperti yang dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Hasil Data Akuisisi Lapangan


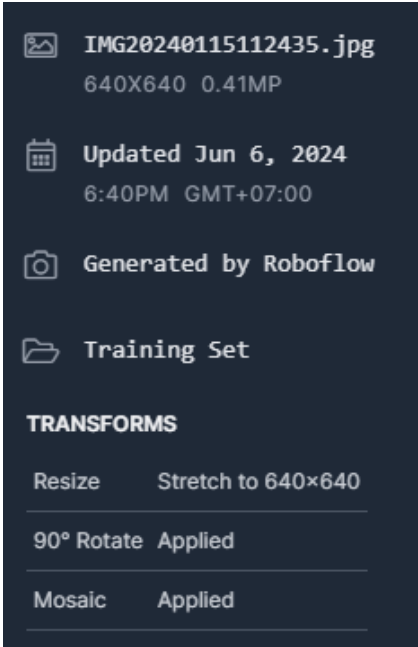




#### 4.1.2 Pre-processing dan Augmentasi Data

Tahapan *pre-processing* yang digunakan adalah *resizing* data agar memiliki ukuran yang sama dan konsisten yaitu 640x640. Selain dilakukan *pre-processing*, dilakukan juga teknik augmentasi sebagai upaya memperkaya dataset yang akan digunakan selama pelatihan. Augmentasi data pada pelatihan ini mencakup anotasi data menggunakan *bounding box* poligon, *labeling* data, penerapan teknik rotasi *clockwise* dan *counter clockwise* serta *mosaic* data. Proses augmentasi dilakukan melalui *software Online Roboflow* untuk meningkatkan variasi dan jumlah data pelatihan. Hasil dari proses ini adalah dataset baru yang berisi data-data gambar yang telah memiliki label dan kotak anotasi, data yang sudah dirotasi dan data *mosaic* berupa gabungan beberapa data tunggal ke dalam satu gambar. Hasil dari *pre-processing* dan augmentasi data latih dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Hasil *Pre-processing* dan Augmentasi Data

Gambar	Atribut
	

Dalam penelitian ini, peneliti menggunakan *software online* Roboflow untuk melakukan *pre-processing* dan augmentasi pada dataset model. Teknik *pre-processing* yang digunakan adalah *resizing*, sedangkan teknik augmentasi yang digunakan meliputi rotasi dan *mozaic*. Roboflow memiliki fitur yang

memungkinkan pengguna untuk men-generate satu gambar menjadi tiga versi, sehingga menambah jumlah dataset. Proses generasi ini hanya diterapkan pada dataset pelatihan. Awalnya, jumlah dataset pelatihan adalah 196 gambar. Setelah dilakukan augmentasi, dataset pelatihan meningkat menjadi 588 gambar ( $196 \times 3$ ). Jumlah total dataset yang digunakan dalam pembangunan model adalah 672 gambar, yang terdiri dari 588 gambar pelatihan, 56 gambar validasi, dan 28 gambar uji.

### 4.1.3 Perancangan Model

#### 4.1.3.1 Pemilihan Model

Model yang digunakan dalam penelitian ini YOLO versi 8 yang pada saat penelitian dilakukan merupakan versi terbaru dari seri model deteksi YOLO. Dalam upaya meningkatkan kinerja model dalam melakukan tugas deteksi objek, maka dilakukan pemilihan nilai *hyperparameter* yang paling sesuai untuk diterapkan.

#### 4.1.3.2 Menentukan Arsitektur Model

Arsitektur model YOLO pada penelitian menggunakan model arsitektur YOLO yang asli. Arsitektur model YOLOv8 telah memiliki lapisan *batch normalization* setiap setelah lapisan konvolusi, sehingga penerapan *batch normalization* tidak lagi diperlukan dan model akan dilatih pada arsitektur asli dari model YOLOv8.

#### 4.1.3.3 Persiapan Lingkungan Pengembangan Model

Sebelum menjalankan kode program untuk membangun model, terlebih dahulu dilakukan pengaturan pada *software* agar memudahkan peneliti. Pada penelitian ini, peneliti memanfaatkan *software Online* dari Google yaitu Google Colab untuk membangun model. Pelatihan model merupakan tahapan yang memakan sumber daya paling banyak, termasuk waktu dan perangkat sehingga perlu dilakukan inisiasi kode ke GPU *device* untuk mempercepat proses pelatihan. Pelatihan pada penelitian ini memanfaatkan akses gratis GPU dari Google Colab untuk mempercepat waktu pelatihan model. Setelah menyambungkan model ke GPU, selanjutnya dilakukan instalasi *package* Ultralytics yang berisi *package-package* Python untuk membangun model menggunakan Kode 4.2.

#### Kode 4. 1 Baris Kode Instalasi *Package* Ultralytics

```
!pip install ultralytics
```

Perintah di atas mencakup pustaka penting untuk pembelajaran dan pemrosesan gambar, termasuk di antaranya PyTorch, OpenCV, dan NumPy, serta berbagai komponen NVIDIA CUDA untuk akselerasi GPU. Pengaturan ini memungkinkan pengguna untuk melatih dan menjalankan model deteksi objek dengan efisien, memanfaatkan kemampuan GPU untuk meningkatkan kinerja. Langkah selanjutnya adalah mengunduh variasi model YOLOv8 medium menggunakan baris Kode 4.2.

#### Kode 4. 2 Baris Kode Instalasi Model

```
import os
from ultralytics import YOLO
model = YOLO("yolov8m.pt")
```

Langkah selanjutnya adalah mengunduh dataset dari Roboflow untuk digunakan dengan model YOLOv8. Pertama, paket `roboflow` diinstal dengan perintah `!pip install roboflow`. Kemudian, kelas `Roboflow` diimpor dan objek `Roboflow` dibuat menggunakan kunci API yang valid. Selanjutnya, kode mengakses workspace dan proyek tertentu di Roboflow dengan ID dan nama yang diberikan. Dari proyek tersebut, versi pertama diambil, dan akhirnya, dataset diunduh dalam format yang sesuai untuk YOLOv8. Baris kode akses dataset dijalankan bersamaan dengan instalasi *package* Roboflow menggunakan baris Kode 4.4.

#### Kode 4. 3 Baris Kode Instalasi Roboflow dan Akses Dataset

```
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="030f5Tof110xZs56GRgD")
project = rf.workspace("ta-ntme6").project("project-daun-teh")
version = project.version(1)
dataset = version.download("yolov8")
```

Jika semua baris kode di atas berhasil dijalankan maka artinya model telah siap untuk melakukan proses pelatihan model.

#### 4.1.3.4 Pelatihan Model

Pada tahapan ini, model dilatih menggunakan arsitektur dan nilai parameter original YOLOv8 tanpa ada modifikasi yaitu epoch 100 dan *batch* 16. Kemudian dilakukan validasi untuk membantu mengetahui kinerja model sehingga hasil analisisnya bisa dijadikan bahan untuk pengambilan tindakan dan keputusan selanjutnya. Pelatihan dan validasi model dilakukan menggunakan Kode 4.4.

Kode 4. 4 Baris Kode Pelatihan dan Validasi Model

```
results = model.Train(data="/content/project-daun-teh-
1/data.yaml", epoch=100, batch=16)
from ultralytics import YOLO
model = YOLO("runs/detect/Train/weights/best.pt")
result = model.val()
```

Hasil dari proses validasi ini menunjukkan hasil akurasi sebesar 95% pada proses pelatihan dan validasi yang membutuhkan waktu 1 jam 54 menit. Ini merupakan hasil jika pelatihan model secara keseluruhan menggunakan nilai *default*. Dan sebagai upaya peningkatan kinerja model dan pengurangan waktu komputasi maka dilakukan pelatihan kembali sesuai dengan konfigurasi nilai pada Tabel 3.5.

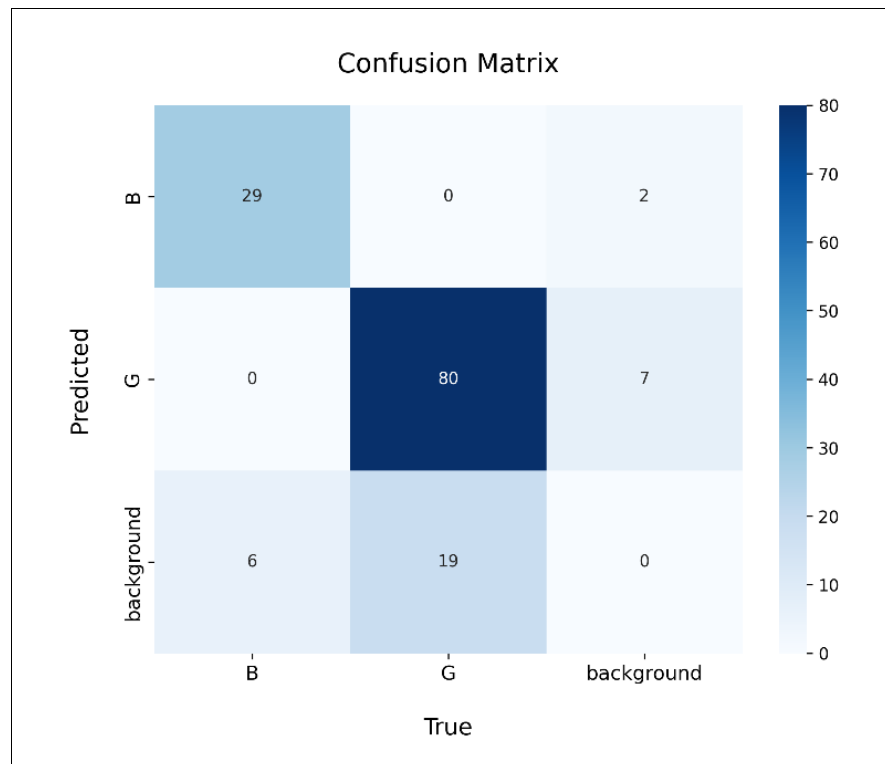
#### 4.1.3.5 Menyesuaikan Nilai *Hyperparameter*

Dari kombinasi nilai *hyperparameter* yang telah ditentukan, dilakukan pelatihan untuk mencapai nilai paling optimal. Pada Tabel 4.4 tertera hasil pelatihan dari konfigurasi parameter pada Tabel 3.5.

Tabel 4. 4 Tabel Hasil Percobaan

Konfigurasi Nilai	HASIL (%)							
	TP	FP	TN	FN	Akurasi	Presisi	Recall	F1 - Score
<i>Train 1</i>	29	2	106	6	0,94	0,94	0,83	0,88
<i>Train 2</i>	23	0	101	12	0,91	1,00	0,66	0,79
<i>Train 3</i>	29	3	107	6	0,94	0,91	0,83	0,87
<i>Train 4</i>	29	0	110	6	0,96	1,00	0,83	0,91

*Train 1* merupakan model yang dilatih menggunakan kombinasi *hyperparameter* 35 epoch, *batch size* 8, dan nilai *learning rate* sebesar 0,01. Pelatihan ini memperoleh hasil akurasi sebesar 94% setelah divalidasi menggunakan nilai *confidence* dan IoU 0,7. Karena data yang digunakan tidak seimbang, tingkat akurasi yang tinggi tidak selalu merepresentasikan kinerja model secara keseluruhan. Hasil dari *Train 1* dapat dilihat pada Gambar 4.1.

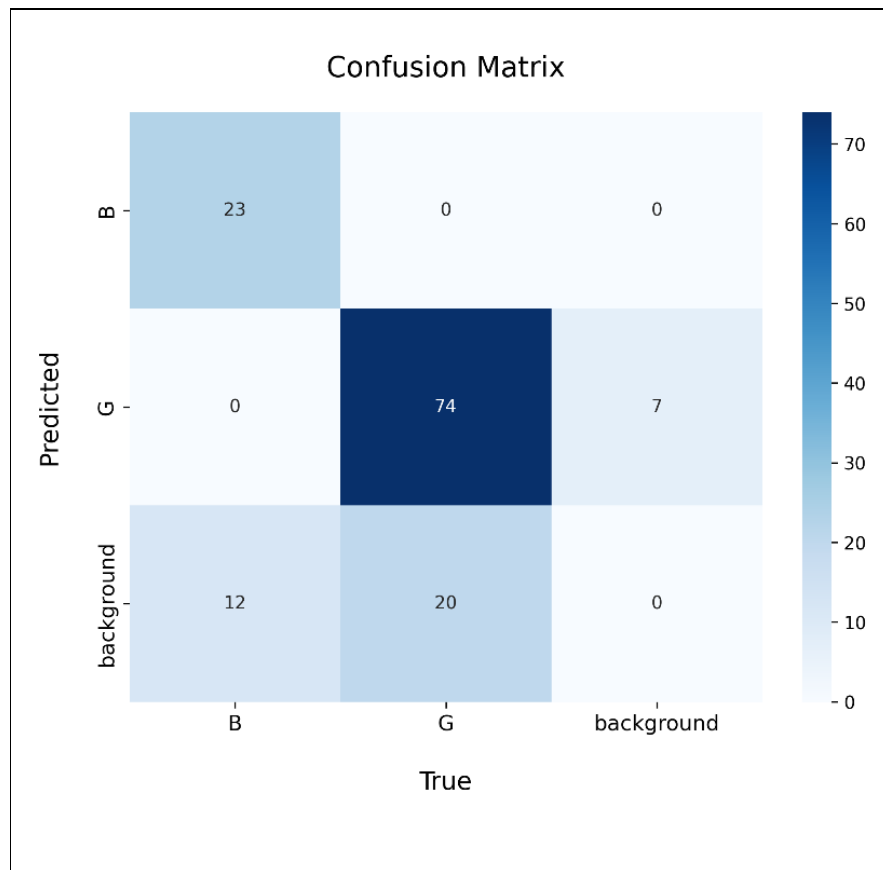


Gambar 4. 1 Hasil dari *Train 1*

Dari hasil pelatihan, hanya tercatat 2 *instance* kesalahan deteksi positif (*False positive*), yaitu kesalahan model mendeteksi *background* menjadi kelas *Brown*. Selain itu, terdapat 6 *instance* kesalahan deteksi negatif (*False negative*), artinya model melakukan kesalahan dalam mendeteksi kelas *Brown* menjadi kelas *Gray* atau *background*. Secara keseluruhan, model hanya mampu mendeteksi 83% dari objek yang ada di dataset (*Recall*). Namun, ketepatan model dalam melakukan prediksi mencapai 94% (*Precision*), yang menunjukkan hasil prediksi model untuk *instance* yang dideteksi sudah cukup baik. Skor F1 sebesar 88% mencerminkan bahwa model memiliki keseimbangan yang cukup baik antara *precision* dan *recall*, meskipun masih ada ruang untuk perbaikan. Keseimbangan antara *precision* dan

*recall* yang ditunjukkan oleh skor F1 menggambarkan bahwa model ini memiliki performa yang baik secara keseluruhan, namun tetap perlu perhatian pada peningkatan kemampuan model.

Selanjutnya pada *Train 2*, model dilatih dengan nilai *hyperparameter* yang sama hanya berbeda pada *batch size* yang digunakan yaitu 16.

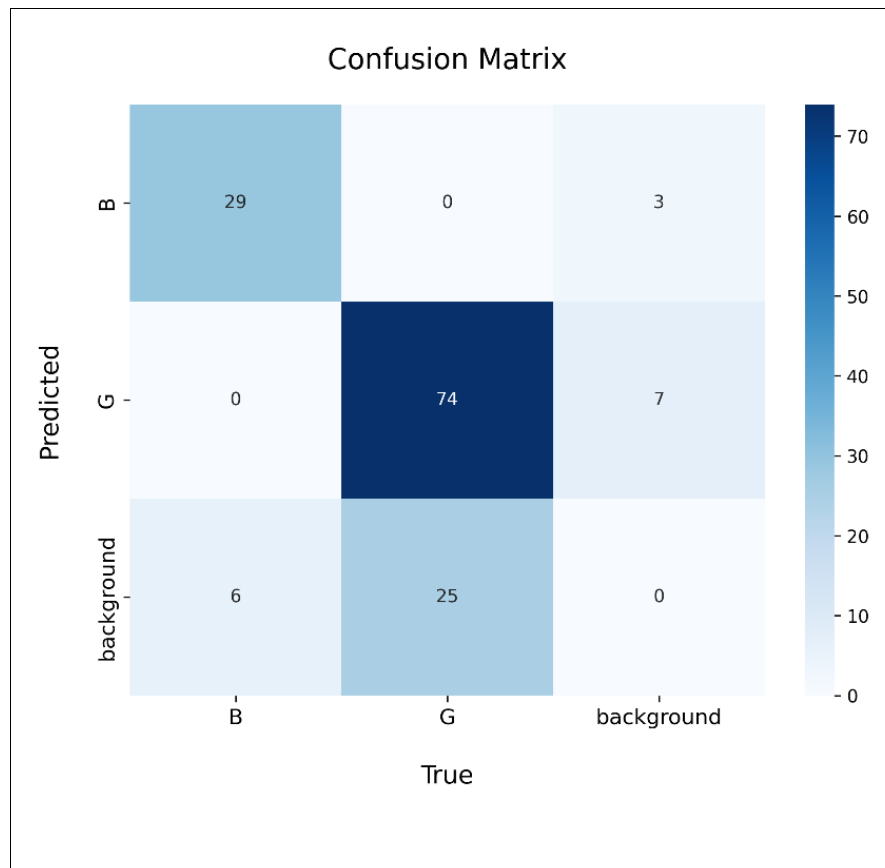


Gambar 4. 2 Hasil dari *Train 2*

Pada Gambar 4.2 dapat dilihat, tingkat akurasi yang diperoleh dari pelatihan menggunakan kombinasi ini sedikit lebih rendah dari pelatihan menggunakan *batch* 8, yaitu sebesar 91% setelah divalidasi menggunakan nilai *confidence* dan IoU 0,7. Namun dalam hal deteksi positif, model dengan konfigurasi ini memperoleh nilai *false* positif 0 yang artinya semua prediksi model sama dengan data sebenarnya dan diperoleh nilai presisi 1 atau 100%. Sementara untuk kesalahan deteksi negatif, model *Train 2* melakukan kesalahan deteksi *false* negatif paling banyak dengan total 12 *instance*, menunjukkan bahwa model gagal mendeteksi objek kelas *Brown* menjadi kelas *Gray* atau *background*. Model ini berhasil mendeteksi 66% dari

semua objek yang ada di dataset (*Recall*), dengan 100% dari prediksi tersebut adalah benar (*Precision*). Terakhir, Skor F1 sebesar 79% mencerminkan keseimbangan antara *precision* dan *recall*, meskipun masih menunjukkan bahwa performa model dapat lebih baik.

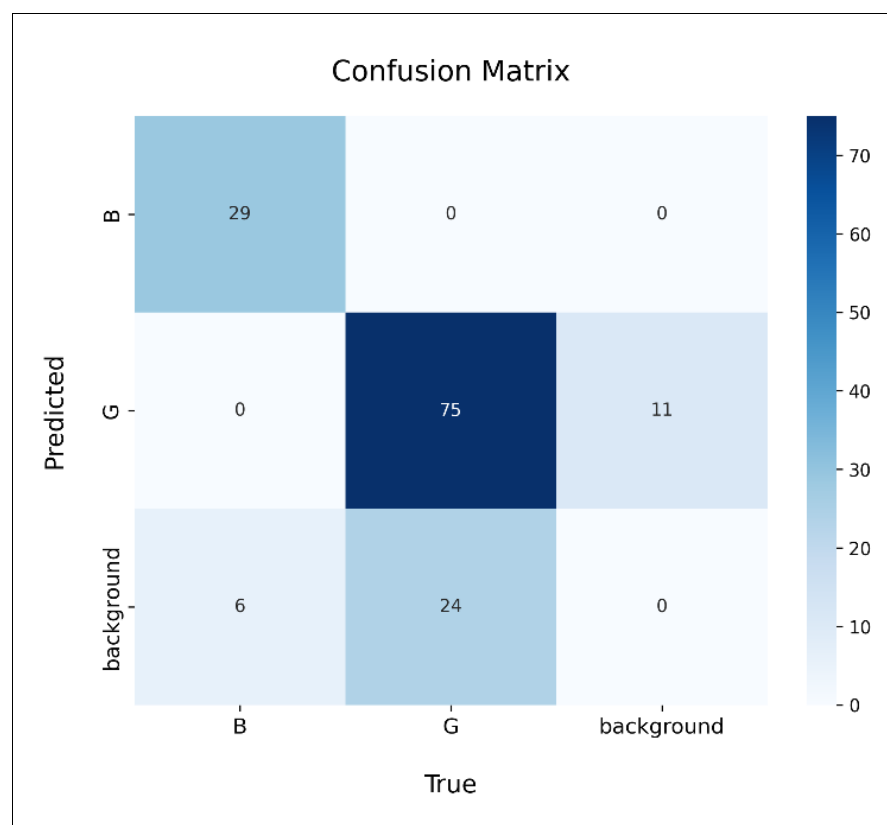
*Hyperparameter* yang digunakan pada *Train 3* adalah 35 epoch, ukuran *batch* 8, dan *learning rate* 0.001 dengan hasil *Train 3* dapat dilihat pada Gambar 4.3.



Gambar 4. 3 Hasil dari *Train 3*

Hasil pelatihan ini hampir sama dengan hasil *Train 1*, dengan hanya perbedaan satu nilai lebih tinggi pada TN dan FP. Dari hasil pelatihan, tercatat 3 *instance* kesalahan deteksi positif (*False positive*), yaitu kesalahan model mendeteksi *background* menjadi kelas *Brown*. Selain itu, terdapat 6 *instance* kesalahan deteksi negatif (*False negative*), artinya model melakukan kesalahan dalam mendeteksi kelas *Brown* menjadi kelas *Gray* atau *background*. Secara keseluruhan, model menemukan 83% objek dalam dataset (*Recall*). Namun, prediksi model memiliki ketepatan 91% (*Precision*), yang menunjukkan bahwa prediksinya cukup baik

untuk kasus yang ditemukan. Skor F1 sebesar 87% menunjukkan bahwa model memiliki keseimbangan yang cukup baik antara *precision* dan *recall*, tetapi masih ada ruang untuk perbaikan. Keseimbangan antara *precision* dan *recall* menunjukkan bahwa model bekerja dengan baik secara keseluruhan, tetapi masih ada ruang untuk meningkatkan kemampuan model. Pada *Train 4*, model dilatih dengan *hyperparameter* 35 epoch, *batch size* 16, dan *learning rate* 0,001. Setelah pelatihan, model ini mencapai akurasi sebesar 96% dengan validasi menggunakan nilai *confidence* dan IoU 0,7. Akurasi ini menunjukkan bahwa model memiliki kinerja yang baik dalam mendeteksi objek pada dataset yang digunakan.



Gambar 4. 4 Hasil dari *Train 4*

Dari hasil pelatihan, tercatat 6 *instance* kesalahan deteksi positif (*False positive*), yaitu ketika model salah mendeteksi *background* sebagai kelas *Brown*. Namun, dalam konfigurasi ini, nilai presisi yang diperoleh adalah 100% karena tidak ada kesalahan deteksi *false negatif*, yang berarti semua prediksi model adalah benar. Untuk hasil *recall*, seperti pada pelatihan sebelumnya, model berhasil mendeteksi 83% dari seluruh prediksi positif dalam dataset. Oleh karena itu, nilai



F1 yang diperoleh adalah 91%, menunjukkan bahwa keseimbangan antara *recall* dan presisi pada model ini sangat baik.

Berdasarkan data hasil pelatihan, *batch size* dan *learning rate* memiliki pengaruh signifikan terhadap hasil pelatihan model. *Batch size* yang lebih besar (16) cenderung memberikan hasil yang lebih stabil dan konvergensi yang lebih cepat, meningkatkan presisi model dengan mengurangi jumlah *false positives*. Namun, ketika menggunakan *learning rate* yang lebih tinggi (0.01), *batch size* yang lebih besar dapat mengorbankan *recall*, meningkatkan jumlah *false negatives*. Sebaliknya, dengan *learning rate* yang lebih rendah (0.001), *batch size* yang lebih besar meningkatkan baik presisi maupun *recall*, memberikan model yang lebih seimbang. *Learning rate* yang lebih tinggi (0.01) dapat mempercepat konvergensi tetapi berisiko melewati solusi optimal, menghasilkan model yang lebih ketat yang baik untuk presisi tetapi buruk untuk *recall*. *Learning rate* yang lebih rendah (0.001) memberikan hasil yang lebih stabil dan seimbang, terutama dengan *batch size* yang lebih besar. Sehingga dari semua konfigurasi nilai pada pelatihan, dipilih *Train 4* dengan konfigurasi nilai epoch 35, *batch size* 16 dan *learning rate* 0.001 karena memberikan performa terbaik dengan akurasi 96%, presisi 100%, *recall* 83%, dan F1-Score 91%.

#### 4.1.4 Pengujian Model

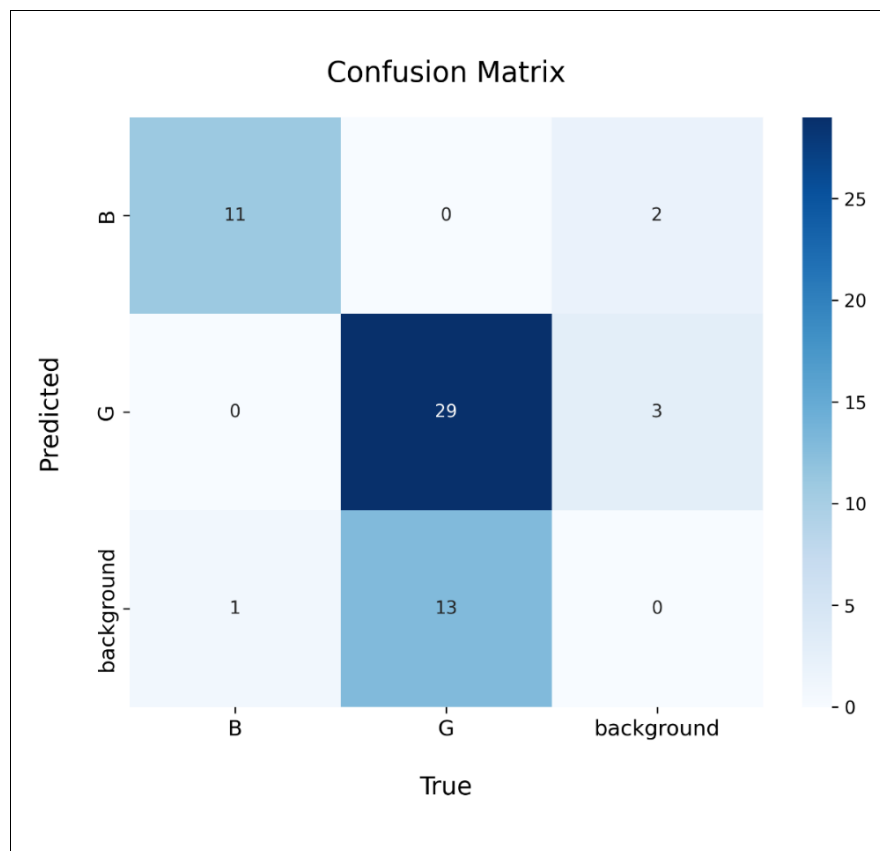
Setelah dilakukan pelatihan dan validasi pada model, maka langkah terakhir adalah pengujian akhir. Pengujian dilakukan dengan menggunakan dataset gambar daun teh berpenyakit yang belum pernah dilihat oleh model dengan tujuan Model dilatih menggunakan 28 data gambar yang terdiri dari 18 gambar penyakit *Gray Blight* dan 10 gambar penyakit *Brown Blight*. Lebih rincinya, dataset untuk pengujian memuat 28 gambar dengan total 54 *instance* yang terdiri dari 12 *instance* kelas penyakit *Brown Blight* dan 42 *instance* kelas penyakit *Gray Blight*. Untuk melakukan pengujian pada model, dijalankan Kode 4.7.

Kode 4. 5 Baris Kode Pengujian Model

```
from ultralytics import YOLO
model = YOLO("runs/detect/Train/weights/best.pt")
result = model.val(split='test')
```

#### 4.1.4. Evaluasi

Evaluasi pada penelitian ini dilakukan menggunakan matriks evaluasi *Confusion Matrix* yang termasuk perhitungan nilai akurasi, presisi, *recall* dan F1-Score dari model. Berdasarkan hasil pengujian model, diperoleh hasil matriks pada Gambar 4.1.

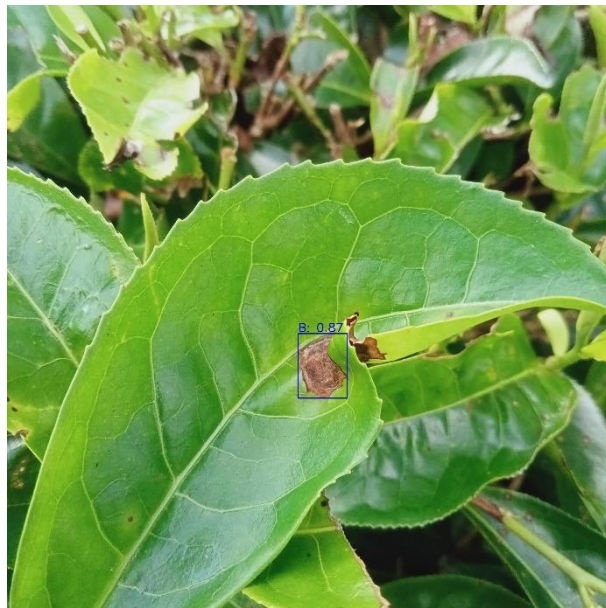


Gambar 4. 5 Hasil *Confusion Matrix*

Matriks hasil dari model yang dibangun terdiri dari matriks 3x3 yang memuat sumbu prediksi (y) dan sumbu data yang sebenarnya (x). Matriks yang dihasilkan oleh suatu pemodelan seharusnya memiliki dimensi sesuai dengan kelas pada model. Namun dalam hal ini, terdapat kelas *background* yang artinya model juga mendeteksi *background* menjadi kelas (baik kelas *Gray* atau kelas *Brown*) ataupun kelas menjadi *background* sehingga menghasilkan *matrix* berukuran 3x3.

Dari matriks hasil pengujian dapat dilihat terdapat 5 *instance background* yang terdeteksi sebagai kelas, yaitu 2 *instance background* sebagai kelas B dan 3 *instance*

sebagai kelas G. *Instance background* yang terdeteksi sebagai kelas B dihitung sebagai *False positive* (FP) karena merupakan *instance* diluar kelas B. Sedangkan *instance background* yang terdeteksi sebagai kelas G dihitung sebagai *False negative* (FN). Jika ditotal, seluruh *instance* yang berhasil dideteksi oleh model berjumlah 59. Dari hasil pengujian diperoleh nilai TP sebanyak 11 *instance* dan nilai TN sebanyak 45 *instance*. Sementara dapat dilihat pada matriks pada Gambar 4.1, nilai FN diperoleh dari kesalahan deteksi 1 *instance* kelas B menjadi *background*. Kesalah deteksi 1 *instance false negative* dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Kesalahan Hasil Deteksi (FN)

Kesalahan deteksi ini dikarenakan nilai *confidence score* yang dihasilkan untuk pada kotak pembatas tidak memenuhi syarat untuk dianggap sebagai prediksi valid yaitu 0,7. Nilai FP pada penelitian ini, terdiri dari jumlah kesalahan deteksi selain kelas B menjadi kelas B yaitu 2 *instance*. Dari data tersebut kemudian digunakan untuk melakukan perhitungan untuk mendapatkan nilai akurasi, presisi, *recall* dan F1-Score-nya. Hasil dari perhitungan matriks pada Tabel 4.6 menunjukkan kinerja dan performa model dalam melakukan tugas deteksi penyakit pada daun teh di penelitian ini.

Tabel 4. 5 Tabel Hasil Pengujian

<b>Akurasi</b>	<b>Presisi</b>	<b><i>Recall</i></b>	<b>F1-Score</b>
95%	85%	92%	88%

Berdasarkan hasil pengujian pada Tabel 4.6 diperoleh hasil akurasi sebesar 95%, *recall* sebesar 92%, presisi sebesar 85% dan nilai F1-Score sebesar 88%. Angka ini menunjukkan bahwa 95% dari hasil prediksi model adalah benar baik untuk data positif maupun data negatif. Namun nilai akurasi yang tidak sempurna 100% menunjukkan model masih melakukan kesalahan deteksi di mana 1 hasil prediksi dianggap sebagai prediksi *false positive* (yang mempengaruhi presisi) dan 2 hasil prediksi dianggap sebagai prediksi *false negative* (yang mempengaruhi *recall*).

Namun akurasi saja tidak cukup untuk menggambarkan kinerja model secara keseluruhan, sehingga dilakukan perhitungan matriks lain yaitu presisi, *recall* dan F1-Score. Hasil perhitungan nilai *recall* pada pengujian penelitian ini adalah sebesar 92%. Nilai ini menunjukkan model berhasil mendeteksi 92% dari total *instance* positif yang ada pada dataset. Selanjutnya, hasil pengujian menunjukkan hasil presisi sebesar 85% yang artinya 85% dari hasil prediksi positif yang dilakukan model adalah benar - benar positif. Terakhir dalam penelitian ini dilakukan perhitungan nilai F1-score yaitu sebesar 88%.

## 4.2 Pembahasan

Akurasi merupakan matriks yang menunjukkan kinerja model secara umum dalam melakukan sebuah tugas yang dalam penelitian ini adalah tugas identifikasi objek pada daun teh. Namun dalam tugas deteksi pada kelas yang tidak seimbang akurasi tidak selalu mencerminkan bagaimana kinerja dari model yang dibangun. Hal ini dikarenakan ada kemungkinan model bias terhadap kelas mayoritas dan mengabaikan kelas minoritas sehingga akurasi tinggi hanya berlaku pada satu kelas saja. Pada penelitian ini, dimana kelas *Brown* hanya memiliki 98 data dan kelas *Gray* memiliki 182, hasil akurasi yang diperoleh adalah sebesar 95%. Namun untuk mengetahui kinerja model dengan lebih akurat, dilakukan juga perhitungan matriks

evaluasi presisi, *recall* dan F1-Score. Hasil perhitungan matriks presisi, *recall* dan F1-Score pada penelitian ini memiliki nilai masing-masing 85%, 92% dan 88%. Hasil perhitungan yang tinggi meskipun dalam kasus kelas yang tidak seimbang menunjukkan bahwa model memiliki kemampuan yang baik dalam tugas identifikasi objek penyakit pada daun teh. Keberhasilan model ini dipengaruhi oleh beberapa faktor termasuk kualitas dan kuantitas data, nilai *hyperparameter* pelatihan serta teknik augmentasi yang digunakan.

Kualitas data memiliki pengaruh yang signifikan terhadap performa model deteksi objek seperti YOLOv8. Gambar yang buram atau tidak jelas pada data pelatihan dapat mempengaruhi model dalam mempelajari data dan menurunkan kemampuan model untuk mendeteksi dan mengenali objek dengan benar. Sebaliknya, jika data yang digunakan memiliki kejernihan dan kualitas yang bagus dan tajam membantu model dalam membedakan fitur penting dari objek yang berbeda dan meningkatkan kemampuan walaupun pada kasus data yang tidak seimbang seperti pada penelitian ini. Resolusi gambar yang rendah dapat menyebabkan hilangnya detail penting, sehingga mengurangi kemampuan model untuk mendeteksi objek kecil atau objek yang berdekatan. Sementara kuantitas data juga memiliki pengaruh besar terhadap performa model deteksi objek seperti YOLOv8. Dengan lebih banyak data, model dapat mengenali pola dan fitur yang lebih umum dan jarang, yang meningkatkan kemampuan generalisasi model. Jika dataset terlalu kecil, model tidak memiliki cukup contoh untuk belajar dengan baik. Sebaliknya, dataset yang besar dan bervariasi memungkinkan model untuk lebih memahami karakteristik yang mendasar dari objek yang ingin dideteksi jika dilatih dengan *hyperparameter* yang sesuai.

Konfigurasi nilai *hyperparameter* yang digunakan dalam membangun model juga berpengaruh terutama *hyperparameter* yang digunakan dalam penelitian ini. Nilai epoch yang lebih tinggi akan memberikan model lebih banyak kesempatan untuk mempelajari dataset. Namun untuk beberapa kasus, nilai epoch yang tinggi justru membuat model mengalami situasi *overfitting*, yaitu situasi dimana model gagal dalam mendeteksi data selain data latih (model terfokus hanya pada data latih) atau *underfitting* yaitu situasi dimana model gagal dalam menangkap pola data latih. Sehingga untuk menghindari hal ini, perlu dilakukan analisa terhadap hasil

pelatihan pada epoch yang telah ditentukan. Yang dalam penelitian ini, epoch 35 adalah nilai epoch optimal (tidak *underfitting* dan *overfitting*) karena memiliki nilai *loss* yang terus menurun dan matriks presisi *recall* yang naik pada titik optimal.

Sementara itu, pemilihan *batch size* mempengaruhi stabilitas dan kecepatan konvergensi model. *Batch size* yang lebih kecil bisa membantu model untuk generalisasi lebih baik tetapi dengan stabilitas yang lebih rendah, sedangkan *batch size* yang lebih besar cenderung memberikan pembaruan yang lebih stabil dengan risiko *overfitting*. Menggunakan *batch size* 16 dengan 35 epoch pada dataset berjumlah 588 sampel pada penelitian ini artinya model diperbarui sekitar 1295 kali selama pelatihan. Konfigurasi ini memberikan keseimbangan yang baik antara stabilitas gradien dan kemampuan generalisasi yang membantu pembaruan parameter menjadi lebih konsisten. Ketika dikonfigurasi dengan nilai *learning rate*, *batch size* pada penelitian ini memberikan pengaruh yang signifikan. Penggunaan *batch size* 16 dengan *learning rate* 0.001 memberikan hasil terbaik dalam penelitian ini, dengan akurasi 0.96, presisi 1.00, dan *recall* 0.83 (*Train* 4). *Batch size* yang lebih besar meningkatkan stabilitas gradien dan memungkinkan pembaruan parameter yang lebih efektif, sementara *learning rate* yang lebih rendah mengurangi risiko *overfitting* dan meningkatkan kemampuan generalisasi model. Sebaliknya, *batch size* 8 (*Train* 1 dan *Train* 3) juga memberikan hasil yang baik tetapi dengan sedikit perbedaan dalam presisi dan *recall* tergantung pada *learning rate* yang digunakan. *Learning rate* yang lebih tinggi (0.01) mempercepat konvergensi dan dapat memberikan hasil yang baik dengan *batch size* yang lebih kecil (8). Namun, dengan *batch size* yang lebih besar (16), *learning rate* yang lebih rendah (0.001) memberikan keseimbangan yang lebih baik antara presisi dan *recall*, meningkatkan akurasi keseluruhan model. Dengan *learning rate* 0.001 dan *batch size* 16, model mencapai kinerja terbaik dengan akurasi 0.96, presisi 1.00, dan *recall* 0.83.

Selain itu, teknik augmentasi juga berpengaruh pada hasil pemodelan. Penggunaan teknik augmentasi rotasi dan *mosaic* berguna untuk memperkaya dataset yang terbatas, sehingga model memiliki lebih banyak variasi data untuk dipelajari. Dari penelitian ini, ada beberapa aspek yang perlu diperbaiki, yaitu:

1. Kualitas dan kuantitas dataset. Untuk meningkatkan kualitas dataset, disarankan menggunakan peralatan akuisisi yang lebih baik, seperti kamera khusus fotografi, untuk menghasilkan gambar yang lebih jernih. Selain itu, pencahayaan saat pengambilan data harus diatur lebih optimal untuk mengurangi gangguan cahaya matahari yang mempengaruhi kualitas data. Menambah jumlah data yang dikumpulkan juga sangat penting untuk memberikan lebih banyak contoh bagi model untuk dipelajari. Dalam penelitian ini, dataset yang lolos proses validasi oleh ahli hanya 280.
2. Konfigurasi nilai *hyperparameter*. Konfigurasi nilai *hyperparameter* seperti *epoch*, *batch size*, dan *learning rate* memiliki peran penting dalam meningkatkan performa model. Memilih nilai *epoch* yang tepat membantu model untuk belajar secara optimal tanpa mengalami *overfitting* atau *underfitting*. Selain itu, menyesuaikan *batch size* dan *learning rate* juga sangat penting; *batch size* yang tepat mempengaruhi stabilitas dan kecepatan pelatihan, sementara *learning rate* yang sesuai memastikan pembaruan parameter yang efisien untuk mencapai konvergensi yang baik. Kombinasi yang optimal dari ketiga *hyperparameter* ini dapat secara signifikan mempengaruhi kemampuan model dalam menyelesaikan tugas dengan akurasi dan efisiensi yang tinggi.
3. Teknik Augmentasi. Teknik augmentasi yang digunakan harus lebih bervariasi untuk memperkaya dataset yang ada. Penggunaan teknik augmentasi seperti rotasi dan *mosaic* dapat membantu model mendapatkan lebih banyak variasi data untuk dipelajari, sehingga meningkatkan kemampuannya dalam mendeteksi objek. Augmentasi data sangat penting terutama ketika jumlah data yang tersedia terbatas, karena dapat mensimulasikan kondisi yang berbeda dan menambah variasi dalam dataset.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Penelitian ini berhasil menjawab dua rumusan masalah utama sehingga didapatkan kesimpulan:

1. Model YOLOv8 (*You Only Look Once version 8*) dapat digunakan untuk berbagai tugas deteksi objek termasuk identifikasi penyakit pada daun teh. Hal ini dibuktikan dengan hasil pengujian dimana model berhasil melakukan tugas identifikasi dua jenis penyakit pada daun teh.
2. Berdasarkan hasil evaluasi, model YOLOv8 yang dibangun dengan konfigurasi 35 epoch, *batch* 16, *confidence* dan IoU 0,7 serta *learning rate* 0,001 mencapai hasil yang paling baik. Untuk identifikasi penyakit pada daun teh, model menunjukkan matriks evaluasi dengan akurasi sebesar 95%, presisi sebesar 85%, *recall* sebesar 92%, dan F1-Score sebesar 88%. Presisi yang tinggi menunjukkan bahwa dari semua *instance* yang diprediksi oleh model, 85% dari prediksi tersebut adalah data sebenarnya. Sementara itu, *recall* sebesar 92% menunjukkan bahwa model berhasil mendeteksi 92% dari semua data yang sebenarnya ada, yang berarti model dapat menangkap sebagian besar *instance* pada dataset. F1-Score sebesar 88% menggambarkan keseimbangan yang baik antara presisi dan *recall*, menunjukkan bahwa model memiliki performa yang sangat baik secara keseluruhan dalam mendeteksi penyakit pada daun teh. Akurasi model YOLOv8 dalam mendeteksi objek pada daun teh sebesar 95% dipengaruhi oleh beberapa faktor. Kualitas data yang tinggi, seperti gambar yang jelas dan berkualitas baik, berkontribusi pada kemampuan model untuk mendeteksi objek dengan benar. Kuantitas data yang memadai membantu model dalam mengenali pola dan fitur yang beragam, mengurangi risiko *overfitting*. Selain itu, konfigurasi nilai *hyperparameter* yang optimal, termasuk jumlah epoch, *batch* size, dan *learning rate*, serta penerapan teknik augmentasi yang efektif, juga berperan penting dalam meningkatkan akurasi model.



## 5.2 Saran

Secara keseluruhan, studi ini berhasil mencapai tujuannya untuk mengidentifikasi objek. Meskipun menghadapi beberapa kelemahan, model yang dikembangkan menunjukkan peluang agar memiliki kinerja yang lebih baik. Oleh karena itu, terdapat beberapa saran untuk penelitian selanjutnya agar hasil yang diperoleh dapat meningkat:

1. Pengambilan Dataset dengan Lebih Jelas: Memastikan dataset yang digunakan memiliki kualitas gambar yang tinggi dan informasi anotasi yang akurat. Gambar yang jelas dan anotasi yang tepat akan membantu model untuk belajar dengan lebih baik dan menghasilkan prediksi yang lebih akurat.
2. Penambahan Jumlah Dataset yang Digunakan: Meningkatkan jumlah data yang digunakan dalam pelatihan model dapat membantu dalam meningkatkan generalisasi model terhadap berbagai situasi atau kondisi.
3. Penggunaan Alat dan Bahan Penelitian Lain yang Lebih Mendukung: Penggunaan peralatan dan bahan penelitian yang lebih canggih atau spesifik untuk tugas identifikasi objek dapat membantu meningkatkan kualitas dan akurasi pengambilan data. Termasuk penggunaan kamera dengan resolusi tinggi, perangkat lunak anotasi gambar yang canggih serta *device* yang mendukung pembangunan model menggunakan nilai *hyperparameter* yang lebih tinggi.
4. Penambahan Teknik Augmentasi: Teknik augmentasi dapat digunakan untuk memperkaya dataset dengan variasi yang lebih banyak, seperti rotasi, pergeseran, atau distorsi gambar. Hal ini membantu model untuk belajar dari berbagai sudut pandang dan situasi, sehingga meningkatkan kemampuan model untuk mengenali objek dalam berbagai kondisi.

## DAFTAR PUSTAKA

- [1] Badan Koordinasi Penanaman Modal (BPKM), “Sektor Pertanian Indonesia di Mata Dunia,” Badan Koordinasi Penanaman Modal (BPKM). Accessed: Nov. 10, 2023. [Online]. Available: <https://investindonesia.go.id/id/artikel-investasi/detail/sektor-pertanian-indonesia-di-mata-dunia>
- [2] M. A. 'Rizaty, “Produksi Teh di Indonesia Turun Menjadi 136.800 Ton pada 2022,” [dataindonesia.id](https://dataindonesia.id). Accessed: Nov. 17, 2023. [Online]. Available: <https://dataindonesia.id/agribisnis-kehutanan/detail/produksi-teh-di-indonesia-turun-menjadi-136800-ton-pada-2022>
- [3] S. 'Widi, “Berapa Konsumsi Teh per Kapita di Indonesia?,” Data Indonesia.
- [4] S. D. Hanggraeni, F.-F. ... Analisis, S. D. Hanggraeni, K. Budiraharjo, and S. I. Santoso, “ANALISIS FAKTOR-FAKTOR YANG MEMPENGARUHI LOYALITAS KONSUMEN TEH CAP BANDULAN DI KABUPATEN PEKALONGAN ANALYSIS OF FACTORS AFFECTING LOYALTY OF ‘CAP BANDULAN’ TEA CONSUMERS IN PEKALONGAN REGENCY,” 2021.
- [5] “PT Perkebunan Nusantara IV,” [ptpn4.co.id](https://www.ptpn4.co.id/). Accessed: Nov. 17, 2023. [Online]. Available: <https://www.ptpn4.co.id/>
- [6] S. K. 'Wildah and A. 'Latif, “Kombinasi Metode Fitur Ekstraksi untuk Identifikasi Penyakit pada Daun Teh,” *Jurnal Sistem dan Teknologi Informasi*, vol. 11, 2023.
- [7] A. B. Saputra, K. Kurniah, R. Risma, and A. N. Hidayah, “Deteksi pestisida Deltamethrin pada daun teh dengan variasi semprot (3x dan 6x) menggunakan spektroskopi raman,” *Kultivasi*, vol. 19, no. 1, p. 1047, Mar. 2020, doi: 10.24198/kultivasi.v19i1.24898.
- [8] 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 Di Taman Nasional Sebangau,” *Jurnal Informatika*, vol. 10, no. 1, pp. 19–27, Mar. 2023, doi: 10.31294/inf.v10i1.13922.

- [9] R. Sapkota *et al.*, “Comprehensive Performance Evaluation of YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments,” vol. 2024, doi: 10.48550/arXiv.2407.12040.
- [10] M. Althaf, P. Perkasa, R. Reza El Akbar, M. Al Husaini, and R. Rizal, “VISUAL ENTITY OBJECT DETECTION SYSTEM IN SOCCER MATCHES BASED ON VARIOUS YOLO ARCHITECTURE,” vol. 5, no. 3, pp. 811–820, 2015, doi: 10.52436/1.jutif.2024.5.3.2015.
- [11] A. Sundaresan Geetha, M. A. R. Alif, M. Hussain, and P. Allen, “Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection: Performance Metrics and Model Efficacy,” *Vehicles*, vol. 6, no. 3, pp. 1364–1382, Aug. 2024, doi: 10.3390/vehicles6030065.
- [12] A. Prakash, “A Comparative Analysis of Object Identification Labelling Platforms: Basketball Perspective,” *Int J Med Net*, vol. 2, p. 2024.
- [13] Q. Lin, G. Ye, J. Wang, and H. Liu, “RoboFlow: a Data-centric Workflow Management System for Developing AI-enhanced Robots.” [Online]. Available: <https://sites.google.com/u.northwestern.edu/roboflow>
- [14] M. J. A. Soeb *et al.*, “Tea leaf disease detection and identification based on YOLOv7 (YOLO-T),” *Sci Rep*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-33270-4.
- [15] R. Sembiring and W. Tarigan, “Pelayanan Pekabar Injil Nederlandsche Zendeling-Genootschap (NZG) Untuk Meningkatkan Pertanian di Tanah Karo,” *Jurnal Pendidikan Agama dan Teologi*, vol. 3, no. 9, pp. 1–8, 2023, doi: 10.56393/intheos.v3i9.1743.
- [16] Y. Wan, L. Zou, L. Zeng, H. Tong, and Y. Chen, “A new colletotrichum species associated with brown blight disease on camellia sinensis,” *Plant Dis*, vol. 105, no. 5, May 2021, doi: 10.1094/PDIS-09-20-1912-RE.
- [17] S. H. Lee, S. R. Lin, and S. F. Chen, “Identification of tea foliar diseases and pest damage under practical field conditions using a convolutional neural network,” *Plant Pathol*, vol. 69, no. 9, pp. 1731–1739, Dec. 2020, doi: 10.1111/ppa.13251.
- [18] A. Rahman Ramli and A. Khalil Gibran, “Analisis Kinerja Algoritma Machine Learning Untuk Deteksi Penyakit Daun Teh Dengan Particle

- Swarm Optimization.” [Online]. Available: <https://www.kaggle.com/code/arafathhmm/tea-leaf->
- [19] J. A. Pandian, S. N. Nisha, K. Kanchanadevi, A. K. Pandey, and S. K. Rima, “Grey Blight Disease Detection on Tea Leaves Using Improved Deep Convolutional Neural Network,” *Comput Intell Neurosci*, vol. 2023, pp. 1–11, Jan. 2023, doi: 10.1155/2023/7876302.
  - [20] “Shasi,” R. Rastogi, R. R. Maaliw, and aSHOK Kumar, *Theory And Applications In Deep Learning*. Xoffencer International Publication, 2023.
  - [21] A. Ramadhanu and H. Syahputra, “PENGENALAN TEKNOLOGI PENGOLAHAN CITRA DIGITAL (DIGITAL IMAGE PROCESSING) UNTUK SANTRI DI RAHMATAN LIL’ALAMIN INTERNATIONAL ISLAMIC BOARDING SCHOOL,” *Communnity Development Journal*, vol. 3, no. 2, pp. 1239–1244, 2022.
  - [22] S. Tasnim and W. Qi, “Progress in Object Detection: An In-Depth Analysis of Methods and Use Cases,” *European Journal of Electrical Engineering and Computer Science*, vol. 7, no. 4, pp. 39–45, Jul. 2023, doi: 10.24018/ejece.2023.7.4.537.
  - [23] J. Terven and D. Cordova-Esparza, “A Comprehensive Review of YOLO: From YOLOv1 and Beyond,” Apr. 2023, [Online]. Available: <http://arxiv.org/abs/2304.00501>
  - [24] M. Anwar, Y. Kristian, and E. Setyati, “CLASSIFICATION OF CHILI PLANT DISEASES EQUIPPED WITH LEAF AND FRUIT IMAGE SEGMENTATION USING YOLO V7,” *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 6, no. 1, 2023.
  - [25] “ALGORITHM PRINCIPLES AND IMPLEMENTATION WITH YOLOV8,” [mmyolo.readthedocs.io](https://mmyolo.readthedocs.io).
  - [26] M. Qiu, L. Huang, and B. H. Tang, “ASFF-YOLOv5: Multielement Detection Method for Road Traffic in UAV Images Based on Multiscale Feature Fusion,” *Remote Sens (Basel)*, vol. 14, no. 14, Jul. 2022, doi: 10.3390/rs14143498.
  - [27] tarandeep singh, “Hyperparameter tuning,” [geeksforgeeks](https://www.geeksforgeeks.org/hyperparameter-tuning/).

- [28] Reyvan Maulid, “Konsep Hyperparameter Tuning pada Machine Learning,” DQLab. Accessed: Jun. 18, 2024. [Online]. Available: <https://www.trivusi.web.id/2022/08/epoch-dan-batch-size.html>
- [29] Trivusi, “Apa Bedanya Epoch dan Batch Size pada Deep Learning?” Accessed: Jun. 18, 2024. [Online]. Available: <https://www.trivusi.web.id/2022/08/epoch-dan-batch-size.html>
- [30] G. 'Boesch, “What is Intersection over Union (IoU)?,” viso.ai.
- [31] G. 'Munawar, R. 'Excel, A. 'Jocher, “Ultralytics YOLO Docs,” ultralytics.
- [32] A. 'Ramdani, T. 'Retnowardhani, “Apakah Deep Learning ?,” mmsi.binus.ac.id/. Accessed: Jul. 03, 2024. [Online]. Available: <https://mmsi.binus.ac.id/2019/11/26/apakah-deep-learning/>
- [33] “Roboflow,” Roboflow, Computer Application Company. Accessed: Nov. 13, 2023. [Online]. Available: <https://roboflow.com/>
- [34] “Vibhutijain99,” “Training VS Testing VS Validation,” geeksforgeeks.org. Accessed: Nov. 17, 2023. [Online]. Available: <https://www.geeksforgeeks.org/training-vs-testing-vs-validation-sets/>

## LAMPIRAN

### 1. Hasil Wawancara

Wawancara dilakukan dengan bapak Purwanto selaku Kepala Bagian yang bertanggung jawab atas perkebunan teh di Unit Bah-Butong pada tanggal 8 Januari 2024. Berikut merupakan pertanyaan yang peneliti ajukan kepada narasumber sebagai acuan melakukan penelitian.

No.	Pertanyaan	Jawaban
1.	Apakah pemberian pestisida dilakukan ke semua tanaman termasuk yang tidak terkena penyakit (tanaman yang sehat)?	Pestisida hanya diberikan kepada daun teh yang terkena penyakit. Pemberian pestisida dilakukan dengan mengisolasi pada daun teh yang terkena penyakit.
2.	Apakah pemberian pestisida dilakukan sesuai dengan jenis penyakit yang ada atau secara <i>general</i> ?	Tidak, pemberian pestisida dilakukan sesuai jenis penyakit yang menyerang. Untuk jenis penyakit yang berbeda, diberikan jenis pestisida yang berbeda pula.
3.	Penyakit yang sering muncul apa saja?	Untuk jenis penyakit yang paling sering muncul adalah <i>Helopltis</i> , <i>Brown Blight</i> , <i>Gray Blight</i> , <i>Red Spot</i> serta <i>Blister Blight</i>
4.	Kapan penyakit-penyakit daun teh sering muncul?	Untuk jenis penyakit berbeda waktu kemunculannya tergantung pada cuaca yang musim yang sedang terjadi. Yang sering muncul pada musim hujan adalah <i>Blister Blight</i> .
5.	Apa penyebab utama kemunculan penyakit pada daun teh	Penyebab kemunculan penyakit daun teh beragam, ada yang muncul disebabkan oleh cuaca, jamur dan hama.
6.	Apa tindakan yang dilakukan jika penyakit sudah mulai menyebar?	Dilakukan isolasi pada tanaman yang terkena penyakit dengan memberi jarak dan segera menangani penyakit yang ada.
7.	Tindakan apa yang dilakukan jika penyakit masih bermunculan setelah diberi pestisida menyeluruh?	Pada saat setelah panen, dilakukan pemberian pestisida secara menyeluruh untuk mencegah tanaman teh terserang penyakit. Namun jika masih ada tanaman teh yang terkena penyakit maka akan dilakukan isolasi pada tanaman yang

		terserang dan diberi pestisida sesuai jenis penyakitnya.
8.	Bagaimana pihak perkebunan melakukan identifikasi penyakit pada daun tehnya?	Identifikasi penyakit dilakukan dengan mengenali jenis penyakit secara langsung pada daun teh tanpa menggunakan bantuan teknologi apa pun. Hanya berdasarkan pengamatan mata dan pengetahuan pihak perkebunan yang bekerja.
9.	Apakah tantangan terbesar yang dialami pihak pengelola perkebunan saat melakukan identifikasi penyakit?	Kesulitan identifikasi pada penyakit yang memiliki kemiripan seperti <i>Gray Blight</i> dan <i>Brown Blight</i> , apalagi jika keadaan matahari sedang cerah maka penyakit <i>Brown Blight</i> akan sekilas terlihat seperti <i>Gray Blight</i> .
10.	Pernahkah terjadi kesalahan identifikasi? Dan apa akibatnya?	Pernah. Salah identifikasi penyakit pernah terjadi sekitar 5 kali di mana hal ini mengakibatkan jenis pestisida yang diberikan juga salah dan malah merusak tanaman serta merugikan pihak perkebunan.

## 2. Bukti Dokumentasi Wawancara

Wawancara dilakukan dengan bapak Purwanto selaku Kepala Bagian yang bertanggung jawab atas perkebunan teh di Unit Bah-Butong.





### 3. Dataset Penelitian

Dataset penelitian terdiri dari 600 data penyakit pada daun teh yang diambil langsung di PTPN IV Unit Bah-Butong. Data ini disimpan dalam 2 folder terpisah yaitu folder *Gray Blight* untuk 300 gambar penyakit *Gray*, dan folder *Brown Blight* untuk 300 gambar penyakit *Brown*. Namun setelah validasi, hanya terdapat 280 data (182 *Gray* dan 98 *Brown*) yang layak untuk digunakan dalam penelitian dan tersimpan pada folder *Gray Blight* Edit (untuk penyakit *Gray*) dan folder *Brown Blight* Edit (untuk penyakit *Brown*). Sementara folder Mix 1 dan Mix 2 berisi data gambar yang blur, tidak jelas, gambar dengan penyakit di luar *Brown* dan *Gray* sehingga tidak layak digunakan untuk membangun model.

Dataset pada penelitian ini dapat diakses pada link:

<https://drive.google.com/drive/folders/1q7JrDBxGD00j9idQVHCvX6QllpCuaNwA?usp=sharing>

Sampel penyakit yang diidentifikasi:

<i><b>Brown Blight</b></i>	<i><b>Gray Blight</b></i>
	

### 4. Surat pernyataan Validasi

Dataset yang digunakan selama pelatihan telah divalidasi oleh ahli penyakit tanaman yaitu Ibu Dr. Danarsi Diptaningsari, S.P., M Si., seorang peneliti di Organisasi Riset Pertanian dan Pangan BRIN Cibinong Science Center, Jl. Raya Jakarta-Bogor KM 46, Cibinong, Bogor, 16911.





**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
INSTITUT TEKNOLOGI SUMATERA**

Jalan Terusan Ryacudu, Way Hui, Jati Agung, Lampung Selatan 35365  
Telepon (0721) 8030188, Fax. (0721) 8030189, Email: pusat@itera.ac.id  
www.itera.ac.id

Yang bertanda tangan di bawah ini,

Nama : Dr. Danarsi Diptaningsari, S.P., M. Si.

Profesi : Peneliti



Instansi : Badan Riset dan Inovasi Nasional (BRIN) Cibinong, Bogor

Dengan ini menyatakan bahwa data yang digunakan dalam penelitian mahasiswa atas nama Wella Amanda dengan NIM 120140057 dengan judul:

Judul Tugas Akhir : Identifikasi Penyakit pada Daun Teh Menggunakan Metode *You Only Look Once Versi 8* (YOLOv8)

Telah dilakukan validasi terhadap data yang digunakan oleh saya selaku ahli dalam bidang Penyakit Tanaman. Berikut rincian data yang telah divalidasi :

1. Jenis data : Data berupa gambar penyakit *Brown Blight* dan *Gray Blight* pada daun teh dengan ekstensi \*.jpg.
2. Sumber data : Data diambil langsung dari Perseroan Terbatas Perkebunan Teh Nusantara IV (PTPN IV) Unit Bah-Butong di Sumatera Utara
3. Periode Pengumpulan : Data diambil sekitar bulan Desember akhir – Januari awal 2024

No.	Gambar Penyakit pada Daun Teh	Jenis Penyakit	Menurut Pakar Ahli
1.		<i>Brown Blight</i>	Sesuai
2.		<i>Gray Blight</i>	Sesuai

Deskripsi Validasi :

Saya telah melakukan pengecekan terhadap keakuratan, kesesuaian dan keandalan data yang telah dikumpulkan dalam penelitian ini. Proses validasi menunjukkan hasil bahwa data tersebut sudah valid dan layak digunakan untuk penelitian.

Demikian surat pernyataan ini dibuat dengan yang sebenar benarnya untuk dapat digunakan sebagaimana mestinya.

Bogor, 20 Juli 2024

Peneliti



Dr. Danarsi Diptaningsari, S.P., M. Si.



Dokumen ini ditandatangani secara elektronik menggunakan sertifikat dari BSI-E, silahkan lakukan verifikasi pada dokumen elektronik yang dapat diunduh dengan melakukan scan QR Code