

BAB III

ANALISA DAN PERANCANGAN

3.1 Dataset

Dataset yang digunakan dalam penelitian ini merupakan citra buah sawit dari lingkungan PT Herfinta. Dataset tersebut digunakan untuk melatih model *YOLOv8* agar dapat mendeteksi dan mengklasifikasikan tingkat kematangan buah sawit. Dataset ini terdiri dari gambar buah sawit yang diambil dari berbagai sudut pandang dan kondisi pencahayaan yang berbeda untuk meningkatkan akurasi model.

Dataset dibagi menjadi tiga kelompok utama, yaitu data train, data validasi, dan data uji. Pembagian data dilakukan agar model dapat dilatih secara optimal dan diuji performanya dengan data yang belum pernah dilihat sebelumnya. Berikut adalah rincian pembagian dataset:

Tabel 3.1 Dataset

Kategori Data	Jumlah Citra	Persentase
Data Train	1.228	70%
Data Validasi	348	20%
Data Test	165	10%
Total	1.742	100%

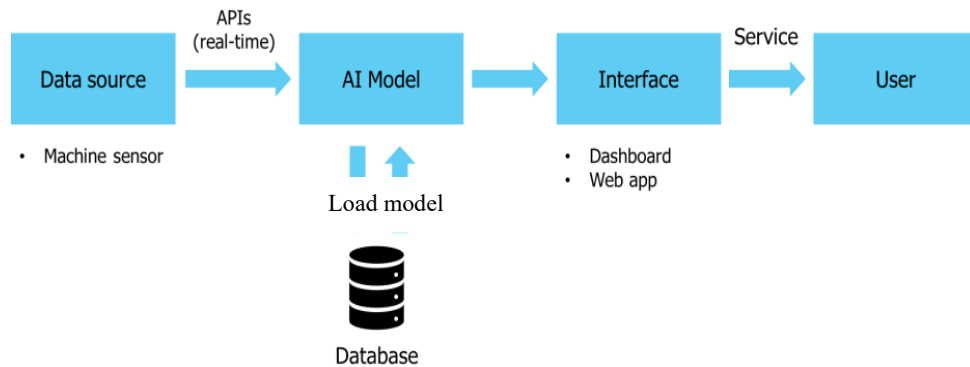
1. Data Train Sebanyak 1.228 citra digunakan untuk melatih model YOLOv8. Data ini digunakan agar model dapat mempelajari fitur penting dari setiap tingkat kematangan buah sawit.
2. Data Validasi Sebanyak 348 citra digunakan untuk memvalidasi hasil training pada setiap epoch. Data ini membantu memantau performa model dan mencegah terjadinya overfitting.
3. Data Test Sebanyak 165 citra digunakan untuk menguji performa akhir model. Data ini tidak digunakan selama proses pelatihan, sehingga hasil uji dapat memberikan gambaran objektif mengenai kemampuan generalisasi model.

3.2 Pemodelan Sistem

Pemodelan sistem merupakan proses perancangan dan penyusunan komponen-komponen sistem secara terstruktur untuk merepresentasikan alur kerja, fungsi, serta interaksi antar bagian dalam mencapai tujuan tertentu. Dalam konteks ini, pemodelan sistem YOLOv8 digunakan untuk membangun sebuah sistem deteksi otomatis yang mampu mengidentifikasi tingkat kematangan buah sawit melalui citra visual.

Pemodelan sistem dilakukan untuk menggambarkan alur kerja dari sistem deteksi kematangan buah sawit berbasis *YOLOv8*. Proses pemodelan sistem ini melibatkan beberapa tahapan utama, mulai dari pengumpulan data hingga

implementasi model pada sistem berbasis web. Ilustrasi bisa di lihat pada Gambar 3.1.



Gambar 3.1 Pemodelan Sistem

Dalam Gambar 3.1 dari permodelan sistem, terdapat 5 tahapan proses yang dijalankan oleh sistem yang dibangun untuk mendeteksi buah kelapa sawit. Berikut adalah tahapan-tahapan tersebut:

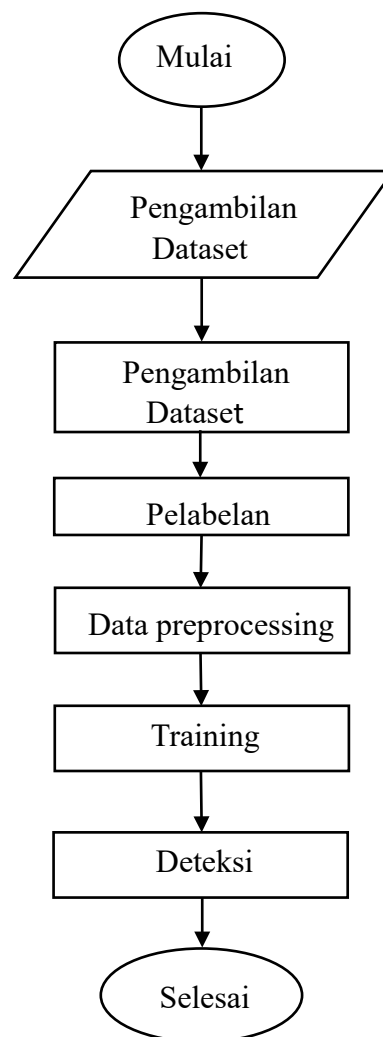
1. Langkah pertama yang dilakukan adalah memilih file data berupa Gambar pada direktori.
2. Langkah kedua adalah proses deteksi buah sawit dengan menggunakan algoritma YOLOv8
3. Langkah ketiga adalah load model hasil training yang berisi bobot dan bias dataset, dan dilakukan proses pencocokan dari data uji dengan model sehingga menghasilkan kelas objek yang dikenali berdasarkan dataset.
4. Langkah keempat adalah sistem menampilkan hasil deteksi objek berupa bounding box, dan label pada data uji.

5. Langkah kelima keluaran sistem menampilkan hasil deteksi berupa kematangan buah sawit.

3.3 Deteksi Objek Buah Kelapa Sawit dengan *Yolov8*

Flowchart merupakan gambaran proses berjalannya alir dari system pendeteksian kematangan pada buah sawit dengan menggunakan algoritma *YoloV8*.

Flowchart dari sistem pendeteksian menggunakan algoritma *YoloV8*.



Gambar 3.2 Perancangan sistem

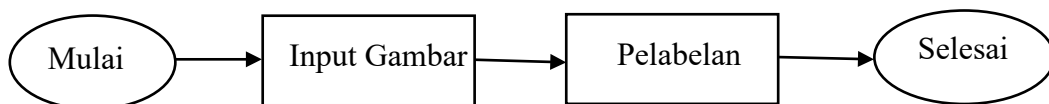
3.3.1 Pengambilan Dataset



Gambar 3.3 Proses Pengambilan Dataset

Proses pengambilan dataset di ambil dalam bentuk foto dengan menggunakan kamera *Smartphone* atau *Laptop* pada buah sawit yang akan di deteksi. Dataset digunakan sebagai sampel yang akan digunakan untuk melatih sistem pendeteksian yang akan dibangun.

3.3.2 Pelabelan Dataset

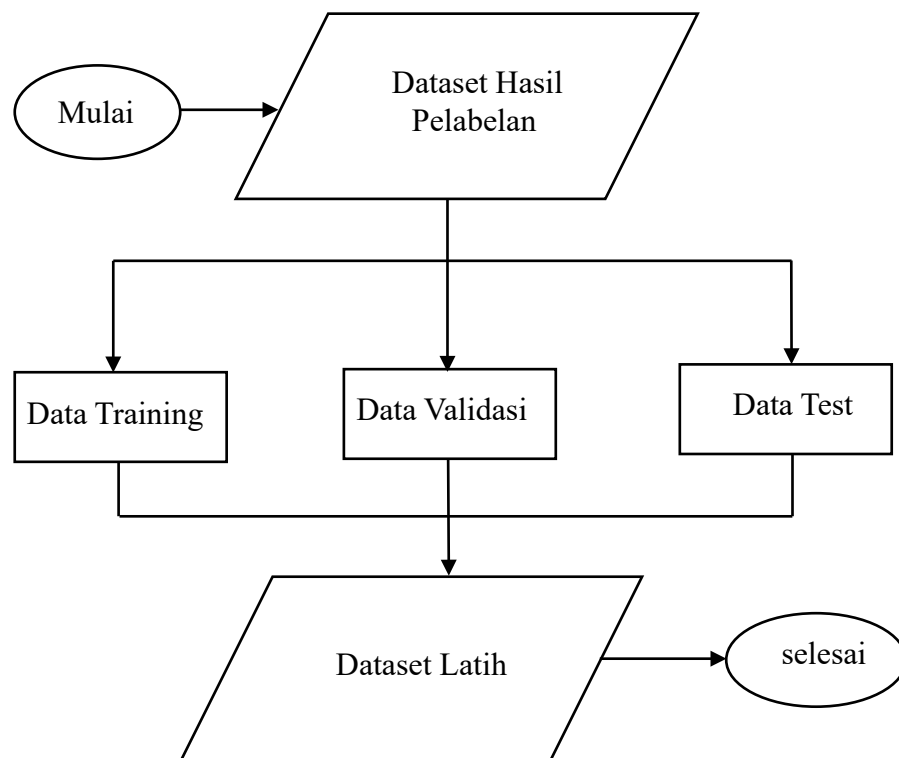


Gambar 3.4 Proses Pelabelan Dataset

Pada proses ini dataset yang telah melalui proses pengolahan gambar akan di beri label menggunakan roboflow. Roboflow bertujuan untuk memberi bounding box pada setiap objek yang terdapat pada gambar yang akan digunakan untuk melatih model.

3.3.3 Data preprocessing

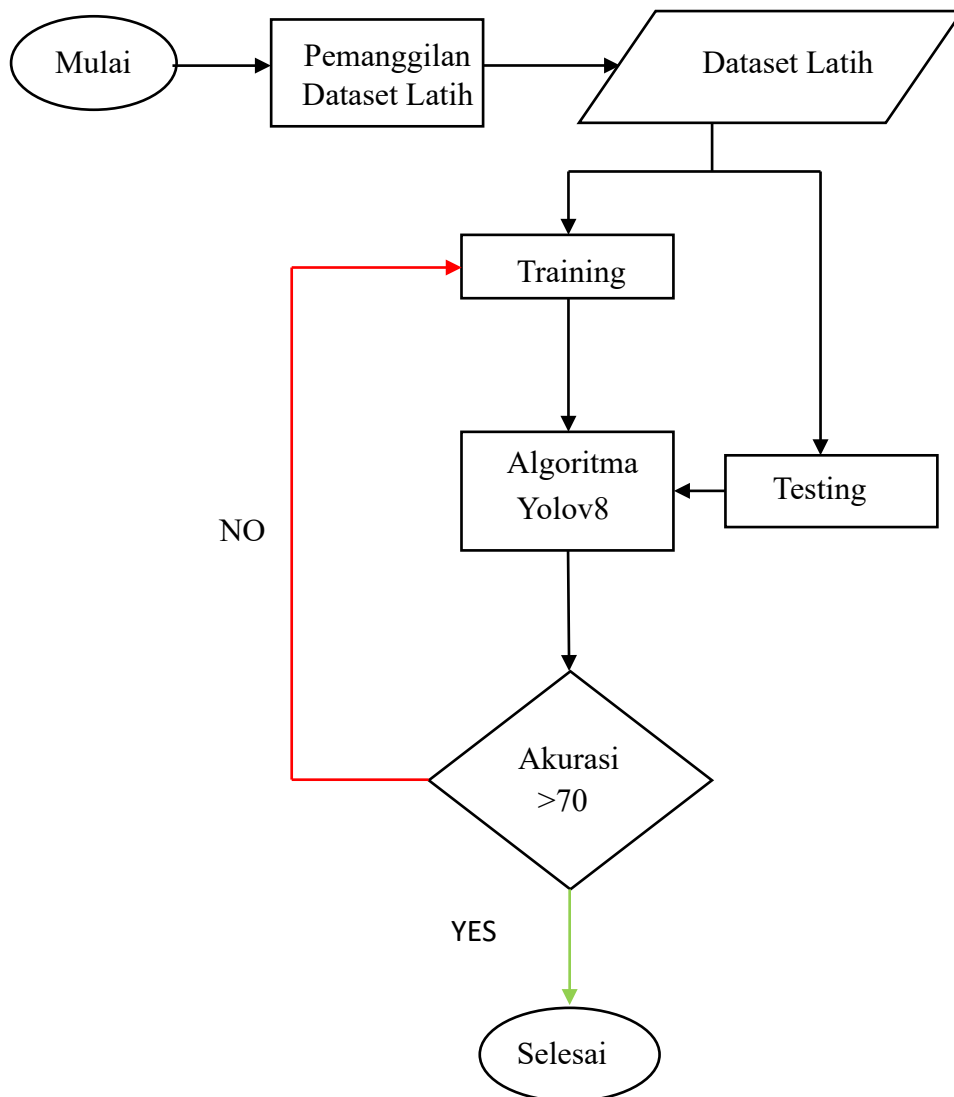
Dataset yang telah dilabeli akan di bagian menjadi 3 yaitu data testing dan data training dan data validasi. Pembagian data training digunakan untuk melatih model untuk mendeteksi. Data validasi digunakan untuk mengevaluasi model dalam mendeteksi dan data test sebagai bahan test untuk menguji hasil performa dari model dan mendapatkan akurasi dengan data yang baru saja dilihat.



Gambar 3.5 Proses Pembagian Dataset

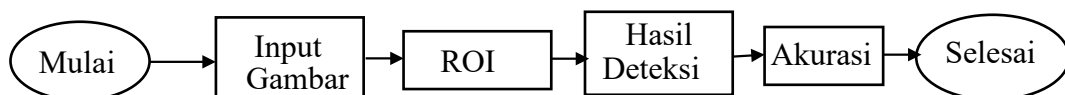
3.3.4 Training Dataset

Pada proses ini data yang sudah dibagi menjadi data training dan data testing akan di panggil. Dimana data yang berbentuk API yang berada pada roboflow akan di panggil melalui google collab. Selanjutnya pengujian data model.



Gambar 3.6 Proses Training & Testing

3.3.5 Deteksi



Gambar 3.7 Proses Deteksi

Proses pendeteksian dimulai dengan menginput gambar yang telah dikumpulkan, selanjutnya pemanggilan gambar untuk menguji sistem dengan menggunakan library pada *YoloV8*. Selanjutnya pengujian data model menggunakan algoritma *YoloV8* dengan gambar yang telah diberi bounding sebagai label, setelah melakukan pengujian akan menghasilkan akurasi dan *ROI* sebagai batasan area deteksi.

3.4 Pengujian Sistem

Teknik pengujian sistem pada penelitian ini menggunakan *confusional matrix*. *confusional matrix* digunakan dalam membandingkan data yang sebenarnya dan data prediksi. *confusional matrix* memiliki 2 kondisi yaitu Positif dan Negative dengan 4 kombinasi *true* positif (TP), *true* negative (TN), *false* positif (FP), dan *false* negative (FN). Hal ini memungkinkan *confusional matrix* memberikan akurasi dari prediksi. Tahapan yang digunakan untuk menguji sistem dengan menggunakan *confusional matrix*.

1. Data Training

Pada tahap ini dataset mentah yang telah diolah akan digunakan untuk melatih sistem dengan menggunakan algoritma *YoloV8*. *YoloV8* dilatih untuk mendapatkan akurasi tinggi.

2. Data Testing

Dengan menggunakan metode *YoloV8*, data testing di gunakan untuk melihat performa sistem seberapa akurat dalam mendeteksi.

3.4.1 Confusion Matrix

Confusion Matrix merupakan metode evaluasi kinerja model deteksi objek yang digunakan untuk menghitung akurasi prediksi model *Confusion Matrix* adalah alat evaluasi yang digunakan dalam klasifikasi untuk menilai kinerja model dengan membandingkan hasil prediksi dengan data yang sebenarnya. Matriks ini berbentuk tabel yang menunjukkan jumlah prediksi yang benar dan salah dalam setiap kelas.. Matrix ini berbentuk tabel yang memperlihatkan perbandingan antara hasil prediksi dengan data aktual. *Confusion Matrix* akan membantu mengukur seberapa baik model dalam mengklasifikasikan buah sawit ke dalam dua kategori Matang, dan Mentah.

Confusion Matrix terdiri dari empat komponen utama:

- *True Positive* (TP): Prediksi benar dan sesuai dengan kelas aktual (buah sawit matang terdeteksi sebagai matang).
- *True Negative* (TN): Prediksi salah yang benar-benar tidak sesuai (tidak ada buah sawit terdeteksi, dan memang tidak ada).
- *False Positive* (FP): Prediksi salah, di mana model mendeteksi kelas yang salah (buah sawit mentah terdeteksi sebagai matang).
- *False Negative* (FN): Prediksi salah, di mana model gagal mendeteksi objek yang sebenarnya ada (buah sawit matang tidak terdeteksi).

Confusion Matrix dapat digunakan untuk menghitung beberapa metrik evaluasi model, antara lain:

1. Mengukur seberapa sering model melakukan prediksi yang benar.

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

2. Mengukur ketepatan prediksi positif dari model, atau seberapa banyak prediksi positif yang benar.

$$\text{Precision} = \frac{TP}{TP+FP}$$

3. Mengukur seberapa baik model mendeteksi semua kasus positif.

$$\text{Recall} = \frac{TP}{TP+FN}$$

4. Merupakan harmonic mean dari Precision dan Recall, digunakan saat ada ketidakseimbangan data.

$$\text{F1-Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Untuk mengevaluasi model *YOLOv8*, *Confusion Matrix* dihitung berdasarkan hasil prediksi terhadap data uji. Dari hasil matriks, dapat diketahui seberapa sering model melakukan kesalahan klasifikasi dan dalam kategori apa kesalahan tersebut terjadi.

Sebagai contoh, misalkan sistem deteksi kematangan buah sawit diuji pada 100 gambar dan menghasilkan *Confusion Matrix* berikut:

Tabel 3.2 Contoh perhitungan *Confusion Matrix*

Kategori	Prediksi Matang	Prediksi Mentah
Sebenarnya Matang	40 (TP)	10 (FN)
Sebenarnya tidak Matang	15 (FP)	35 (TN)

Dari tabel di atas, perhitungan metrik evaluasi adalah sebagai berikut:

- $\text{Accuracy} = \frac{40+35}{40+35+15+10} = \frac{75}{100} = 0.75 = 75\%$
- $\text{Precision} = \frac{40}{40+15} = \frac{40}{55} = 0,727 = 72,7\%$
- $\text{Recall} = \frac{40}{40+10} = \frac{40}{50} = 0,8 = 80\%$
- $\text{F1 Score} = 2 \times \frac{0,727 \times 0,8}{0,727 + 0,8} = 2 \times \frac{0,5816}{1,527} = 0,76 = 76\%$
- $\text{Specificity} = \frac{35}{35+15} = 0,7 = 70\%$

Berdasarkan hasil perhitungan di atas, model memiliki akurasi sebesar 75%, dengan recall yang cukup tinggi (80%), menunjukkan bahwa model cukup

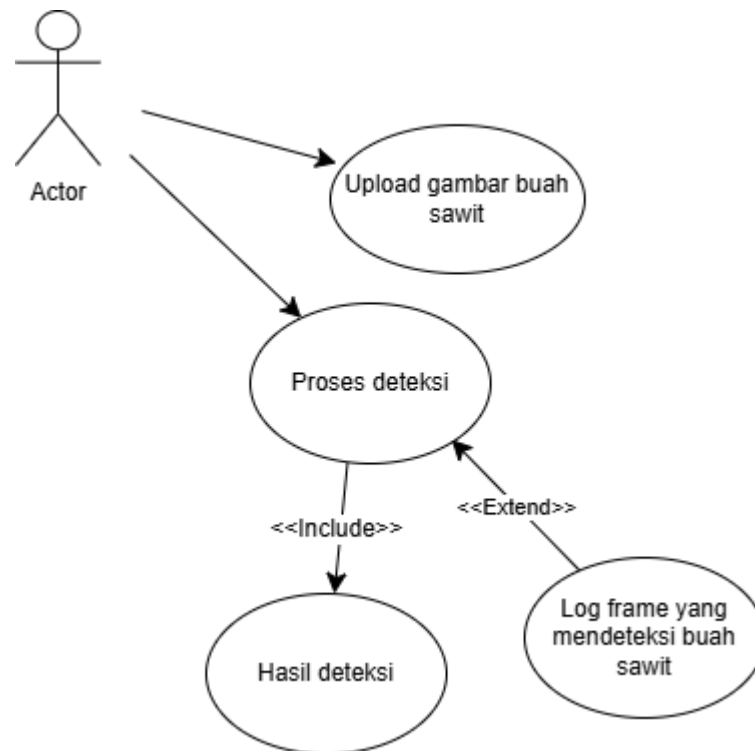
baik dalam mendeteksi buah yang benar-benar matang. Namun, spesifisitas yang lebih rendah (70%) menunjukkan bahwa model masih sering salah mendeteksi buah yang belum matang sebagai matang (False Positive masih cukup tinggi). Dengan memahami Confusion Matrix dan perhitungan metrik ini, sistem deteksi kematangan buah sawit berbasis *YOLOv8* dapat lebih dioptimalkan untuk hasil yang lebih akurat dan efisien.

3.5 *Unified Modeling Language*

Dalam perancangan sistem ini, akan dipertimbangkan aliran kerja sistem yang akan dikembangkan, serta rancangan aktor-aktor dan proses-proses yang akan berinteraksi dalam aplikasi tersebut dengan memanfaatkan *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *class diagram* dan *Blok Diagram*.

3.5.1 *Use Case Diagram*

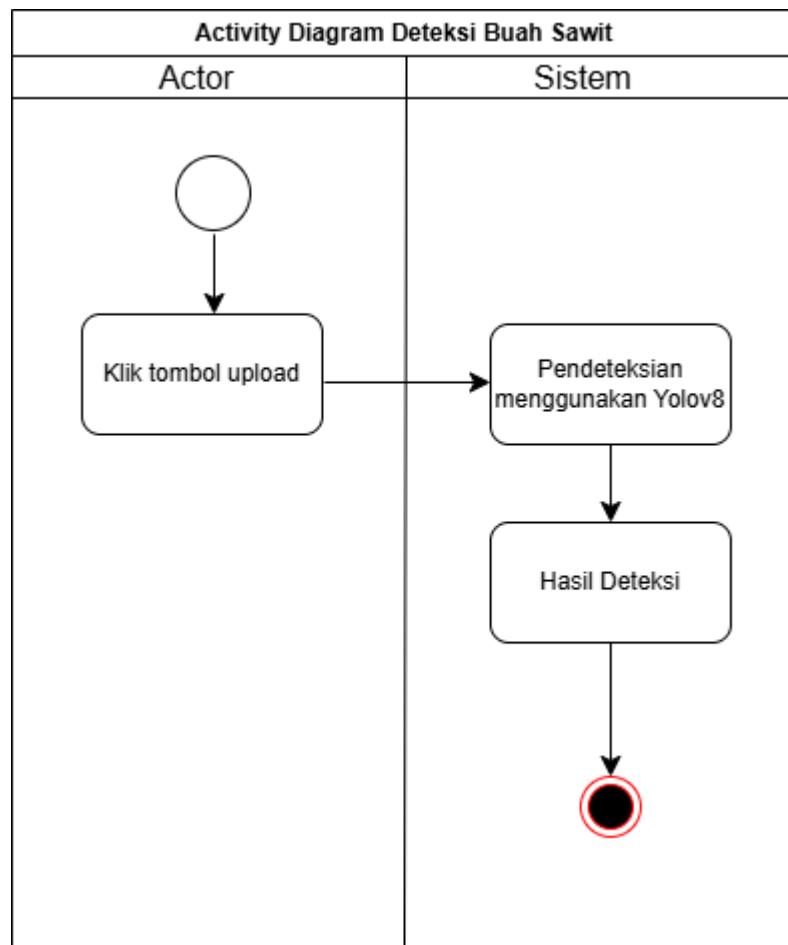
Diagram Kasus Pengguna (*Use Case Diagram*) adalah representasi grafis yang dapat menggambarkan interaksi antara pengguna dengan sistem. Diagram Kasus Pengguna ini memberikan penjelasan mengenai fungsi-fungsi yang dilakukan oleh sistem. *Use Case Diagram* pada sistem deteksi kematangan buah sawit tersaji pada Gambar 3.8.



Gambar 3.8 Use Case Diagram Deteksi Buah Sawit

3.5.2 Activity Diagram

Activity diagram merupakan suatu proses yang menjelaskan alur kerja suatu sistem yang dibuat, bertujuan untuk menjelaskan lebih lanjut proses skenario *use case*. Use case adalah komponen gambaran fungsional dalam sebuah sistem. Sehingga pengguna maupun pembuat saling mengenal dan mengerti mengenai alur sistem yang akan dibuat. Metodologi yang digunakan dalam analisis sistem untuk mengidentifikasi, mengklarifikasi, dan mengatur persyaratan sistem. Kasus penggunaan terdiri dari serangkaian kemungkinan urutan interaksi antara sistem dan pengguna.



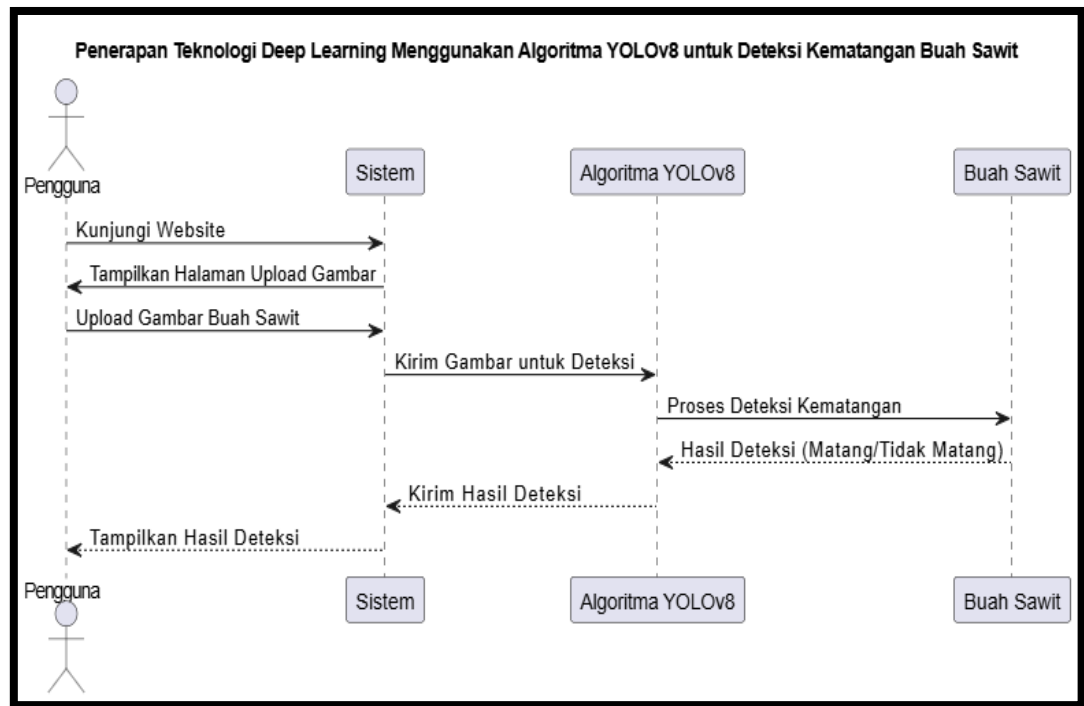
Gambar 3.9 Activity Diagram Deteksi Buah

Pada Gambar 3.9 menjelaskan mengenai interaksi yang terjadi antara user dan sistem, ketika user menekan tombol “Upload” pada halaman utama. Maka sistem akan melakukan proses Algoritma YOLOv8 untuk mendeteksi objek buah sawit

3.5.3 Sequence Diagram

Dalam *sequence diagram* ini, sistem menerima input berupa citra buah sawit, kemudian YOLOv8 melakukan deteksi objek dan klasifikasi tingkat

kematangan. Hasilnya akan dikirimkan ke pengguna melalui antarmuka web untuk ditampilkan. Gambar pada Sequence Diagram bisa di lihat pada gambar 3.10.



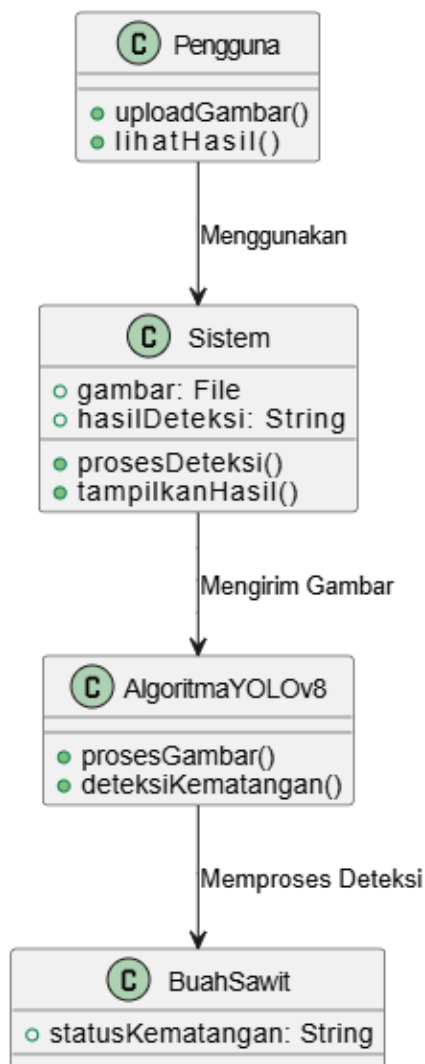
Gambar 3.10 Sequence Diagram Deteksi Buah

3.5.4 Class Diagram

Class diagram berikut menggambarkan struktur sistem deteksi kematangan buah sawit menggunakan YOLOv8 berbasis Deep Learning. Diagram ini menunjukkan hubungan antara kelas-kelas utama dalam sistem, termasuk atribut dan metode yang digunakan dalam setiap kelas. Dalam sistem ini, terdapat beberapa kelas utama seperti *User*, *ImageProcessor*, *YOLOv8Model*, dan *ResultDisplay*. Kelas *User* bertindak sebagai aktor yang mengunggah gambar sawit ke sistem. Kelas *ImageProcessor* bertanggung jawab untuk menerima dan memproses gambar

sebelum dikirim ke model *YOLOv8*. Kelas *YOLOv8Model* melakukan deteksi dan klasifikasi tingkat kematangan buah sawit, sedangkan *ResultDisplay* menampilkan hasil prediksi kepada pengguna melalui antarmuka web. *Class Diagram* bisa di lihat pada gambar 3.11.

Class Diagram: Deteksi Kematangan Buah Sawit

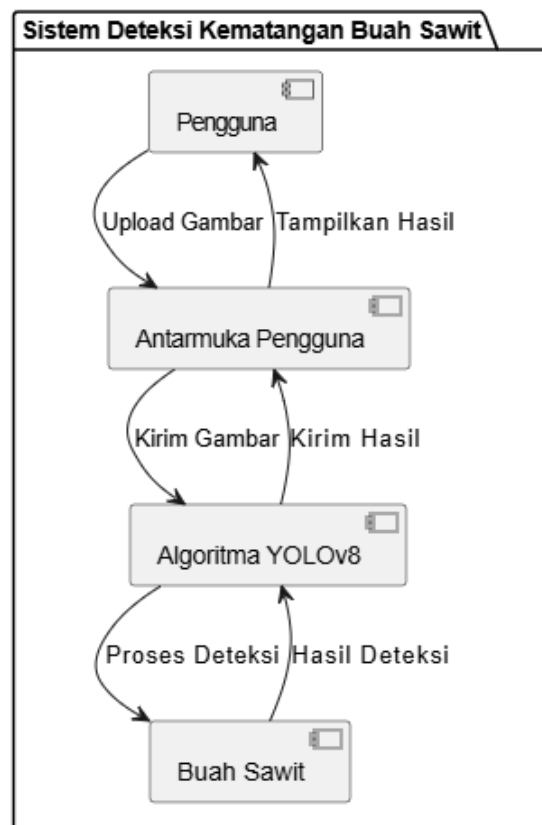


Gambar 3.11 Class Diagram Deteksi Buah Sawit

3.5.5 Blok Diagram

Blok diagram berikut menggambarkan alur kerja sistem deteksi kematangan buah sawit menggunakan *YOLOv8*. Pengguna mengunggah gambar sawit melalui antarmuka web, lalu gambar diproses oleh server dan dikirim ke model *YOLOv8* untuk deteksi. Hasil prediksi dikembalikan ke server dan ditampilkan di web, serta dapat disimpan dalam database untuk analisis lebih untuk memastikan sistem berjalan dengan optimal. Blok diagram pada penelitian bisa dilihat pada gambar 3.12.

Block Diagram: Deteksi Kematangan Buah Sawit

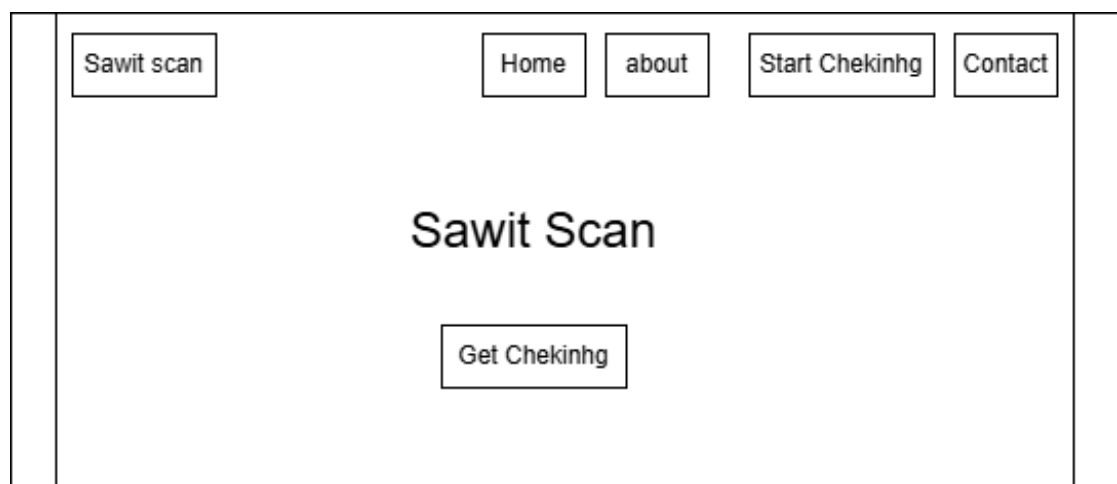


Gambar 3.12 Blok Diagram Deteksi Buah Sawit

3.6 Perancangan Antarmuka (*Interface*)

Pemodelan sistem merupakan proses perancangan dan penyusunan komponen-komponen sistem secara terstruktur untuk merepresentasikan alur kerja, fungsi, serta interaksi antar bagian dalam mencapai tujuan tertentu. Dalam konteks ini, pemodelan sistem YOLOv8 digunakan untuk membangun sebuah sistem deteksi otomatis yang mampu mengidentifikasi tingkat kematangan buah sawit melalui citra visual.

1. Perancangan antar muka Dashboard



Gambar 3.13 Rancangan Dashboard

Pada halaman dashboard ada beberapa menu yang dapat di pilih untuk masuk ke dalam semua laman yang ada. Antar muka dirancang lebih keliatan simple agar pengguna dapat dengan mudah mengakses nya.

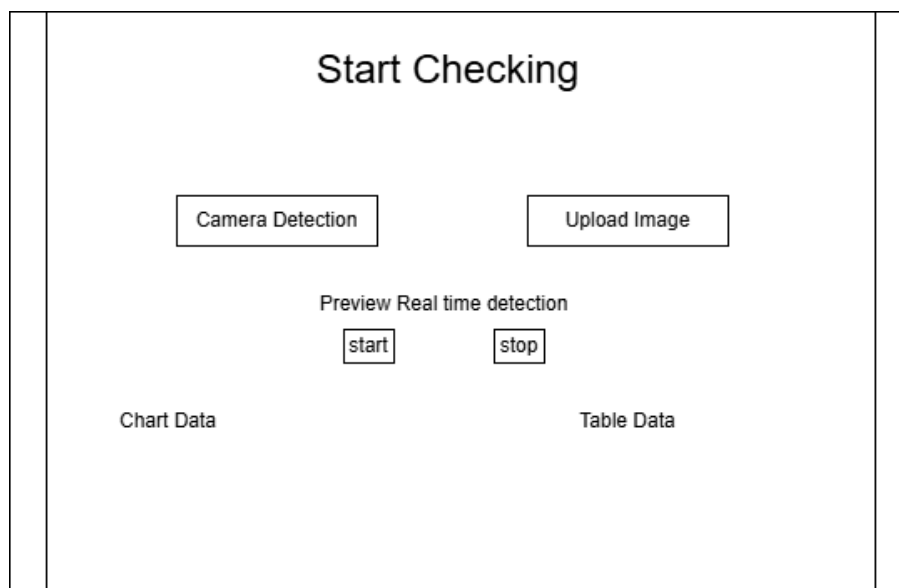
2. Perancangan antar muka About



Gambar 3.14 Rancangan About

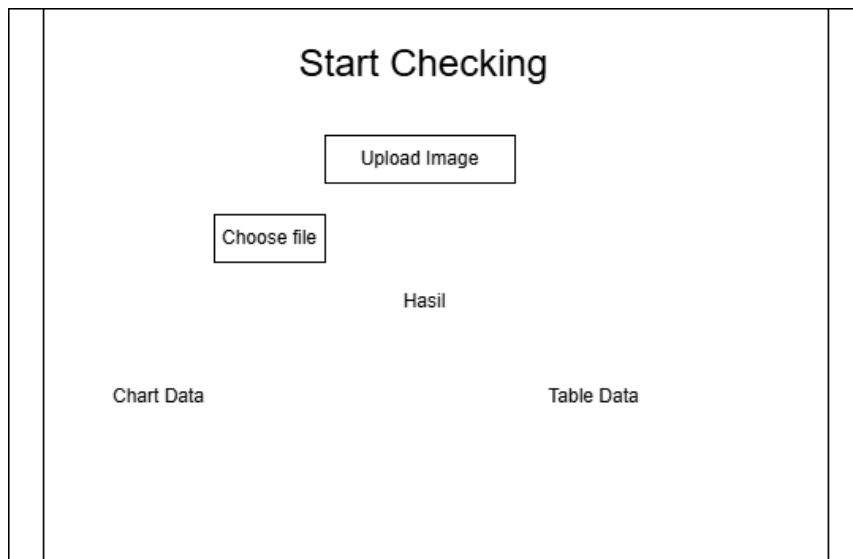
Pada halaman About hanya membahas mengenai beberapa alasan untuk melakukan penelitian in.

3. Perancangan antarmuka Start Cheking



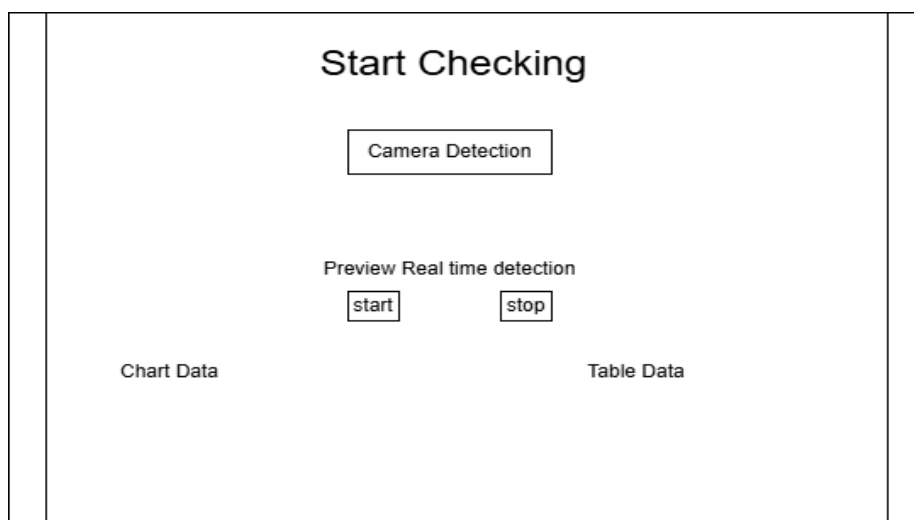
Gambar 3.15 Rancangan Start Cheking

4. Perancangan antar muka Start Cheking



Gambar 3.16 Rancangan Start Cheking

pada Start Cheking terdapat dua opsi dalam mendeteksi yaitu dengan upload image dan kamera deteksi.



Gambar 3.17 Rancangan Start Cheking