



**APLIKASI DETEKSI HAMA DAN PENYAKIT PADA BUAH KAKAO
MENGUNAKAN YOLO V8 BERBASIS WEBSITE**

TUGAS AKHIR

**ALDHITA DWI LESTARI
20670133**

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PGRI SEMARANG**

2024



**APLIKASI DETEKSI HAMA DAN PENYAKIT PADA BUAH KAKAO
MENGUNAKAN YOLO V8 BERBASIS WEBSITE**

TUGAS AKHIR

**Diajukan kepada Fakultas Teknik dan Informatika Universitas PGRI
Semarang untuk Penyusunan Tugas Akhir**

**ALDHITA DWI LESTARI
20670133**

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PGRI SEMARANG**

2024

HALAMAN PERSETUJUAN
APLIKASI DETEKSI HAMA DAN PENYAKIT PADA BUAH KAKAO
MENGGUNAKAN YOLOV8 BERBASIS WEBSITE

Disusun dan diajukan oleh

ALDHITA DWI LESTARI
NPM 20670133

Telah disetujui oleh pembimbing untuk dilanjutkan untuk menempuh sidang
laporan tugas akhir pada tanggal 10 Juli 2024

Pembimbing Utama,



Febrian Murti Dewanto, SE., M.Kom.
NPP 057801172

Pembimbing Pendamping,



Noora Qotrun Nada, S.T., M.Eng.
NPP 158201485

TUGAS AKHIR
APLIKASI DETEKSI HAMA DAN PENYAKIT PADA BUAH KAKAO
MENGGUNAKAN YOLOV8 BERBASIS WEBSITE

Disusun dan diajukan oleh

ALDHITA DWI LESTARI

20670133

Telah dipertahankan di depan Dewan Penguji pada tanggal 18 Juli 2024 dan
dinyatakan telah memenuhi syarat Dewan Penguji

Ketua,

Ibnu Toto Husodo, S.T., M.T
NIDN. 0602126902

Sekretaris,

Bambang Agus H, S.Kom., M.Kom
NIDN. 0601088201

Penguji I,

Khoiriva Latifah, S.Kom., M.Kom.
NIDN. 0617077801

Penguji II,

Noora Outfun N., S.T., M.Eng.
NIDN. 0626028201

Penguji III,

Bambang Agus H, S.Kom., M.Kom
NIDN. 0601088201

MOTTO DAN PERSEMBAHAN

MOTTO

"Keberhasilan bukanlah milik orang yang pintar. Keberhasilan adalah kepunyaan mereka yang senantiasa berusaha."

- B. J. Habibie

“Terkadang, kesulitan harus kamu rasakan terlebih dulu sebelum kebahagiaan yang sempurna datang kepadamu.”

-R. A. Kartini

Persembahan:

Saya persembahkan tugas akhir ini untuk :

1. Kedua orang tua tercinta.
2. Kakak saya.
3. Nenek saya.
4. Almamaterku Universitas PGRI Semarang.

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Aldhita Dwi Lestari

NPM : 20670133

Program Studi : Informatika

Fakultas : Teknik dan Informatika

Menyatakan dengan sebenarnya bahwa tugas akhir yang saya buat ini benar-benar merupakan hasil karya sendiri, bukan plagiarisme.

Apabila pada kemudian hari tugas akhir ini terbukti hasil plagiarisme, saya bersedia menerima sanksi atas perbuatan tersebut.

Semarang, 10 Juli 2024

Yang membuat pernyataan

Aldhita Dwi Lestari

NPM 206700133

ABSTRAK

Komoditas kakao di Indonesia memiliki peranan cukup penting bagi perekonomian nasional, mengingat Indonesia merupakan negara dengan areal penanaman kakao terluas dan penghasil kakao terbesar ketiga di dunia setelah Pantai Gading dan Ghana. Data statistik Direktorat Jenderal Perkebunan tahun 2019, menunjukkan bahwa produksi buah kakao mengalami penurunan setiap tahunnya dikarenakan adanya fakta bahwa kakao rentan terhadap serangan hama dan penyakit menjadi salah satu masalah yang berkontribusi terhadap penurunan hasil panen kakao. Hama yang sering menginfeksi tanaman kakao adalah Penggerek Buah Kakao (PBK) (*Conopomorpha cramerella*) dan Kepik penghisap buah (*Helopeltis spp*), serta penyakit yang umum menyerang kakao meliputi penyakit busuk buah yang disebabkan oleh *Phytophthora palmivora* L dan *Monilia roreri* Cif. Namun, tidak sedikit petani yang masih mengalami kebingungan dalam mendeteksi jenis penyakit dan hama tersebut. Ketidaktahuan para petani ataupun pihak perkebunan kakao mengenai jenis penyakit dan hama yang menyerang tanaman coklat menyebabkan banyak tanaman kakao tidak tertangani dengan tepat. Oleh karena itu, dalam mengatasi hal tersebut penelitian ini bertujuan untuk merancang dan membuat sebuah sistem aplikasi deteksi penyakit dan hama pada buah kakao menggunakan YOLOv8 yang dapat membantu memudahkan petani kakao dalam mengidentifikasi hama dan penyakit pada buah kakao secara efisien dan akurat. Aplikasi dirancang berbasis website dengan tampilan menarik dan sederhana untuk memudahkan pengguna. Metode pengembangan yang digunakan adalah model *Waterfall* dengan tingkat akurasi mencapai 94%. empat jenis pengujian, yaitu *black box*, *white box*, UAT (*User Acceptance Testing*), dan pengujian lapangan. Hasil pengujian *black box* dan *Whitebox* menunjukkan tingkat keberhasilan 100%, UAT mencapai 90,8%, sementara hasil pengujian lapangan menunjukkan bahwa aplikasi dapat mendeteksi hama dan penyakit pada buah kakao dengan baik dan dapat bermanfaat bagi pengguna, khususnya petani.

Kata Kunci : YOLOv8, Buah Kakao, Aplikasi Deteksi, Website, Hama, Penyakit.

PRAKATA

Dengan rasa syukur yang tak terhingga kepada Allah SWT atas berkah dan rahmat-Nya, penulis berhasil menyelesaikan tugas akhir ini dengan lancar. Tugas akhir berjudul "APLIKASI DETEKSI HAMA DAN PENYAKIT PADA BUAH KAKAO MENGGUNAKAN YOLOV8 BERBASIS WEBSITE" disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer.

Proses penyusunan tugas akhir ini tidak terlepas dari berbagai rintangan dan hambatan yang dihadapi. Namun, dengan bantuan, arahan, dan dukungan yang diberikan oleh berbagai pihak, terutama pembimbing, penulis berhasil mengatasi setiap kendala dengan baik. Oleh karena itu, penulis ingin mengucapkan rasa terima kasih yang tulus kepada:

1. Dr. Sri Suciati, M.Hum., selaku Rektor Universitas PGRI Semarang.
2. Bapak Ibnu Toto Husodo, ST., MT selaku Dekan Fakultas Teknik dan Informatika Universitas PGRI Semarang yang telah memberikan izin kepada penulis untuk melakukan penelitian.
3. Bapak Bambang Agus Herlambang, S.Kom., M.Kom. selaku Ketua Program Studi Informatika Universitas PGRI Semarang.
4. Bapak Febrian Murti Dewanto, SE., M.Kom. selaku Dosen Pembimbing I Program Studi Informatika Fakultas Teknik dan Informatika Universitas PGRI Semarang yang telah membimbing penulis saat menyusun tugas akhir dengan penuh dedikasi yang tinggi.
5. Ibu Noora Qotrun Nada, S.T., M.Eng. selaku Dosen Pembimbing II Program Studi Informatika Fakultas Teknik dan Informatika Universitas PGRI Semarang yang telah membimbing penulis saat menyusun tugas akhir dengan penuh dedikasi yang tinggi.
6. Seluruh Dosen beserta Staf di Program Studi Informatika Fakultas Teknik dan Informatika Universitas PGRI Semarang atas segala ilmu dan bimbingan selama masa studi.
7. Kedua orang tua tercinta Bapak Mardiyana dan Ibu Ngatini, kakak, serta nenek

penulis yang senantiasa mendoakan penulis demi terselesaikannya tugas akhir ini.

8. Untuk seseorang dengan NPM 20670087 yang selalu memberikan semangat dan dukungan.
9. Serta untuk semua teman-teman dekat penulis yang tidak bisa disebutkan satu persatu yang telah memberikan semangat dalam penyusunan tugas akhir ini.

Akhir kata penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dan memberikan dukungan. Penulis berharap tugas akhir ini dapat memberikan manfaat bagi banyak orang khususnya di bidang Informatika.

Semarang, 10 Juli 2024

Penulis

DAFTAR ISI

SAMPUL LUAR.....	i
SAMPUL DALAM.....	ii
HALAMAN PERSETUJUAN.....	Error! Bookmark not defined.
HALAMAN PENGESAHAN.....	Error! Bookmark not defined.
MOTTO DAN PERSEMBAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN.....	vi
ABSTRAK.....	vii
PRAKATA.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
BAB I.....	1
PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Identifikasi Masalah.....	3
C. Rumusan Masalah.....	3
D. Batasan Masalah.....	3
E. Tujuan Penelitian.....	4
F. Manfaat Penelitian.....	4
BAB II.....	6
KAJIAN PUSTAKA/TEORI.....	6
A. Tinjauan Pustaka.....	6
B. Landasan Teori.....	10

C. Kerangka Berpikir.....	22
BAB III.....	24
METODE PENELITIAN	24
A. Pendekatan Penelitian.....	24
B. Fokus Penelitian.....	24
C. Desain Penelitian	24
D. Teknik Pengumpulan Data.....	26
E. Teknik Analisis Data	27
BAB IV	28
HASIL PENGEMBANGAN DAN PEMBAHASAN	28
A. Hasil.....	28
1. Pemodelan Data.....	28
2. Pengembangan Sitem	46
B. Pembahasan	88
BAB V.....	98
KESIMPULAN DAN SARAN	98
A. KESIMPULAN.....	98
B. SARAN	100
DAFTAR PUSTAKA	101
LAMPIRAN	105

DAFTAR TABEL

Tabel 2. 1 Perbandingan Penelitian Sebelumnya.....	6
Tabel 2. 2 Use Case Diagram.....	17
Tabel 2. 3 Activity Diagram.....	18
Tabel 2. 4 Class Diagram.....	19
Tabel 2. 5 Sequence diagram	20
Tabel 4. 1 Tabel Jumlah dataset tiap kelas.....	32
Tabel 4. 2 Tabel jumlah train data.....	32
Tabel 4. 3 Tabel jumlah valid data.....	33
Tabel 4. 4 Tabel jumlah test data	33
Tabel 4. 5 Train data folder.....	47
Tabel 4. 6 Valid data folder.....	48
Tabel 4. 7 Test data folder.....	48
Tabel 4. 8 Kebutuhan perangkat keras komputer.....	49
Tabel 4. 9 Kebutuhan perangkat lunak komputer	49
Tabel 4. 10 Hasil Pengujian Black Box	70
Tabel 4. 11 Whitebox Testing.....	74
Tabel 4. 12 Value Test	81
Tabel 4. 13 User Acceptance Testing (UAT)	83

DAFTAR GAMBAR

Gambar 2. 1 Langkah-langkah metode waterfall.....	15
Gambar 2. 2 Kerangka Berpikir.....	23
Gambar 3. 1 Skema Desain Penelitian.....	25
Gambar 4. 1 Dataset Hama dan Penyakit Buah Kakao.....	28
Gambar 4. 2 Annotation dan Labelling Data.....	31
Gambar 4. 3 Preprocessing data.....	34
Gambar 4. 4 Training Model.....	35
Gambar 4. 5 Confusion matrix.....	36
Gambar 4. 6 Rumus Confusion Matrix.....	36
Gambar 4. 7 F1-Confidence Curve.....	38
Gambar 4. 8 Precision-Confidence Curve.....	39
Gambar 4. 9 Recall-Confidence Curve.....	40
Gambar 4. 10 Precision-Recall Curve.....	41
Gambar 4. 11 Test model kelas buah sehat.....	42
Gambar 4. 12 Test model kelas busuk buah.....	43
Gambar 4. 13 Test model kelas penggerek buah.....	44
Gambar 4. 14 Test model kelas penghisap buah.....	45
Gambar 4. 15 Source code Upload model training.....	46
Gambar 4. 16 Use Case Diagram.....	51
Gambar 4. 17 Activity Diagram Beranda.....	52
Gambar 4. 18 Acitivity diagram deteksi.....	53
Gambar 4. 19 Activity diagram artikel.....	54
Gambar 4. 20 Acitivity Diagram Tentang.....	55
Gambar 4. 21 Sequence diagram beranda.....	56
Gambar 4. 22 Sequence diagram deteksi.....	57
Gambar 4. 23 Sequence diagram artikel.....	58
Gambar 4. 24 Sequence diagram tentang.....	59
Gambar 4. 25 Class diagram.....	59
Gambar 4. 26 Desain User Interface Halaman Beranda.....	61

Gambar 4. 27 Desain User Interface Halaman Deteksi	62
Gambar 4. 28 Halaman Artikel	63
Gambar 4. 29 Halaman Tentang	64
Gambar 4. 30 Implementasi halaman beranda.....	66
Gambar 4. 31 Implementasi halaman deteksi	67
Gambar 4. 32 Implementasi halaman artikel	68
Gambar 4. 33 Implementasi halaman tentang.....	69
Gambar 4. 34 Flowgraph Basis Path.....	81
Gambar 4. 35 Pengujian Lapangan dengan Ketua Kelompok Tani Ngudi Mulyo	85
Gambar 4. 36 Pengujian Lapangan bersama petani kakao.....	86
Gambar 4. 37 Hasil Pengujian Lapangan.....	87

BAB I

PENDAHULUAN

A. Latar Belakang

Pertumbuhan industri pertanian, khususnya produksi buah-buahan, memiliki peran penting dalam pemenuhan kebutuhan pangan global. Salah satu buah yang memiliki peran signifikan dalam industri pertanian adalah buah kakao (*Theobroma cacao L.*). Komoditas kakao di Indonesia memiliki peranan cukup penting bagi perekonomian nasional. Komoditas kakao mempunyai potensi untuk dikembangkan melalui pengelolaan berkelanjutan karena selain bermanfaat bagi perekonomian nasional, juga mendorong pertumbuhan wilayah suatu daerah dan agroindustri mengingat Indonesia merupakan negara dengan areal penanaman kakao terluas dan penghasil kakao terbesar ketiga di dunia setelah Pantai Gading dan Ghana[1].

Industri pertanian kakao ini memiliki permintaan yang sangat besar dalam pasar global yang menghasilkan produk turunannya seperti coklat, kakao bubuk, dan minuman coklat yang sangat digemari oleh orang-orang berbagai negara di dunia. Sehingga produktivitas dan kualitas dari buah kakao menjadi sangat perlu untuk diperhatikan. Data statistik Direktorat Jenderal Perkebunan tahun 2019, menunjukkan bahwa produksi buah kakao mengalami penurunan setiap tahunnya dikarenakan adanya fakta bahwa kakao rentan terhadap serangan hama dan penyakit menjadi salah satu masalah yang berkontribusi terhadap penurunan hasil panen kakao[2]. Gangguan hama ini dapat mengakibatkan kematian tanaman, penurunan kualitas buah, dan kerusakan fisik. Gangguan ini diperkirakan menyebabkan kerugian global di seluruh dunia mencapai 30–40% produksi kakao dunia [3]. Hama yang sering menginfeksi tanaman kakao adalah Penggerek Buah Kakao (PBK) (*Conopomorpha cramerella*) dan Kepik penghisap buah (*Helopeltis spp*) (Pusat Penelitian Kakao dan Kopi Indonesia, 2013), sementara itu penyakit

yang umum menyerang kakao meliputi penyakit busuk buah yang disebabkan oleh *Phytophthora palmivora* L dan *Monilia roreri* Cif [4]. Penyakit tersebut dapat mengakibatkan penurunan kualitas buah jika tidak segera ditangani dengan tepat. Namun, tidak sedikit petani yang masih mengalami kebingungan dalam mendeteksi jenis penyakit dan hama tersebut. Ketidaktahuan para petani ataupun pihak perkebunan kakao mengenai jenis penyakit dan hama yang menyerang tanaman coklat menyebabkan banyak tanaman kakao tidak tertangani dengan tepat[5]. Oleh karena itu, diperlukan suatu cara agar dapat memudahkan petani kakao untuk mendeteksi jenis hama dan penyakit pada buah kakao.

Di era digital saat ini, teknologi kecerdasan buatan telah menawarkan pendekatan inovatif dalam deteksi dan pemantauan penyakit tanaman. Salah satu teknologi yang dapat digunakan adalah *You Only Look Once* (YOLO), sebuah metode deteksi objek berbasis deep learning yang cepat dan efisien. YOLO merupakan metode identifikasi objek secara real-time yang memanfaatkan jaringan saraf convolutional, dimana setiap lapisan *convolutional* jaringan memiliki proses konvolusi[6]. Pemanfaatan teknologi deteksi objek menggunakan algoritma seperti YOLOv8 dapat digunakan sebagai landasan untuk meningkatkan efisiensi dan efektivitas pengelolaan penyakit tanaman. Teknologi ini memungkinkan petani untuk mendeteksi dini adanya hama dan penyakit pada buah kakao, sehingga tindakan pencegahan dan pengendalian dapat dilakukan dengan tepat waktu yang diharapkan dapat membantu mengurangi akibat kerugian bagi petani kakao.

Oleh karena itu, penelitian ini bertujuan untuk merancang dan membuat sebuah sistem aplikasi deteksi penyakit dan hama pada buah kakao menggunakan YOLOv8 yang dapat membantu memudahkan petani kakao dalam mengidentifikasi hama dan penyakit pada buah kakao secara efisien dan akurat. Aplikasi dirancang berbasis website yang didesain dengan tampilan menarik dan sederhana untuk memudahkan pengguna dalam menggunakan aplikasi deteksi ini. Diharapkan penelitian ini dapat memberikan kontribusi signifikan dalam meningkatkan produktivitas dan kesejahteraan petani dalam

rangka mengurangi kerugian yang dapat disebabkan oleh penyakit dan hama pada buah kakao.

B. Identifikasi Masalah

Permasalahan penelitian dapat diidentifikasi sebagai berikut:

1. Adanya berbagai jenis penyakit dan hama yang menyerang buah kakao sehingga dapat menurunkan produktivitas buah kakao yang berkualitas baik.
2. Petani mengalami kebingungan dalam mengidentifikasi jenis penyakit dan hama pada buah kakao sehingga kesulitan dalam melakukan tindak pencegahan dan pengendalian hama secara tepat.
3. Belum adanya sebuah teknologi yang dapat membantu memudahkan petani kakao untuk mendeteksi jenis penyakit dan hama yang menyerang buah kakao secara efisien dan akurat.

C. Rumusan Masalah

Bagaimana merancang dan membangun aplikasi deteksi hama dan penyakit pada buah kakao menggunakan Yolov8 berbasis website?

D. Batasan Masalah

Mengingat terbatasnya tenaga dan waktu yang ada, serta menjaga dan menghindari pembahasan masalah yang terlalu luas, maka penulis membatasi permasalahan pada penelitian yang meliputi:

1. Aplikasi ini berfokus pada deteksi penyakit dan hama yang menyerang pada bagian buah kakao saja.
2. Dataset diperoleh berdasarkan hasil temuan dari dataset publik Roboflow, yang kemudian dilakukan manajemen kelas kembali melalui Roboflow dengan proyek pribadi penulis bernama dataset “Skripsi” yang hanya terdiri dari empat kelas jenis penyakit dan hama pada buah kakao yaitu; Busuk Buah, Penggerek Buah, Penghisap Buah, dan Buah Sehat.

3. Model dilatih menggunakan YOLO versi 8 sebagai teknologi deteksi objek dalam penentuan jenis hama dan penyakit pada buah kakao melalui Google Colaboratory.
4. Aplikasi hanya dapat mendeteksi dengan metode upload gambar/video saja dengan format gambar jpg dan format video mp4.
5. Aplikasi dirancang dalam bentuk sistem berbasis *website* dengan menggunakan Bahasa PHP dan Python sebagai bahasa pemrogramannya.
6. Aplikasi dirancang menggunakan model *waterfall* yang dilakukan hanya sampai pada tahap pengujian.

E. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dijabarkan sebelumnya, maka tujuan dari penelitian ini yaitu sebagai berikut:

Untuk merancang dan membangun aplikasi deteksi hama dan penyakit pada buah kakao menggunakan Yolov8 Berbasis Website.

F. Manfaat Penelitian

Manfaat yang didapat dari penyusunan tugas akhir ini adalah:

1. Bagi Mahasiswa

Untuk melatih dan menambah kembali serta mengukur seberapa daya tangkap penulis dalam mempraktekkan ilmu yang diperoleh di bangku kuliah dan menerapkannya dalam lingkungan kehidupan yang membutuhkan.

2. Bagi Akademik

Sebagai tolak ukur keberhasilan proses belajar mengajar yang dapat digunakan sebagai bahan evaluasi bagi pihak akademik dan sebagai referensi bagi mahasiswa dalam penelitian lebih lanjut yang berkaitan dengan studi yang dibahas dalam tugas akhir ini.

3. Bagi Petani/Pihak Perkebunan Kakao

Diharapkan dapat menjadi masukan atau pertimbangan bagi petani atau pihak perkebunan kakao untuk dapat menerapkan aplikasi tersebut dalam proses identifikasi deteksi jenis penyakit pada buah kakao sebagai cara dalam meminimalisir adanya kerugian yang disebabkan oleh hama dan penyakit pada buah kakao dengan memberikan hasil deteksi yang efisien dan akurat.

BAB II

KAJIAN PUSTAKA/TEORI

A. Tinjauan Pustaka

Sebelumnya telah terdapat beberapa penelitian tentang perancangan sistem deteksi penyakit pada suatu tanaman dengan menggunakan beberapa metode. Setiap penelitian memiliki kriteria dan pola yang berbeda satu sama lain. Berikut adalah tabel perbandingan penelitian sebelumnya yang menggunakan beberapa metode dalam:

Tabel 2. 1 Perbandingan Penelitian Sebelumnya

No	Nama dan Tahun Penelitian	Judul	Metode	Hasil
1.	Nadini Mardiah Yasen, Silfia Rifka, Rikki Vitria, Yulindon (2023)	Pemanfaatan Yolo Untuk Deteksi Hama Dan Penyakit Pada Daun Cabai Menggunakan Metode <i>Deep Learning</i>	Metode <i>Deep Learning</i> dengan YOLO	Kemampuan aplikasi dalam mengidentifikasi hama dan penyakit pada daun tanaman cabai secara akurat ditunjukkan oleh hasil pengujian. Akurasi hasil pelatihan sebesar 73% menunjukkan bahwa akurasi yang dihasilkan tergolong sangat baik, sehingga dapat digunakan oleh masyarakat.

2.	Ardiansyah, Nur Fitrianingsih Hasan (2023)	Deteksi dan Klasifikasi Penyakit Pada Daun Kopi Menggunakan Yolov7	Metode <i>Deep Learning</i> dengan Yolov7	Hasil dari penelitian ini menunjukkan bahwa deteksi dan klasifikasi objek penyakit tanaman kopi menggunakan YOLOv7, yang berfokus pada prapemrosesan, penambahan data, anotasi, atau pelabelan objek, memberikan hasil yang baik. Selain melatih dataset dengan parameter berikut: model yolov7, pengoptimalisasi SGD, 300 epoch, 8 pekerja, dan 16 ukuran batch, penelitian yang menggunakan perangkat Google Colab yang dilengkapi dengan GPU Tesla T4 ini memberikan hasil sebagai berikut: F1-score 0.93, Precision 0.926, Recall 0.932, mAP@IoU.5 0.956, dan mAP@IoU.5:.95 0.927 untuk semua kelas data yang dilatih.
----	---	---	--	--

				Untuk kelas biner, hasil teratas adalah F1-score 0.99, Precision 0.991, Recall 1, mAP@IoU.5 0.998, dan mAP@IoU.5: .95 0.994.
3.	Mohammad Yazdi Pusadan, Syahrullah, Merry, Ahmad Imam Abdullah (2022)	K-Nearest Neighbor and Feature Extraction on Detection of Pest and Diseases of Cocoa	<i>K-Nearest Neighbor</i> dan <i>Feature Extraction</i>	Berdasarkan hasil dan analisis pengujian, Sistem ini dapat mendeteksi buah kakao berdasarkan kategori yang ada, namun tidak semua data terdeteksi dengan benar karena akurasi yang tidak terlalu baik. Penelitian ini menggunakan Teknik pengolahan citra digital. Hue, Saturation, Value (HSV) adalah metode yang digunakan untuk mengekstrak fitur warna, dan K-Nearest Neighbor adalah algoritma klasifikasi. Data yang digunakan sebanyak 150 citra yang dibagi menjadi 70% data latih dan 30% data uji. untuk pengujian.

				<p>Pendekatan holdout menghasilkan data setelah uji coba dengan nilai k sebesar 5, 7, 11, dan 13. Akurasi terbaik ditemukan pada 84,44% ketika k = 5. Selain itu, akurasi terbaik pada uji coba cross-validation k-5 juga didapatkan pada k=5, dengan nilai akurasi sebesar 99,33%.</p>
4.	<p>Joelyan Vicky, Frisca Ayu, Bagas Julianto (2023)</p>	<p>Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN</p>	<p>CNN</p>	<p>Dari hasil pengujian yang dilakukan dengan model dan testing data yang dijalankan terhadap data daun alpukat yang terkena penyakit, penelitian ini mempunyai hasil yang baik dimana metode CNN mampu mendeteksi penyakit pada daun alpukat dengan tingkat akurasi sebesar 80% untuk pencocokan jenis penyakit daun alpukat yang teridentifikasi pada database yang tersimpan di sistem</p>

				mobile applicatiom yang dibuat.
5.	Md Ershadul Haque, Ashikur Rahman, Iftekhar Junaeid, Samiul Ul Hoque, Manoranjan Paul (2022)	Rice Leaf Disease Classification and Detection Using YOLOv5	CNN (YOLO v5)	Penelitian ini melakukan pendeteksian penyakit pada tanaman padi dengan melakukan anotasi pada 1500 dataset gambar yang mengusulkan klasifikasi dan deteksi penyakit daun padi. Model pembelajaran mendalam YOLOv5 menghasilkan presisi 90%, recall 67%, dan F1-score 81%.

B. Landasan Teori

1. Aplikasi

Aplikasi berasal dari bahasa Inggris app, yang merupakan elemen penting pada smartphone ini yang dibuat dengan menggunakan bahasa pemrograman. Secara umum, program ini didesain untuk dapat memproses perintah pengguna sebagai masukan dan mengeluarkan hasil yang diinginkan[7]. Aplikasi diartikan sebagai program perangkat lunak yang beroperasi pada sistem tertentu dan sangat membantu dalam mendukung berbagai tugas yang dilakukan oleh manusia[8]. Lembar kerja, pemutar media, pengolah kata, dan pengolah data adalah jenis perangkat lunak aplikasi yang utama[9].

2. *Artificial Intelligence*

kecerdasan buatan (*artificial intelligence*) dikenal sebagai ilmu pengetahuan dan teknik untuk mengembangkan teknologi cerdas, khususnya program komputer pintar yang memungkinkan komputer melakukan semua tugas yang biasanya dilakukan manusia. Kecerdasan buatan dibuat dengan menstandarkan kemampuan manusia seperti suara, penglihatan, dan kognisi [10].

3. *Machine Learning*

Dalam bukunya Ibnu Daqiqil Id (2020), Arthur Samuel menyatakan bahwa *Machine Learning* adalah sebuah subbidang ilmu komputer yang memungkinkan komputer untuk belajar tanpa pemrograman secara eksplisit. ML terdiri dari teknik generik (umum) yang tidak memerlukan penulisan kode khusus untuk menghasilkan sesuatu yang menarik atau bermanfaat dari berbagai fakta. Pada dasarnya, algoritma generik dapat membuat aturan model atau kesimpulan dari sekumpulan data ketika diberikan sekumpulan data. Sebagai contoh, email yang mengandung spam dapat diidentifikasi menggunakan algoritma yang dirancang untuk mengidentifikasi tulisan tangan, semua tanpa memerlukan perubahan pengkodean. Ketika diberikan set data pelatihan yang berbeda, algoritma yang sama menghasilkan penalaran klasifikasi yang berbeda [11].

4. *Deep Learning*

Deep Learning, sebuah subbidang dari *Machine Learning*, melatih komputer untuk melakukan tugas-tugas yang mirip dengan manusia dengan menggunakan model matematika yang terinspirasi oleh jaringan saraf otak manusia. Hal ini memungkinkan komputer untuk memahami dan menganalisis data secara lebih mendalam, sehingga memberikan hasil yang lebih akurat [12].

5. YOLO (*You Only Look Once*)

Algoritma *You Only Look Once* (YOLO) merupakan pengembangan dari algoritma *Convolutional Neural Network* (CNN). YOLO dirancang untuk memprediksi kotak pembatas (bounding box) serta kelas objek secara langsung dari gambar secara bersamaan dalam satu proses inferensi.

Arsitektur YOLO terdiri dari lapisan-lapisan konvolusi yang bertugas untuk mengekstraksi fitur-fitur penting dari gambar, diikuti oleh lapisan-lapisan *fully connected* yang bertanggung jawab untuk melakukan prediksi objek dan kotak pembatas. Keunggulan utama YOLO adalah kemampuannya untuk melakukan deteksi objek dalam waktu nyata (real-time) dengan akurasi yang tinggi. Dengan mengintegrasikan deteksi objek dan klasifikasi dalam satu proses, YOLO memungkinkan penggunaan yang efisien dalam aplikasi-aplikasi yang membutuhkan respons cepat, seperti dalam video streaming atau pengawasan berbasis kamera. Selain itu, YOLO juga dikenal karena kemampuannya dalam mendeteksi objek dengan ukuran yang bervariasi serta kemampuannya dalam mengatasi masalah overlapping antar objek. Namun, seperti kebanyakan algoritma deep learning, kinerja YOLO sangat dipengaruhi oleh kualitas data pelatihan dan parameter-parameter seperti ukuran jendela kotak pembatas dan jumlah *grid cell* pada gambar [13].

YOLO v8 adalah algoritma pendeteksian terbaru yang diperkenalkan dalam rangkaian algoritma YOLO saat ini, yang dapat digunakan untuk tugas-tugas seperti pendeteksian target, klasifikasi gambar, dan segmentasi instance [14]. YOLO v8 merupakan generasi terbaru dari YOLO v7. Digunakan YOLO v8 karena memiliki kelebihan dalam melakukan deteksi objek yang lebih baik, pelacakan sistem yang akurat, keamanan sistem yang tangguh, dan masih banyak lagi [15].

6. Definisi Web

Website atau web adalah kumpulan halaman yang terhubung dengan koneksi internet dan mencakup data digital dalam bentuk teks, foto, audio, video, dan animasi lainnya. Sederhananya, situs web adalah halaman yang berisi informasi yang dapat diakses oleh pengguna melalui browser dan juga dapat menyediakan informasi yang relevan bagi penggunanya [16].

Kualitas dapat ditentukan oleh sejumlah faktor, seperti kemampuan situs web untuk melayani tujuan yang dimaksudkan, kemudahan penggunaan bagi pengguna, aksesibilitas dari mana saja dan kapan saja, dan persyaratan lainnya. *Usability* adalah salah satu faktor yang dianggap penting untuk kualitas situs web. *Usability* adalah aspek yang mengukur tingkat kepuasan pengguna serta seberapa mudah bagi pengguna untuk memahami dan menggunakannya untuk mencapai tujuan mereka [17].

7. Roboflow

Roboflow adalah sebuah platform yang memfasilitasi kemudahan dalam pengumpulan dan pengelolaan dataset dalam pemrosesan citra dan computer vision [18]. Platform ini menyediakan berbagai alat untuk mempermudah langkah-langkah seperti pra-pemrosesan gambar, pengelolaan dataset, pembuatan anotasi, dan konversi format dataset. Dengan menggunakan Roboflow, pengguna dapat dengan mudah mengatur dan mempersiapkan dataset mereka untuk pelatihan model pembelajaran mesin dengan lebih efisien dan efektif.

8. Flask

Flask adalah sebuah *micro-framework* berbasis bahasa Python, Flask memiliki beberapa tools dan library. Flask digunakan untuk meningkatkan efisiensi pengembangan. Flask digunakan karena tidak membutuhkan banyak sumber daya dan dapat beroperasi pada program dengan memori yang rendah dan konsumsi energi yang rendah. Framework Flask bersifat ringan, namun tetap memiliki kemampuan untuk menjalankan fungsinya sesuai kebutuhan. Flask terdiri dari dua bagian: Jinja2 adalah mesin template dan Werkzeug digunakan untuk routing,

debugging, dan fungsionalitas Web Server Interface Gateway (WSGI) [19].

9. Metode Waterfall

Metode *waterfall* adalah model untuk mengembangkan sistem informasi secara sistematis dan sekuensial. Waterfall merupakan salah satu metode SDLC, berisi persyaratan bahwa setiap langkah kerja harus diselesaikan sebelum melanjutkan ke langkah berikutnya [20]. Pengertian dari metode *waterfall* adalah rangkaian pada saat proses pengembangan *software* secara berurutan, dimana kemajuan akan tetap terus mengalir kebawah melalui proses perencanaan, pemodelan, implementasi dan pengujian [21]. Berikut adalah penjelasan dalam setiap langkah pada metode waterfall:

a. Analisis Kebutuhan (*Requirements Analysis*)

Tahap pertama ini melibatkan proses identifikasi dan pemahaman terhadap permintaan kebutuhan pengguna dan pemangku kepentingan untuk sistem atau aplikasi yang hendak dikembangkan. Pengguna akan dihubungi oleh tim pengembangan untuk mengumpulkan dan memeriksa persyaratan fungsional dan non-fungsional. Dokumen persyaratan yang menguraikan fitur, kemampuan, dan tujuan dari sistem yang akan dikembangkan adalah produk akhir dari langkah ini.

b. Desain (*Design*)

Pada tahapan ini yaitu melakukan perancangan sistem dilakukan berdasarkan dokumen kebutuhan. Perancangan sistem terdiri dari beberapa aspek, termasuk perancangan arsitektur, desain antarmuka pengguna, desain basis data, dan desain modul atau komponen sistem. Tujuan dari tahap ini adalah menghasilkan rancangan yang jelas dan komprehensif untuk sistem yang akan dikembangkan.

c. Pengembangan (*Development*)

Setelah perancangan selesai, tim pengembang akan mulai mengimplementasikan rancangan yang telah dirancang. Desain yang disetujui berfungsi sebagai dasar untuk menulis kode program.

Pengkodean, pengujian unit, dan integrasi komponen pada sistem adalah bagian dari langkah implementasi. Tahap pengembangan ini menghasilkan kemampuan aplikasi atau sistem yang dihasilkan untuk berfungsi sesuai dengan desain yang ditentukan oleh perancang sistem.

d. Pengujian (*Testing*)

Pada tahapan ini, sistem yang telah diimplementasikan akan dilakukan pengujian untuk memastikan bahwa sistem tersebut dapat berfungsi sesuai dengan persyaratan yang telah ditetapkan. Pengujian dilakukan untuk mengidentifikasi dan memperbaiki cacat atau kesalahan dalam sistem. Tes yang dapat dilakukan meliputi tes unit, tes integrasi, tes fungsional, dan tes kinerja. Tujuan dari tahap ini adalah memastikan kualitas dan kehandalan sistem sebelum diluncurkan.

e. Pemeliharaan (*Maintenance*)

Setelah sistem diuji dan diluncurkan, tahap pemeliharaan dimulai. Pemeliharaan melibatkan pemantauan kinerja sistem, pemecahan masalah, dan peningkatan sistem jika diperlukan. Pemeliharaan dapat mencakup perbaikan *bug*, peningkatan fungsionalitas, dan penyesuaian dengan perubahan kebutuhan pengguna. Tujuan dari tahap ini adalah memastikan sistem tetap berjalan dengan baik dan dapat diandalkan dalam jangka panjang.



Gambar 2. 1 Langkah-langkah metode *waterfall*


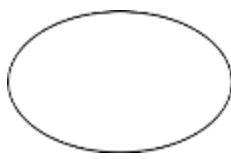
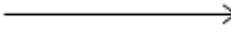
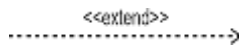
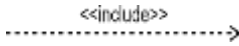
10. UML

Unified Modelling Language (UML) merupakan salah satu standar Bahasa yang sering digunakan di industri untuk mendefinisikan kebutuhan, melakukan analisis dan perancangan, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Kebutuhan akan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan sistem perangkat lunak merupakan sebab dari munculnya pengembangan UML [22]. UML digunakan untuk memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi object. Selain itu, untuk mengembangkan bahasa pemodelan yang dapat dioperasikan oleh mesin dan manusia. UML terdiri dari berbagai diagram yang masing-masing merepresentasikan aspek tertentu dari sistem. Berikut adalah beberapa jenis diagram UML yang umum digunakan:

a. *Use Case Diagram*

Use case diagram adalah Salah satu jenis diagram UML yang menggambarkan hubungan interaksi antara aktor dan sistem [23]. *Use case* dapat digunakan untuk menjelaskan analisis tingkat atas dari kebutuhan sistem, serta fungsionalitas sistem dan interaksi dengan aktor. Aktor adalah entitas yang berinteraksi dengan sistem. Secara umum *use case diagram*, dapat digunakan untuk merepresentasikan interaksi yang diinginkan sistem dengan pengguna, mengatur persyaratan fungsional, mendefinisikan konteks dan kebutuhan sistem, serta memodelkan aliran kejadian dalam *use case* [24]. Untuk penjelasan detail tentang komponen-komponen use case diagram dapat dilihat pada Tabel 2.2.

Tabel 2. 2 *Use Case Diagram*



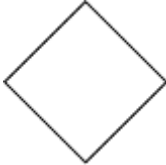


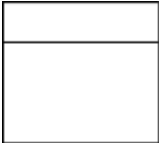
No	Simbol	Keterangan
1.		Aktor : Mewakili peran orang, sistem yang lain, atau alat Ketika berkomunikasi dengan <i>use case</i> .
2.		<i>Use Case</i> : Abstraksi dan interaksi antara sistem dan aktor.
3.		<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i> .
4.		Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.
5.		Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.

b. *Activity Diagram*

Activity diagram adalah suatu diagram yang merepresentasikan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses [25]. Diagram aktivitas bersifat dinamis. Diagram ini merupakan jenis diagram state khusus yang menunjukkan aliran dari satu aktivitas ke aktivitas lainnya dalam sebuah sistem. Penggunaan *activity diagram* meliputi hal-hal berikut seperti, pemodelan aliran kerja yang menggunakan aktivitas, pemodelan kebutuhan bisnis, dan pemahaman

tingkat tinggi dari fungsionalitas sistem [26]. Untuk penjelasan tentang komponen-komponen *activity diagram* dapat dilihat pada Tabel 2.3.





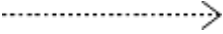
Tabel 2. 3 *Activity Diagram*


No	Simbol	Keterangan
1.		Status awal / <i>start</i> : Sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas : Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan / <i>Decision</i> : Percabangan dimana ada pilihan aktivitas yang lebih dari satu
4.		Penggabungan / <i>join</i> : Penggabungan yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
5.		Status akhir / <i>End</i> : Status akhir dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.		<i>Swimlane</i> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

c. *Class Diagram*

Class diagram menggambarkan struktur sistem dengan mengidentifikasi kelas-kelas yang akan digunakan untuk mengembangkan sistem. *Class diagram* terdiri dari atribut dan operasi yang memungkinkan pemrogram membangun hubungan yang tepat antara dokumentasi perancangan dan perangkat lunak [27]. Terdapat 3 elemen penting di *class diagram* yaitu kelas, atribut, dan metode suatu sistem. Untuk penjelasan detail dapat dilihat pada Tabel 2.4.

Tabel 2. 4 *Class Diagram*


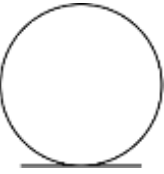
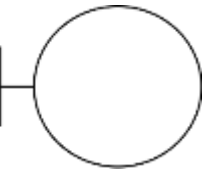
No	Simbol	Keterangan
1.		<i>Class</i> : Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
2.		<i>Association</i> : relasi antar kelas dengan makna yang umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
3.		<i>Directed association</i> : relasi antar kelas dengan makna kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		Generalisasi : relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
5.		<i>Dependency</i> : relasi antar kelas dengan makna ketergantungan antar kelas.




6.		<i>Aggregation</i> : relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).
----	---	--

d. *Sequence Diagram*

Sequence diagram adalah diagram interaksi yang diurutkan secara kronologis (urutan waktu). Setiap diagram sekuensial menggambarkan satu *flow* dari beberapa *flow* di dalam *use case* [28]. Sehingga semakin banyak *use case* yang didefinisikan, maka *sequence diagram* yang harus dibuat juga semakin banyak. Adapun komponen-komponen pada *sequence diagram* dapat dilihat pada Tabel 2.5.

Tabel 2. 5 *Sequence diagram*

No	Simbol	Keterangan
1.		Aktor : Orang, proses, atau sistem lain yang berinteraksi dengan sistem.
2.		<i>Entity Class</i> : Gambaran sistem sebagai landasan dalam Menyusun basis data.
3.		<i>Boundary Class</i> : Menangani komunikasi antar lingkungan sistem.

4.		<i>Control Class</i> : Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika.
5.		<i>Activation</i> : Mewakili proses durasi aktivasi sebuah operasi
6.		<i>Message</i> : Menggambarkan pengiriman pesan.

11. *Black Box Testing*

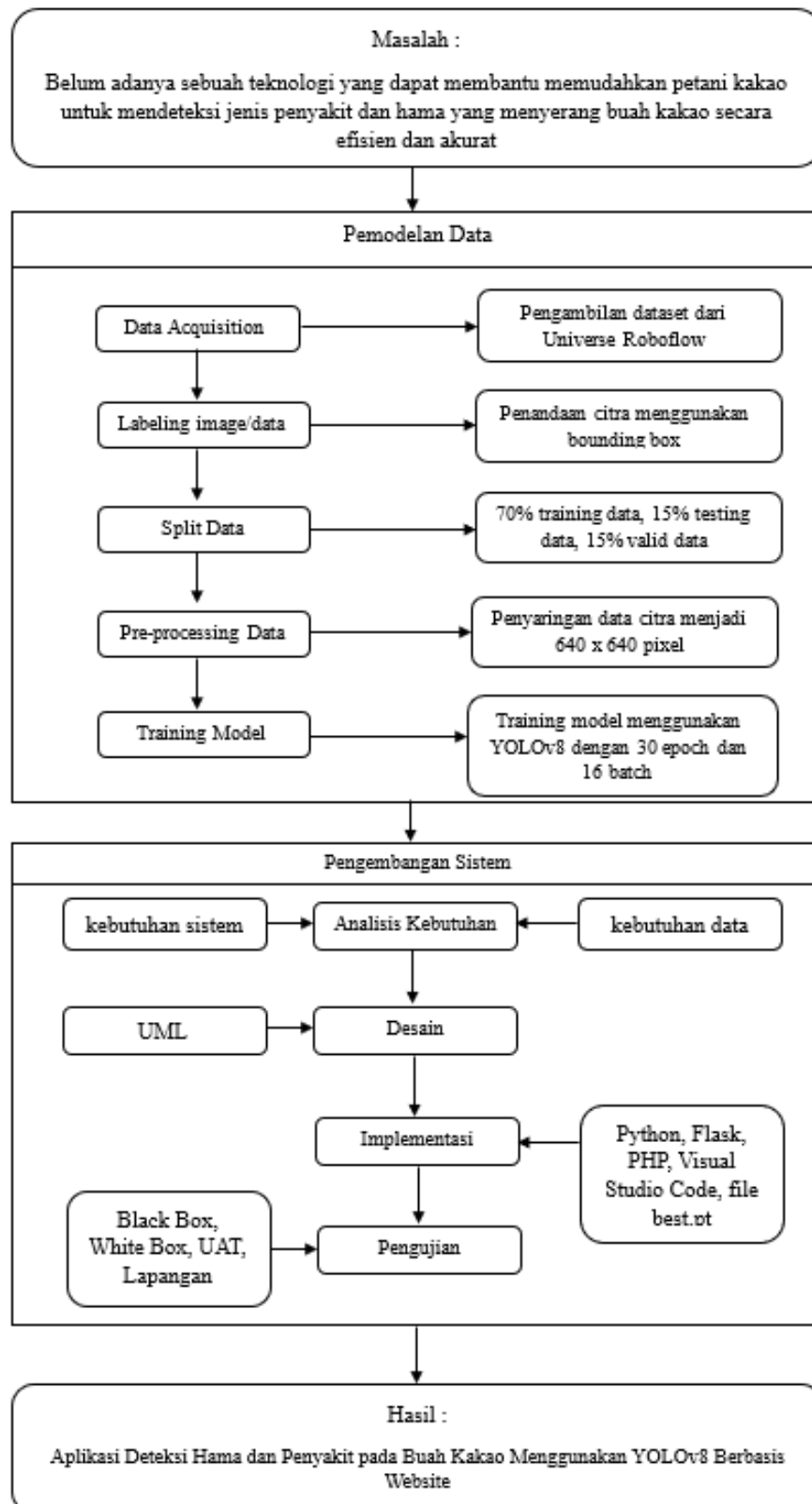
Metode *Blackbox Testing* adalah pendekatan pengujian perangkat lunak yang tidak menentukan detail perangkat lunak. Dalam pengujian ini, hanya memeriksa hasil output berdasarkan nilai input yang diberikan, tanpa mencoba menentukan kode program yang digunakan untuk menghasilkan *output*. Proses *Black Box Testing* dilakukan dengan mencoba program yang telah dibuat dan menguji setiap formulir dengan memasukkan data. Tujuan dari pengujian ini adalah untuk mengkonfirmasi bahwa program tersebut memenuhi kebutuhan perusahaan. Pengujian *Black Box* berusaha untuk memastikan bahwa setiap proses memenuhi persyaratan yang diinginkan. Penguji dapat memahami kumpulan kondisi input dan melakukan pemeriksaan fungsi sistem secara khusus. Oleh karena itu, pengujian menjadi metode pelaksanaan program untuk mengidentifikasi kesalahan atau error, dan kemudian melakukan perbaikan sehingga sistem dapat dianggap sebagai layak digunakan [29].

12. *User Acceptance Testing*

User Acceptance Testing (UAT) adalah pengujian yang dilakukan oleh pengguna akhir dimana pengguna tersebut merupakan staf/karyawan perusahaan yang secara langsung berinteraksi dengan sistem dan memastikan bahwa fungsi-fungsi yang ada telah berjalan sesuai dengan kebutuhannya. *Acceptance testing* merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian *black box* untuk memvalidasi sistem terhadap spesifikasinya. Pengguna akhir bertanggung jawab untuk memverifikasi bahwa semua fungsionalitas yang berlaku telah diuji. Selama UAT, perangkat lunak diuji untuk memastikan bahwa tugas-tugasnya sesuai dengan standar. UAT adalah salah satu prosedur proyek perangkat lunak terakhir dan paling penting yang harus diselesaikan sebelum program dibuat dan dirilis ke pasar [30].

C. Kerangka Berpikir

Kerangka berpikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting. Dalam konteks ini, kerangka berpikir berfungsi sebagai landasan yang mendasari pemahaman yang lebih luas, menjadi dasar bagi pemikiran dan proses penelitian yang akan dilakukan. Untuk mengetahui lebih jelas Aplikasi Deteksi Hama dan Penyakit pada Buah Kakao Menggunakan YOLOv8 Berbasis Website dapat digambarkan melalui kerangka berpikir sebagai berikut:



Gambar 2. 2 Kerangka Berpikir

BAB III

METODE PENELITIAN

A. Pendekatan Penelitian

Pendekatan penelitian adalah sebuah rencana dan prosedur penelitian yang mencakup prosedur yang didasarkan pada asumsi umum untuk menentukan metode dalam pengumpulan dan analisis data. Pendekatan penelitian memiliki manfaat dalam memfasilitasi atau membantu dalam menjawab atau menentukan bagaimana rumusan masalah. Dalam penelitian ini, pendekatan penelitian yang digunakan oleh penulis yaitu menggunakan metode *waterfall*. *Waterfall* digunakan untuk pengembangan sistem dalam penelitian ini karena mengambil pendekatan seperti air terjun yang sistematis dan terstruktur dengan jelas, dimulai dengan kebutuhan sistem dan berkembang melalui analisis, desain, pengkodean, pengujian, dan pemeliharaan. Dinamakan air terjun karena ibarat air terjun yang jatuh dari satu tahap ke tahap berikutnya, menyelesaikannya secara berurutan.

B. Fokus Penelitian

Fokus penelitian difokuskan pada pembahasan sebagai berikut:

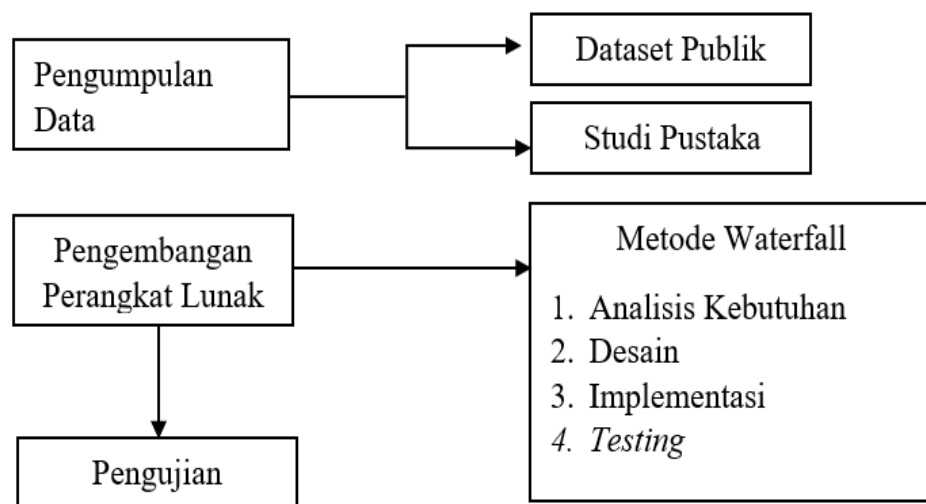
1. Aplikasi berjalan dengan baik dan berbasis website
2. Aplikasi dapat mengidentifikasi penyakit dan hama pada buah kakao berdasarkan foto/video yang diambil.
3. Target pengguna aplikasi adalah masyarakat umum khususnya petani buah kakao.

C. Desain Penelitian

Desain penelitian merupakan langkah awal yang sangat penting dalam proses penelitian, karena memberikan kerangka kerja yang jelas bagi peneliti untuk mengarahkan proses penelitian. Dalam melakukan penelitian terdapat

skema desain penelitian yang dapat dilihat pada gambar 3.1. Langkah-langkah yang akan dilakukan untuk melakukan penelitian tugas akhir ini adalah sebagai berikut:

1. Tahap analisis, pada tahap ini penulis mengumpulkan persyaratan untuk pengembangan sistem yang digunakan untuk penelitian. Pada tahap ini, peneliti akan mengidentifikasi jenis data yang diperlukan dengan memperoleh data yang tepat.
2. Tahap perancangan, pada tahap ini penulis merancang arsitektur, desain, dan spesifikasi teknis software. Perancangan juga melibatkan pembuatan diagram alir dan desain antarmuka pengguna.
3. Tahap Pengembangan/Implementasi, tahap ini penulis melakukan pembuatan kode program sistem sesuai dengan tahap perancangan. Sistem dikembangkan dalam program kecil yang disebut unit, setelah itu unit-unit tersebut diintegrasikan.
4. Tahap pengujian, tahap ini dilakukan untuk mengetahui perangkat lunak yang dibuat telah sesuai dengan desainnya dan fungsi pada perangkat lunak terdapat kesalahan atau tidak. Pada tahap ini diserahkan kepada tim pengujian untuk memeriksa programnya.



Gambar 3. 1 Skema Desain Penelitian

D. Teknik Pengumpulan Data

1. Dataset Publik

Dalam metode ini, peneliti mengumpulkan data melalui pencarian citra (gambar) di internet. Dataset yang digunakan dalam penelitian ini diperoleh dari Universe Roboflow. Penulis mencari beberapa referensi dataset yang relevan dengan jenis penyakit dan hama pada buah kakao. adapun beberapa referensi dataset dari universe roboflow tersebut adalah sebagai berikut:

- a) Dataset “*Cocoa Detection*” oleh andisyamaswandi.

(<https://universe.roboflow.com/andisyamaswandi/cocoa-detection-oy muj>) total gambar pada dataset ini berjumlah 600 gambar. Terbagi menjadi tiga kelas, yaitu kelas Buah Normal, Penggerek Buah, dan Penghisap Buah yang masing-masing kelas tersebut berisi 200 gambar.

- b) Dataset “Tugas Akhir Hama Penyakit Kakao”

(<https://universe.roboflow.com/itera-zgh11/tugas-akhir-hama-penyakit-kakao>) Pada dataset ini berisi sebanyak empat kelas, yaitu kelas Busuk Buah, *Helopeltis*, Kakao Sehat, dan Penggerek Buah Kakao. Namun, pada penelitian ini penulis hanya mengambil kelas Busuk Buah pada dataset tersebut yang berisikan 500 gambar. Namun, penulis hanya mengambil 200 gambar dari kelas Buah Busuk tersebut.

- c) Dataset “*Cacao*”

(<https://universe.roboflow.com/cacao-yeh8i/cacao-w7dav>)

Dataset ini berisi sebanyak tiga kelas, yaitu kelas *Disease*, *Healthy*, dan *Infested*. Namun, Pada penelitian ini penulis hanya mengambil kelas *Healthy* saja yang berisikan 59 gambar.

Setelah mendapatkan referensi dataset diatas, dilakukan kloning kelas dataset yang dipilih tersebut ke dalam projek Skripsi penulis di roboflow untuk kemudian dilakukan *manage class* kembali. Penulis membagi projek menjadi empat kelas, yaitu Buah Sehat, Busuk Buah, Penggerek Buah, dan Penghisap Buah. Kemudian, penulis melakukan

menyimpan dataset tersebut dan membagi dataset citra ke dalam *training data* sebesar 70%, *validation data* sebesar 15%, dan *test data* sebesar 15%, yang akan digunakan untuk melatih model deteksi objek YOLOv8 di *platform* Google Colaboratory.

2. Studi Pustaka

Teknik ini melibatkan pencarian, pengumpulan, analisis literatur atau karya ilmiah yang relevan dengan topik penelitian yang sedang diteliti. Pada Teknik ini membantu peneliti untuk memperoleh wawasan yang mendalam tentang topik penelitian dan memperkuat dasar teoritis dari penelitian terdahulu. Kegiatan meliputi:

- a. Menggunakan buku yang berkaitan dengan penelitian.
- b. Menggunakan jurnal sebagai bahan penelitian terkait.
- c. Menggunakan *website*/internet guna mencari informasi terkait.

E. Teknik Analisis Data

Teknik analisis data yang telah dilakukan setelah semua data yang dibutuhkan terkumpul. Berikut adalah Teknik analisis data yang dilakukan dalam penelitian ini adalah:

1. Akuisisi Data: Mengumpulkan keseluruhan data citra yang diperlukan dengan cara mengunduh dari website penyedia data yaitu Universe Roboflow.
2. Analisis Kebutuhan Pengguna: Melakukan analisis kebutuhan pengguna yang diperlukan untuk pembuatan aplikasi.
3. Pengolahan dan Pemrosesan Data: Mengolah data yang sudah terkumpul, kemudian diproses dengan metode atau algoritma yang sudah ditentukan menjadi suatu aplikasi yang sudah direncanakan.
4. Penyimpulan: Membuat simpulan akhir yang menggambarkan hasil analisis dan temuan yang didapat dari proses penelitian.

BAB IV

HASIL PENGEMBANGAN DAN PEMBAHASAN

A. Hasil

1. Pemodelan Data

Terdapat beberapa tahapan yang dilakukan untuk melakukan pemodelan data, tahapan tersebut dapat dijelaskan sebagai berikut:

A. *Data Acquisition*

Dalam penelitian ini, pengumpulan data dilakukan dengan pencarian dataset yang diunduh dari situs web Universe Roboflow dengan judul dataset "*Cocoa Detection*", "Tugas Akhir Hama Penyakit Kakao", dan "*Cacao*". Dalam beberapa dataset tersebut, penulis mengambil beberapa kelas data yang dibutuhkan untuk pembuatan model deteksi penyakit dan hama pada buah kakao. Data yang telah terkumpul, kemudian dijadikan empat kelas dataset yang digunakan dalam penelitian, yaitu kelas busuk buah, penggerek buah, penghisap buah, dan buah sehat. Contoh dari beberapa kelas dataset ditampilkan dalam Gambar 4.1.



Gambar 4. 1 Dataset Hama dan Penyakit Buah Kakao

1) Penyakit Busuk Buah Kakao

Busuk buah kakao (BBK) adalah penyakit yang umum menyerang kakao. Produksi dapat berkurang 20-40% karena penyakit ini, dan di daerah dengan kelembaban dan curah hujan yang tinggi,

kerugian dapat mencapai 100%. *Phytophthora palmivora* adalah jamur yang menyebabkan busuk buah kakao. Jamur ini sangat sulit diberantas karena jamur ini dapat hidup di bagian tanaman yang terserang atau di dalam tanah sebagai miselium dan klamidospora[32].

Gejala munculnya bercak coklat kehitaman pada bagian pangkal, tengah, dan ujung buah kakao merupakan tanda busuk buah kakao. Setiap umur buah kakao dapat dipengaruhi oleh penyakit ini. Buah yang membusuk tertutupi oleh miselium putih dan warna berubah menjadi hitam dan keras[33].

2) Hama Penggerek Buah Kakao

Hama utama tanaman kakao di Asia Tenggara, khususnya di Kepulauan Melayu (Malaysia, Filipina, Kalimantan, dan Indonesia), adalah hama penggerek buah kakao (PBK), *C cramerella*. Hingga menjelang panen, buah kakao muda akan diserang dan dirusak oleh hama ini. Jika serangga hama lain yang merusak tanaman kakao terdiri dari serangga dewasa dan pra-dewasa, penggerek buah kakao yang merusak buah kakao adalah pra-dewasa yang dikenal sebagai larva[34]. Dengan kata lain Hama penggerek buah kakao (PBK) adalah serangga yang infeksi yang menyerang bagian buah kakao, mulai dari buah muda hingga buah yang sudah tua atau dekat dengan panen.

Secara morfologis, tidak terdapat perbedaan yang dapat diamati secara visual antara buah kakao yang terinfeksi Penggerek Buah Kakao (PBK) dengan yang sehat. Gejala PBK baru muncul saat buah sudah matang. Warna kulit buah kakao yang terkena infeksi cenderung agak jingga atau pucat keputihan, dan terjadi peningkatan bobot buah. Ketika buah diguncang, tidak terdengar suara benturan antara biji dan dinding buah karena adanya lendir dan kotoran yang terbentuk di dalam daging buah serta kerusakan biji. PBK menghasilkan enzim seperti heksokinase, malat dehidrogenase, esterase fluoresen, dan enzim malat polimorfisme yang merusak

dinding buah. Akibat serangan PBK, biji kakao menjadi keriput dan kulit biji menggelap. Serangan hama PBK menyebabkan cita rasa biji kakao Indonesia menurun, kadar kotoran meningkat, dan terjadi kontaminasi bakteri yang lebih banyak [35].

3) Hama Kepik Penghisap Buah Kakao (*Helopeltis spp.*)

Helopeltis spp. merupakan salah satu hama utama tanaman kakao. Serangga ini menyebabkan kerusakan dengan cara melukai dan menghisap sari buah kakao dan tunas muda. Serangan pada buah muda menyebabkan kematian buah dan pada buah setengah tua dapat menyebabkan pembentukan buah tidak normal. Menyusul serangan *Helopeltis spp.* yang seringkali berdampak langsung pada kuantitas atau kualitas biji kakao[36].

Ciri serangan hama kepik penghisap buah kakao diantaranya kulit buah kakao terdapat bercak-bercak hitam (kecoklatan) dan kering, pertumbuhan buah terhambat, buah kaku dan sangat keras serta bentuknya mengkerut dan buah kecil. Pada buah muda sangat rentan terhadap serangan hama kepik penghisap buah kakao[37].

4) Buah Kakao Sehat

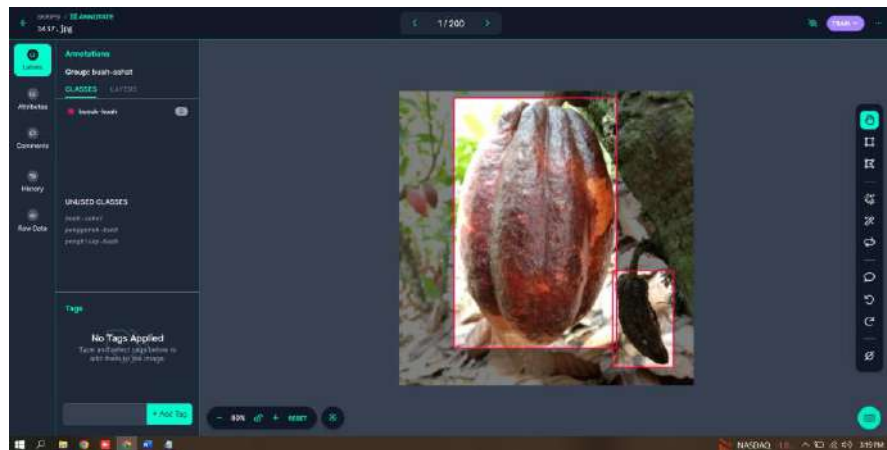
Buah kakao yang sehat merupakan buah kakao yang memiliki kualitas yang baik, dimana buah kakao yang sehat seharusnya memiliki bentuk yang baik dan ukuran yang relatif seragam. Mereka tidak boleh terlalu kecil atau cacat secara fisik seperti membusuk atau berlubang. Buah kakao yang sehat biasanya memiliki aroma yang khas dan tidak ada bau yang tidak biasa atau busuk. Selain itu, Pohon kakao yang sehat biasanya tidak menunjukkan tanda-tanda serangan hama seperti sarang ulat atau telur-telur serangga. Serangan hama dapat mengakibatkan kerusakan pada buah atau daun.

Buah kakao yang sehat dan siap panen memiliki ciri terjadi perubahan warna. Pada buah yang sewaktu belum masak berwarna hijau, maka pada saat masak akan berubah menjadi kuning. Sementara

itu, pada buah yang sewaktu belum masak berwarna merah keunguan, maka sewaktu masak akan berubah menjadi jingga.

B. Labelling Data

Proses *Labelling Data* dilaksanakan melalui penandaan citra menggunakan kotak pembatas (*bounding box*), sambil menetapkan kelas yang relevan pada setiap citra. Metode *labelling data* dilakukan dengan menggunakan website Roboflow pada *tools annotation*, di mana setiap citra diberikan kotak pembatas untuk menandai area yang signifikan untuk kemudian masing-masing diberi label yang menunjukkan informasi mengenai *bounding box* tersebut. Proses ini memungkinkan pengayaan dataset dengan informasi yang diperlukan untuk pelatihan dan penilaian model dalam analisis lanjutan citra. Proses labelling data dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Annotation dan Labelling Data

C. Split Data

Pada tahap pengelompokan data, dataset disegmentasikan menjadi tiga bagian, yakni dataset pelatihan (*training*) dataset validasi (*validation*) dan dataset pengujian (*testing*). Dataset pelatihan merangkap sebanyak 70% dari total keseluruhan dataset, sementara dataset validasi terdiri dari 15% dari keseluruhan dataset dan dataset pengujian terdiri dari 15% dari keseluruhan dataset. Dataset pelatihan, validasi dan pengujian ini akan dimanfaatkan dalam proses pelatihan dan pembuatan model. Pembagian

ini menjadi penting dalam mengoptimalkan performa model, memastikan bahwa model telah melalui pembelajaran dari sejumlah besar data yang beragam secara memadai, serta mampu diuji dengan menggunakan data yang belum pernah dipertimbangkan sebelumnya. Pada jumlah dataset bisa dilihat pada tabel 4.1.

Tabel 4. 1 Tabel Jumlah dataset tiap kelas

No.	Kelas	Jumlah Image
1.	Busuk buah	200
2.	Penggerek buah	200
3.	Penghisap buah	200
4.	Buah sehat	259

a. *Training Data Folder*

Training data folder digunakan untuk menyimpan data citra yang digunakan untuk melatih model. Untuk pembagian jumlah file citra di *training data folder* pada penelitian ini dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Tabel jumlah *train data*

No.	Kelas	Jumlah data
1.	Busuk buah	141
2.	Penggerek buah	146
3.	Penghisap buah	136
4.	Buah sehat	179

b. *Valid Data Folder*

Validation data folder digunakan untuk menyimpan data yang digunakan untuk mengevaluasi kinerja model saat proses pelatihan. Untuk pembagian jumlah file citra di *validation data folder* pada penelitian ini dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Tabel jumlah *valid data*

No.	Kelas	Jumlah data
1.	Busuk buah	26
2.	Penggerek buah	27
3.	Penghisap buah	32
4.	Buah sehat	44

c. *Test Data Folder*

Test data folder digunakan untuk menyimpan data citra yang digunakan untuk menguji kinerja final model setelah proses pelatihan. Untuk pembagian jumlah *file* citra di test data folder pada penelitian ini dapat dilihat pada Tabel 4.4.

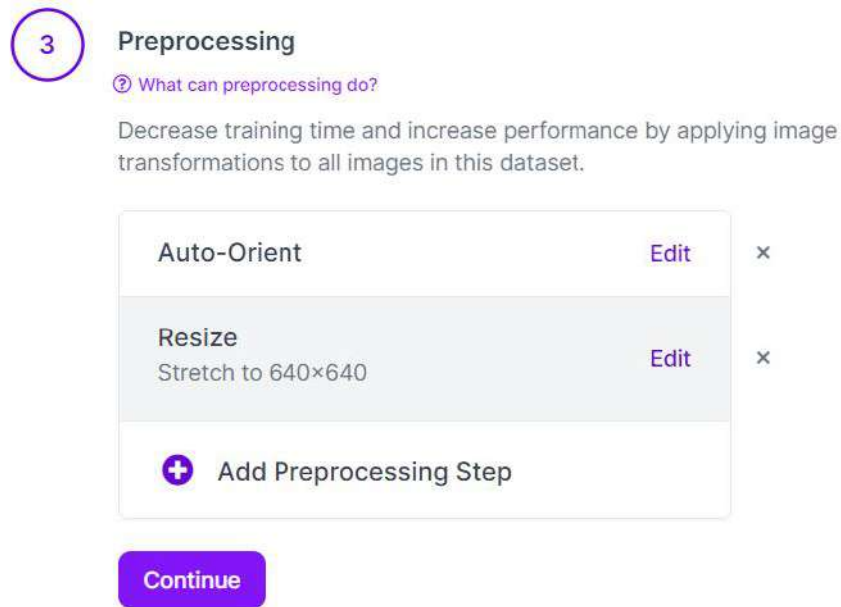
Tabel 4. 4 Tabel jumlah *test data*

No.	Kelas	Jumlah data
1.	Busuk buah	33
2.	Penggerek buah	27
3.	Penghisap buah	32
4.	Buah sehat	36

D. *Preprocessing Data*

Tahap *preprocessing data* (pra-pemrosesan) adalah tahap penting dalam analisis data yang bertujuan untuk membersihkan, mengubah, dan mempersiapkan data mentah agar sesuai untuk analisis lebih lanjut. Tahap *preprocessing* dapat mencakup beberapa langkah, tergantung pada jenis data dan tujuan analisisnya. Pada tahap ini, roboflow otomatis akan menambahkan *Auto-orient* dan *Resize* gambar, namun selain itu pengguna juga dapat menambahkan beberapa *preprocessing step* lainnya, contohnya seperti *Static Crop*, *Dynamic Crop*, *Grayscale*, *Auto-Adjust Contrast*, *Tile*, *Modify Classes*, *Filter Null*, dan *Filter by Tag* adalah fitur yang memungkinkan pengguna untuk memanipulasi data gambar agar sesuai

dengan kebutuhan pelatihan model. Namun, pada tahap ini, dilakukan penyaringan data yang tepat untuk disertakan dalam dataset dan hanya dilakukan penyesuaian ukuran setiap citra menjadi 640 x 640 pixel serta *auto-orient* secara otomatis oleh roboflow. Proses *preprocessing data* pada roboflow dapat dilihat pada Gambar 4.3.



Gambar 4. 3 *Preprocessing data*

E. *Training Model*

Dataset yang sudah selesai dilakukan proses *generate* data di Roboflow, maka roboflow akan membuat versi dari dataset tersebut. Versi yang sudah dibuat kemudian *diexport* untuk dilanjutkan ke tahap selanjutnya untuk dilakukan proses custom training dengan memilih format model YOLO v8. *Custom train model* dilakukan menggunakan *platform* Google Colaboratory dengan menuliskan *source code* untuk *export* versi dataset yang telah siap dilatih dari roboflow. Model dilatih menggunakan 859 citra yang terbagi menjadi empat kelas, antara lain sebagai berikut: busuk buah, penggerek buah, penghisap buah, dan buah sehat. Proses *training* dilakukan dengan menggunakan model YOLOv8 dengan 30 putaran (*epochs*) dengan pengaturan *batch size* otomatis batch

16 dan dengan nilai *confidence* yang menjadi ambang batas (*threshold*) sebesar 0,25. *Output* dari proses *training* ini adalah model dari YOLOv8 berformat .pt. Berikut adalah *output* dari proses *training model* dapat dilihat pada Gambar 4.4.

```
Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 3006428 parameters, 0 gradients, 8.1 GFLOPs

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
all	129	173	0.939	0.926	0.956	0.915
buah-sehat	129	82	0.9	0.707	0.842	0.711
busuk-buah	129	32	0.892	1	0.991	0.965
penggerek-buah	129	27	0.964	0.995	0.994	0.99
penghisap-buah	129	32	1	1	0.995	0.995

```
Speed: 0.4ms preprocess, 3.9ms inference, 0.0ms loss, 6.0ms postprocess per image
Results saved to runs/detect/train
Learn more at https://docs.ultralytics.com/modes/train
```

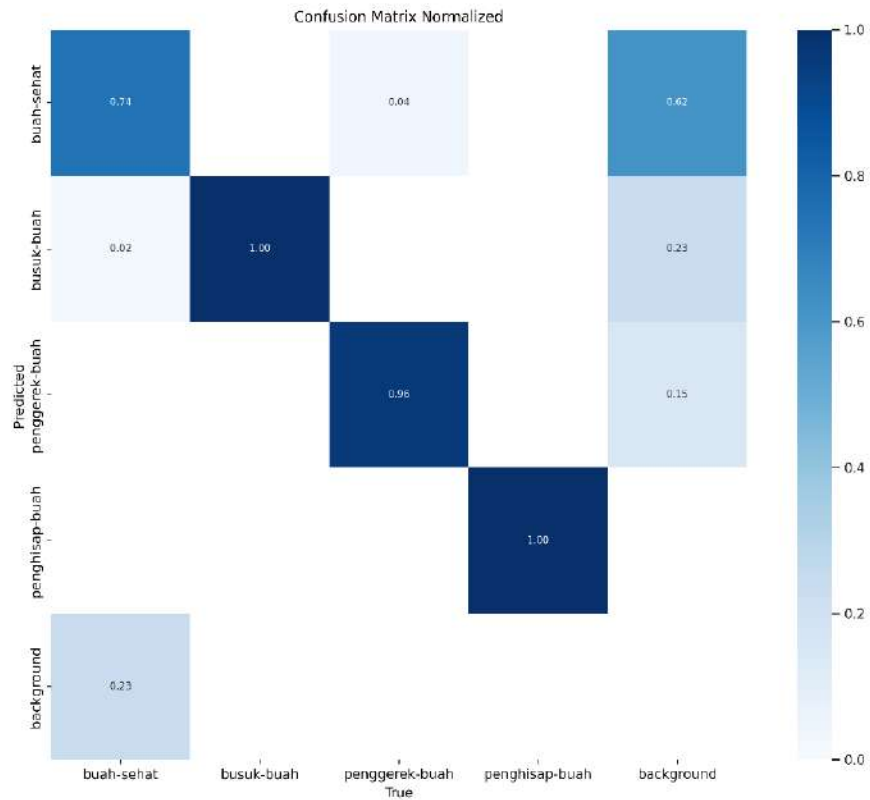
Gambar 4. 4 *Training Model*

F. *Evaluation Model*

Setelah menyelesaikan proses pelatihan model, langkah berikutnya adalah melakukan evaluasi model. Evaluasi model bertujuan untuk menilai seberapa baik kemampuan model dalam menggeneralisasi pola yang telah dipelajari dari data pelatihan terhadap data baru yang tidak pernah dilihat sebelumnya.

1) *Confusion Matrix*

Confusion Matrix adalah alat evaluasi yang digunakan untuk mengukur kinerja deteksi objek dengan membandingkan hasil prediksi model terhadap label yang sebenarnya pada dataset validasi. Matriks ini menampilkan jumlah prediksi yang benar (*true positives*), jumlah prediksi yang salah (*false positives*), serta objek yang gagal terdeteksi (*false negatives*) untuk setiap kelas objek yang diprediksi. Dengan menganalisis matriks kebingungan, pengembang dapat memperoleh wawasan yang lebih mendalam tentang kelemahan dan kekuatan model dalam mengidentifikasi berbagai kelas objek. Hasil dari *Confusion Matrix* ini dapat dilihat pada Gambar 4.5.



Gambar 4. 5 *Confusion matrix*

Untuk menghitung akurasi *Confusion Matrix* dilakukan dengan menerapkan perhitungan persamaan 1 pada rumus berikut ini:

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 4. 6 Rumus *Confusion Matrix*

Keterangan:

True Positive (TP) : Sampel data positif, prediksi negatif

True Negative (TN) : Sampel data negatif, prediksi positif

False Positive (FP) : Sampel data negaif, prediksi positif

False Negatif (FN) : Sampel data positif, prediksi negatif

Rumus:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$\text{Accuracy} = \frac{74+100+96+100}{23+74+100+96+100} \times 100\%$$

$$\text{Accuracy} = \frac{370}{393} \times 100\%$$

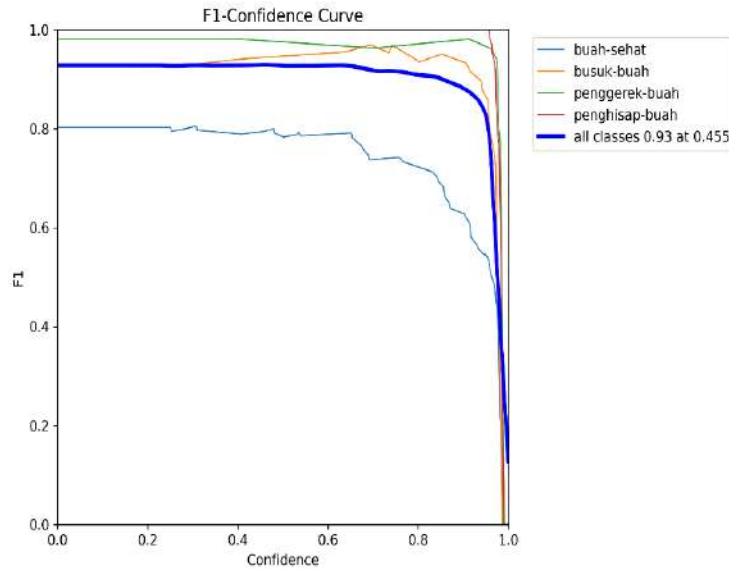
$$= 0,94 \times 100\% = 94\%$$

Hasil akurasi yang diperoleh untuk seluruh data dalam proses pelatihan model dengan 30 *epoch* adalah 94%. Akurasi dianggap baik jika mencapai 76%-100%, cukup baik jika berada di kisaran 55%-75%, kurang baik jika berada di kisaran 40% - 56%, dan tidak baik jika di bawah 40%. Hal ini menunjukkan bahwa model sangat baik dan akurat dalam mendeteksi jenis hama dan penyakit buah kakao.

2) *F1-Confidence Curve*

F1-Confidence Curve adalah representasi grafis yang mengilustrasikan korelasi antara *F1-score* (sebuah *metrik* yang mengukur keseimbangan antara presisi dan *recall*) dengan tingkat kepercayaan (*confidence level*) dari model klasifikasi. Kurva ini memberikan gambaran tentang sejauh mana model mampu menjaga keseimbangan antara presisi dan *recall* pada berbagai tingkat kepercayaan. Dengan mengamati *F1-Confidence Curve*, pengguna dapat memilih ambang (*threshold*) yang tepat untuk menyesuaikan

trade-off antara *presisi* dan *recall* sesuai dengan kebutuhan khusus dari aplikasi atau skenario penggunaan tertentu. Hasil dari *F1-Confidence Curve* ini tersedia dalam Gambar 4.7.



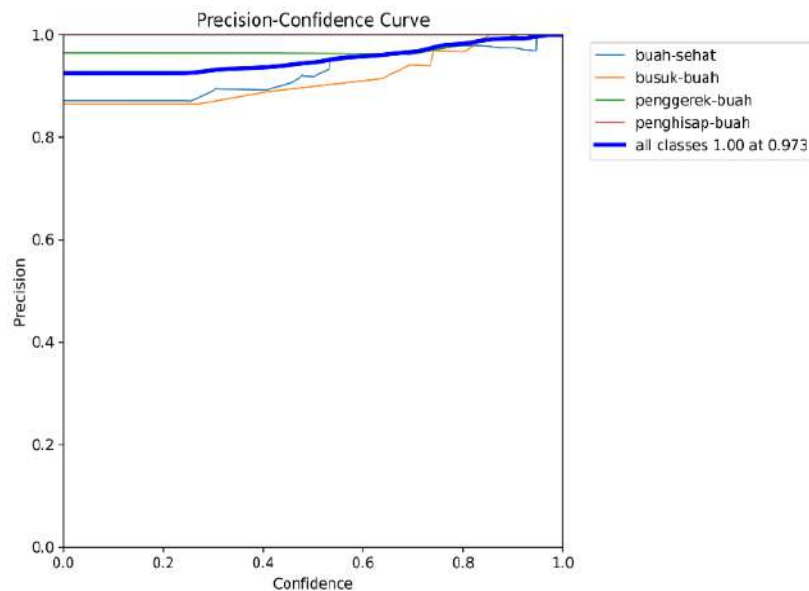
Gambar 4. 7 *F1-Confidence Curve*

Grafik F1 yang dihasilkan dari proses pelatihan model YOLO v8 dengan 30 *epoch* menunjukkan bahwa skor F1 mencapai nilai rata-rata tertinggi sebesar 0,93 pada nilai *confidence* 0,455. Grafik ini memperlihatkan hubungan antara nilai *confidence* dan skor F1, yang mencerminkan kemampuan model dalam menyeimbangkan *presisi* dan *recall* pada berbagai tingkat keyakinan dalam prediksi. Nilai *confidence* 0,455 merupakan titik di mana model mencapai keseimbangan optimal antara *presisi* dan *recall*, yang menghasilkan skor F1 tertinggi.

3) *Precision-Confidence Curve*

Precision-Confidence Curve adalah alat visualisasi yang menggambarkan hubungan antara tingkat *presisi* dan tingkat kepercayaan (*confidence level*) dari model klasifikasi. Kurva ini memberikan pemahaman tentang bagaimana *presisi* model berubah seiring dengan perubahan tingkat kepercayaan. Dengan

menganalisis *Precision-Confidence Curve*, pengguna dapat memahami *trade-off* antara tingkat presisi dan tingkat kepercayaan dari model, serta memilih ambang (*threshold*) yang sesuai untuk memenuhi kebutuhan spesifik dari aplikasi atau kasus penggunaan tertentu. Hasil dari *Precision-Confidence Curve* ini tersedia dalam Gambar 4.8.



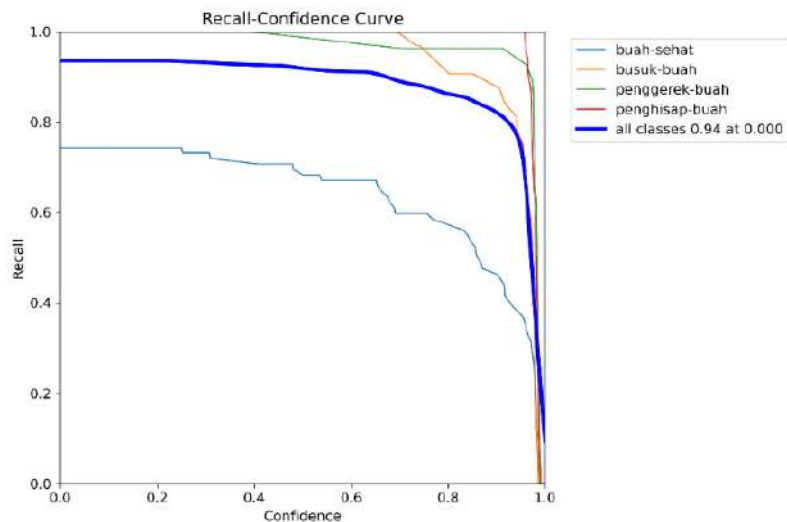
Gambar 4. 8 *Precision-Confidence Curve*

Grafik di atas memvisualisasikan nilai *precision* terhadap nilai *confidence*. Dari hasil pelatihan yang diperoleh, ditemukan bahwa nilai *precision* mencapai rata-rata maksimal sebesar 1,00 pada nilai *confidence* 0,973. Hal ini menandakan bahwa pada titik *confidence* tersebut, model mencapai tingkat presisi tertinggi, di mana semua prediksi yang dibuat oleh model adalah benar tanpa adanya kesalahan positif.

4) *Recall-Confidence Curve*

Recall-Confidence Curve adalah representasi grafis yang menunjukkan hubungan antara tingkat *recall* (seberapa banyak dari semua kelas positif yang berhasil diprediksi oleh model) dan

tingkat kepercayaan (*confidence level*) dari model klasifikasi. Kurva ini membantu dalam memahami bagaimana tingkat *recall* model berubah seiring dengan perubahan tingkat kepercayaan. Dengan menganalisis *Recall-Confidence Curve*, pengguna dapat mengevaluasi seberapa baik model dapat mengidentifikasi kelas positif pada berbagai tingkat kepercayaan, dan memilih ambang (*threshold*) yang sesuai untuk mencapai tingkat *recall* yang diinginkan berdasarkan kebutuhan spesifik dari aplikasi atau kasus penggunaan tertentu. Hasil dari *Recall-Confidence Curve* ini tersedia dalam Gambar 4.9.

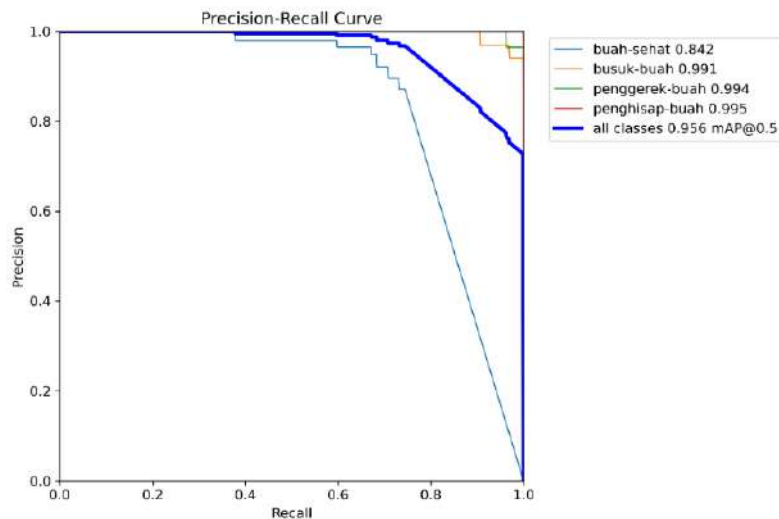


Gambar 4. 9 *Recall-Confidence Curve*

Grafik yang ditampilkan menggambarkan nilai *recall* terhadap nilai *confidence*. Dari hasil pelatihan yang diperoleh, didapati bahwa nilai *recall* mencapai rata-rata maksimal sebesar 0,94 pada nilai *confidence* 0,00. Hal ini menunjukkan bahwa pada tingkat keyakinan yang sangat rendah (0,00), model berhasil mengingat kembali sebagian besar instance positif yang sebenarnya.

5) Precision-Recall Curve

Precision-Recall Curve adalah sebuah grafik yang mengilustrasikan hubungan antara presisi (tingkat proporsi prediksi positif yang benar) dan *recall* (tingkat proporsi kelas positif yang berhasil diprediksi) pada berbagai *threshold* pengambilan keputusan. Kurva ini memberikan gambaran tentang *trade-off* antara presisi dan *recall* yang dihasilkan oleh model klasifikasi. Dengan menganalisis *Precision-Recall Curve*, pengguna dapat memilih *threshold* yang sesuai untuk mencapai keseimbangan yang optimal antara presisi dan *recall*, sesuai dengan prioritas dan kebutuhan spesifik dari aplikasi atau kasus penggunaan tertentu. Hasil dari *Precision-Recall Curve* ini tersedia dalam Gambar 4.10.



Gambar 4. 10 *Precision-Recall Curve*

Grafik yang ditampilkan menggambarkan nilai *Precision-Recall* terhadap nilai *confidence*. Dari hasil pelatihan yang diperoleh, nilai *Precision-Recall* mencapai rata-rata maksimal sebesar 0,956 pada nilai mAP 0,5. Semakin tinggi skor mAP, semakin baik performa yang dihasilkan oleh model, menunjukkan bahwa model memiliki keseimbangan yang baik

antara *presisi* dan *recall* pada tingkat kepercayaan yang ditunjukkan.

Skor mAP, atau "*mean Average Precision*", merupakan metrik evaluasi umum dalam deteksi objek di bidang *computer vision*. Metrik ini mengukur kemampuan suatu model dalam mengidentifikasi dan mengklasifikasikan objek dalam gambar dengan akurasi tinggi. Proses perhitungan mAP melibatkan perhitungan nilai *precision* dan *recall* untuk setiap kelas objek yang diidentifikasi, diikuti dengan perhitungan *average precision* (AP) untuk setiap kelas, dan akhirnya mengambil rata-rata dari semua nilai AP untuk mendapatkan skor mAP keseluruhan. Skor mAP memberikan gambaran tentang seberapa baik model dapat menemukan objek yang relevan dalam gambar dan seberapa akurat model dalam mengklasifikasikan objek tersebut.

G. *Test Model*

Model yang sudah jadi kemudian di dicoba untuk mendeteksi menggunakan gambar dari luar dataset dari semua class class untuk mengujinya. Menggunakan tingkat *condifedence* score 0.25 model hanya akan menampilkan *output* dengan skor ≥ 0.25 Berikut adalah beberapa dari hasil-hasilnya:

- 1) Buah Sehat



Gambar 4. 11 *Test model* kelas buah sehat

Test model yang pertama dilakukan yaitu menguji model dengan gambar buah sehat dari luar dataset yang digunakan. Pada pengujian ini model berhasil mendeteksi buah kakao yang masuk ke dalam kategori buah sehat pada gambar. Hasil deteksi didapatkan buah sehat 0,78, yang mana ini memberikan informasi bahwa hasil deteksi tersebut memiliki tingkat kepercayaan 0,78 keakuratan hasil deteksi dari batas nilai *confidence* yang ditentukan sebesar 0,25. Sehingga hasil tersebut dapat dipercayai menjadi suatu hasil yang benar dan akurat.

2) Busuk Buah



Gambar 4. 12 *Test model* kelas busuk buah

Test model yang kedua yaitu dilakukan untuk menguji model dengan gambar busuk buah dari luar dataset yang digunakan. Pada pengujian ini model berhasil mendeteksi busuk buah pada buah kakao dalam gambar. Hasil deteksi didapatkan busuk buah 0,98, busuk buah 0,45 dan ada satu buah kakao yang tidak terdeteksi. Ini memberikan informasi bahwa hasil deteksi tersebut memiliki tingkat kepercayaan 0,98, 0,45, dan dibawah 0,25 keakuratan hasil deteksi dari batas nilai *confidence* yang ditentukan sebesar 0,25. Perbedaan dari nilai *confidence* tersebut

dapat dipengaruhi oleh adanya kualitas gambar yang digunakan, tingkat pencahayaan, ataupun tingkat ketajaman atau kejelasan pada gambar sehingga dapat membuat tingkat kepercayaan hasil deteksi yang ditampilkan lebih rendah dibandingkan dengan tingkat kepercayaan yang dihasilkan pada gambar yang memiliki kualitas gambar lebih baik. Adapun gambar yang tidak dapat terdeteksi itu karena hasil dari tingkat kepercayaan dibawah 0,25 sehingga gambar otomatis diabaikan untuk ditampilkan hasil deteksinya.

3) Penggerek Buah

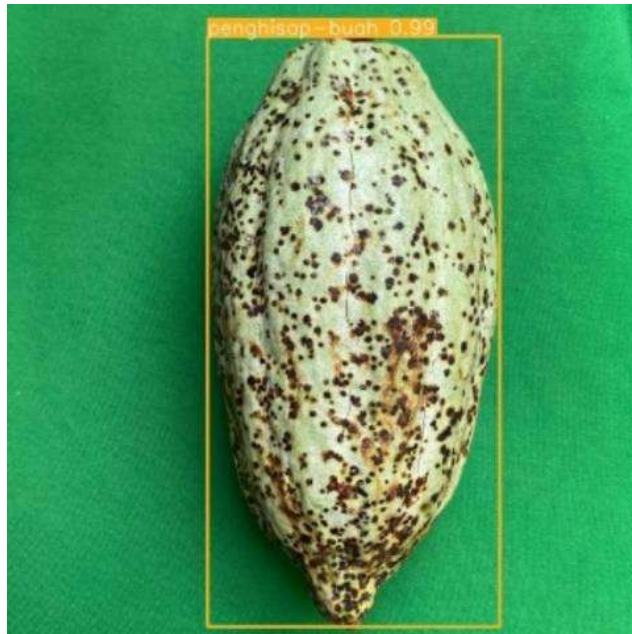


Gambar 4. 13 *Test model* kelas penggerek buah

Test model ketiga dilakukan untuk menguji model dengan gambar buah sehat dari luar dataset yang digunakan. Pada pengujian ini model berhasil mendeteksi buah kakao yang masuk ke dalam kategori penggerek buah kakao pada gambar. Hasil deteksi didapatkan penggerek buah 0,98, yang mana ini

memberikan informasi bahwa hasil deteksi tersebut memiliki tingkat kepercayaan 0,98 keakuratan hasil deteksi dari batas nilai *confidence* yang ditentukan sebesar 0,25. Sehingga hasil tersebut dapat dipercayai menjadi suatu hasil yang benar dan akurat.

4) Penghisap Buah



Gambar 4. 14 *Test model* kelas penghisap buah

Test model keempat dilakukan dengan menguji model menggunakan gambar penghisap buah kakao dari luar dataset yang digunakan. Pada pengujian ini model berhasil mendeteksi buah kakao yang masuk ke dalam kategori jenis hama penghisap buah pada gambar. Hasil deteksi didapatkan penghisap buah 0,99, yang mana ini memberikan informasi bahwa hasil deteksi tersebut memiliki tingkat kepercayaan 0,99 keakuratan hasil deteksi dari batas nilai *confidence* yang ditentukan sebesar 0,25. Sehingga hasil tersebut dapat dipercayai menjadi suatu hasil yang benar dan akurat.

H. Upload Model

Langkah *upload* model adalah titik penting di mana model yang telah diperbaharui dengan pengetahuan dari dataset pelatihan diimplementasikan ke dalam solusi yang nyata. Setelah selesai melakukan pelatihan dengan dataset yang sesuai, model YOLOv8 akan menghasilkan file bobot atau *weight* yang memuat informasi tentang parameter yang telah disesuaikan selama proses pelatihan. sehingga kita harus menyimpan file *weights* tersebut dalam bentuk file *best.pt*.

Setelah itu kita perlu *upload* model ke *platform* yang dituju. Pada tahap ini, model yang telah dilatih diunggah ke *platform* roboflow dengan menuliskan source code *upload* model dari dokumentasi yang sudah disediakan oleh roboflow. berikut

```
[ ] project.version(4).deploy(model_type="yolov8", model_path="/content/runs/detect/train")  
🔗 view the status of your deployment at: https://app.roboflow.com/aldhitaproject-d4fju/skripsi-5qawc/4  
Share your model with the world at: https://universe.roboflow.com/aldhitaproject-d4fju/skripsi-5qawc/model/4
```

Gambar 4. 15 Source code Upload model training

merupakan source code *upload* model dapat dilihat pada gambar berikut.

2. Pengembangan Sitem

Dalam pengembangan aplikasi penulis menggunakan metode *waterfall*, dimana tahapan pengembangannya diatur sesuai urutan yang telah ditentukan dalam metode tersebut.

a. Analisis Kebutuhan

1) Analisis Kebutuhan Data

Data yang diperlukan untuk penelitian terdiri dari citra yang digunakan sebagai sampel pelatihan untuk mengembangkan model deteksi citra. Sampel citra tersebut diunduh dari internet melalui *platform* Universe Roboflow, kemudian diproses di Roboflow untuk dilakukan proses *annotasion* dan *generate* dataset. Penulis mengumpulkan dataset jenis penyakit dan hama pada buah kakao

dengan total jumlah dataset yang berhasil dikumpulkan berisi 859 dataset citra. Dalam dataset tersebut, penulis membagi kedalam empat kelas, yaitu kelas Buah Sehat yang berisi 259 gambar, Busuk Buah yang berisi 200 gambar, Penggerek Buah yang berisi 200 gambar, dan Penghisap Buah yang berisi 200 gambar. Dataset tersebut penulis beri nama dalam projek dataset “Skripsi”. Dataset tersebut kemudian dilakukan pembagian data dengan persentase *training data* 70%, *Valid data* 15%, dan *test data* 15%. Projek Penulis dalam dataset Skripsi tersebut dapat diakses melalui alamat URL Berikut <https://universe.roboflow.com/aldhitaproject-d4fju/skripsi-5qawc/model/4>.

Detail pembagian data pada dataset Skripsi tersebut dibagi menjadi tiga folder. Yaitu folder Train, Valid, dan Test, yang mana isi dari masing-masing folder tersebut terbagi menjadi dua folder yaitu folder image dan folder labels. Berikut merupakan tabel data dari dataset Skripsi:

a. *Train data folder*

Train data folder merupakan data yang digunakan untuk menyimpan data citra yang digunakan untuk melatih model. Adapun pembagiannya sebagai berikut:

Tabel 4. 5 *Train data folder*

No	Nama Folder	Jumlah Data
1.	<i>Image</i>	602
2.	<i>Labels</i>	602

b. *Valid data folder*

Valid data folder digunakan untuk menyimpan data yang digunakan untuk mengevaluasi kinerja model. Adapun pembagiannya sebagai berikut:

Tabel 4. 6 *Valid data folder*

No	Nama Folder	Jumlah Data
1.	<i>Image</i>	129
2.	<i>Labels</i>	129

c. *Test data folder*

Test data folder digunakan untuk menyimpan data yang digunakan untuk menguji atau mengevaluasi model setelah pelatihan selesai. Adapun pembagiannya sebagai berikut:

Tabel 4. 7 *Test data folder*

No	Nama Folder	Jumlah Data
1.	<i>Image</i>	128
2.	<i>Labels</i>	128

2) Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahap krusial dalam pengembangan perangkat lunak yang bertujuan untuk memahami, mendokumentasikan, dan mengatur persyaratan yang harus dipenuhi oleh sistem tersebut. mengidentifikasi permasalahan yang akan diatasi atau kebutuhan yang akan dipenuhi oleh sistem yang akan dikembangkan. Dalam konteks penelitian ini, berikut adalah kebutuhan yang akan digunakan oleh peneliti untuk merancang aplikasi:

a. Kebutuhan Perangkat Keras *Computer*

Berikut ini adalah spesifikasi perangkat keras minimum yang mendukung aplikasi klasifikasi jenis sampah terurai dan tidak terurai, yaitu :

Tabel 4. 8 Kebutuhan perangkat keras komputer

RAM	GPU	CPU	Penyimpanan	Resolusi
8GB (minimal)	Nvidia GeForce MX110	Minimal Intel Core i3 atau yang setara	Setidaknya 256 GB SSD untuk kecepatan akses data yang lebih baik.	1280x720 (Minimal)

b. Kebutuhan Perangkat Lunak Komputer

Tabel 4. 9 Kebutuhan perangkat lunak komputer

Sistem Operasi	Aplikasi
Windows 10 / 11	<ul style="list-style-type: none"> - Google Colaboratory - Roboflow - Kaggle - Visual Studio Code - Figma - Chrome Browser - Draw.io - Google Drive - Github

3) Analisis Kebutuhan Fungsional

1. Aplikasi dapat menampilkan halaman utama aplikasi yaitu halaman beranda yang memberikan informasi tentang aplikasi, panduan deteksi, persentasi *model visualize* yang digunakan, informasi terkait dataset, dan informasi grafik yang didapat pada proses *training model*.

2. Aplikasi dapat menampilkan halaman deteksi yang memuat tombol *chosen file* untuk *input* gambar/video dari *file manager user* dan tombol *upload* untuk *mengupload* gambar untuk dilakukan proses deteksi.
3. Aplikasi dapat mendeteksi hasil foto/video yang telah *diupload* dan mengeluarkan *output* hasil deteksi pada tampilan *result* yang ada dalam menu/halaman deteksi.
4. Aplikasi dapat menampilkan halaman artikel dan dapat mengkategorikan topik artikel yang ada.
5. Aplikasi dapat menampilkan halaman tentang untuk memberikan informasi singkat tentang aplikasi dan informasi terkait *developer* (pengembang aplikasi).

b. Desain

Desain sistem ini berfungsi untuk memudahkan proses implementasi atau pengembangan kode dengan memberikan gambaran yang jelas tentang apa yang akan dibuat [31]. Desain yang dibuat oleh penulis bertujuan untuk mengembangkan aplikasi deteksi penyakit dan hama pada buah kakao antara lain :

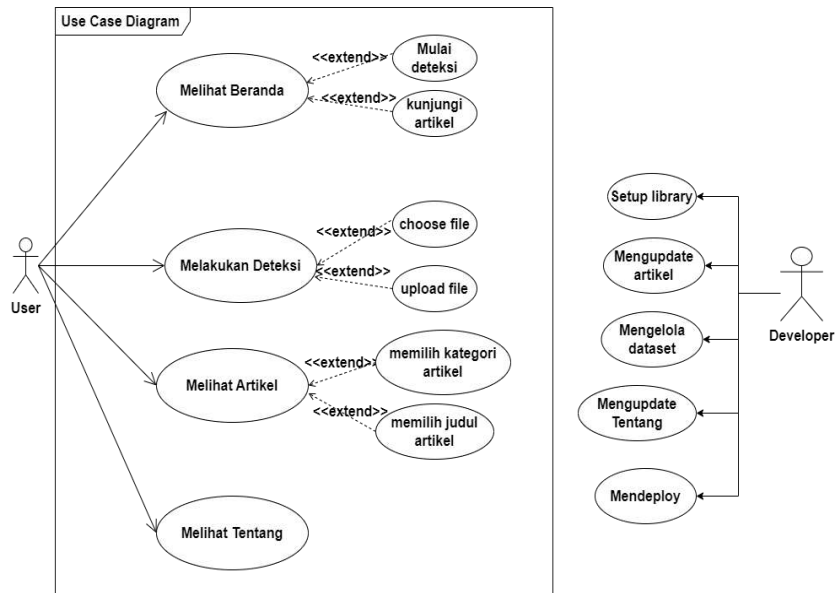
1. Desain *Unified Modeling Language (UML)*

Dalam desain UML, penulis menggunakan berbagai jenis diagram untuk menggambarkan berbagai aspek aplikasi atau sistem yang akan dikembangkan. Beberapa jenis diagram UML yang digunakan oleh penulis antara lain:

a) *Use Case Diagram*

Use Case Diagram digunakan dalam desain UML untuk menjelaskan bagaimana aktor eksternal berinteraksi dengan sistem yang sedang dikembangkan. *Use Case Diagram* memungkinkan penulis untuk mendapatkan gambaran yang jelas

tentang kemampuan sistem dan bagaimana aktor akan berinteraksi dengannya. *Use Case Diagram* dapat dilihat pada gambar 4.16



Gambar 4. 16 Use Case Diagram

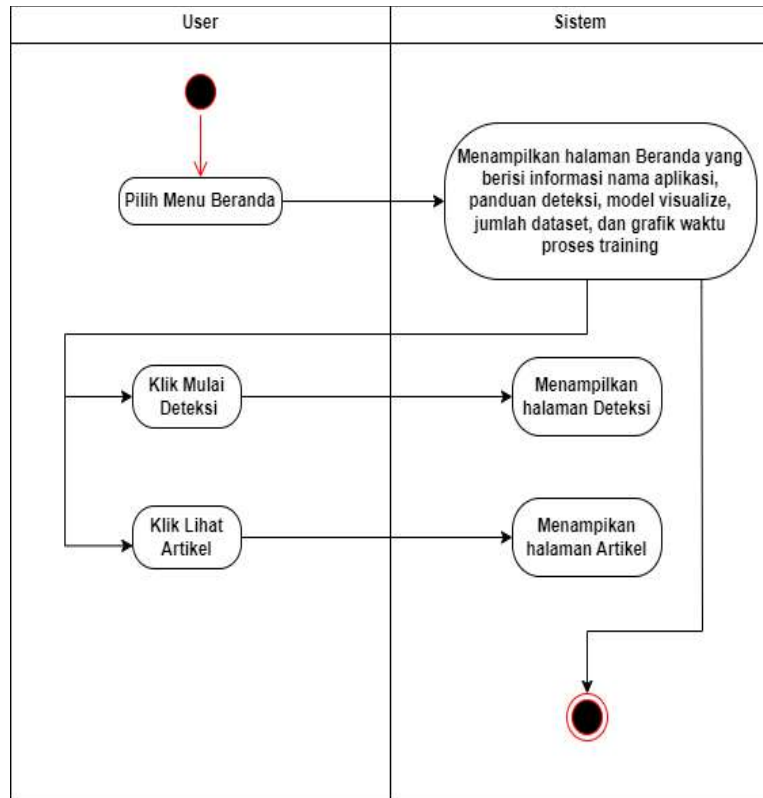
Berdasarkan Diagram pada gambar 4.16, dapat disimpulkan bahwa dalam sistem, terdapat dua aktor yaitu *user* dan *developer*. *User* memiliki akses penuh atas penggunaan aplikasi. *User* dapat mengakses setiap menu yang ada dalam aplikasi, seperti melihat halaman beranda, melakukan deteksi dengan *upload* gambar atau video, melihat isi artikel dan memilih kategori artikel yang ada didalamnya, serta dapat melihat halaman tentang untuk mengetahui informasi aplikasi dan informasi *developer*. Selain itu, pada aktor *developer* (pengembang) memiliki hak akses untuk melakukan *update* artikel, *update halaman tentang* dan mengelola model dataset.

b) *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan aliran kerja atau proses bisnis dari suatu sistem atau bagian dari

sistem. Diagram ini menunjukkan serangkaian aktivitas (tindakan atau tugas) dan bagaimana aktivitas-aktivitas tersebut berhubungan satu sama lain dalam suatu proses. *Activity Diagram* dapat dilihat pada gambar 4.17.

1. *Activity Diagram* Beranda

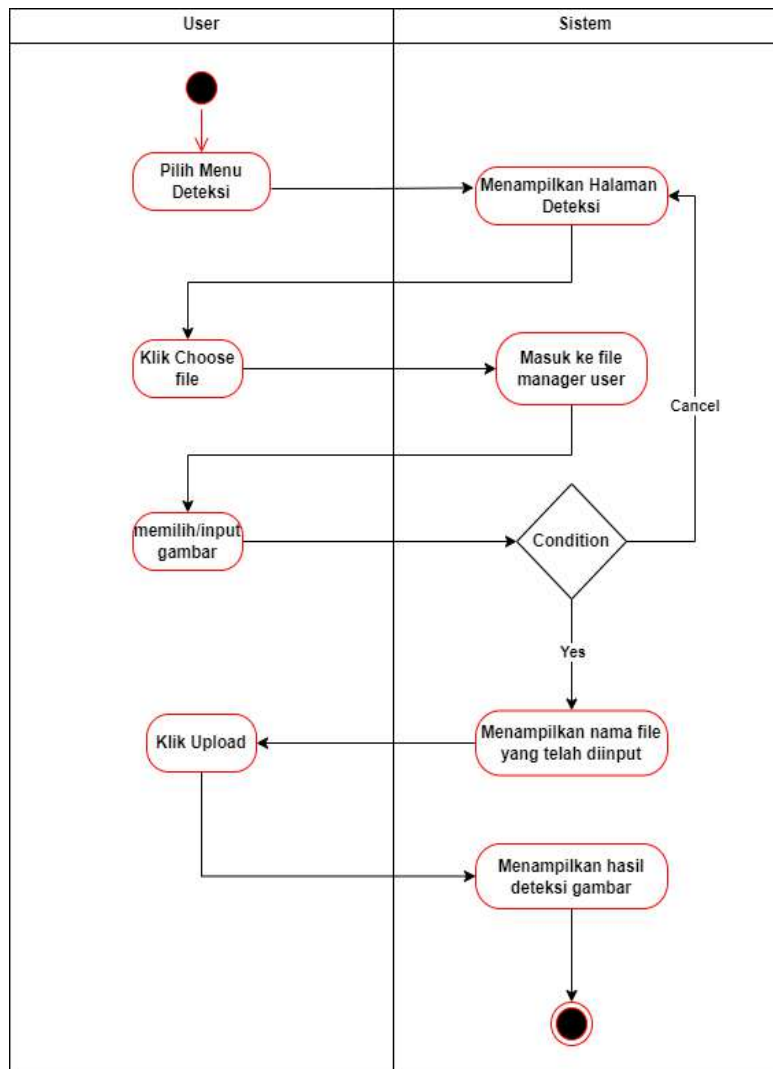


Gambar 4. 17 *Activity Diagram* Beranda

Berdasarkan Gambar 4.17, dijelaskan bahwa saat user memilih menu beranda maka sistem akan otomatis menampilkan halaman beranda yang didalamnya terdapat berbagai informasi mengenai nama aplikasi, panduan deteksi, model visualize, jumlah dataset, grafik-grafik yang didapat sewaktu proses training model yolov8. Didalam tampilan halaman beranda tersebut terdapat beberapa button yang dapat langsung mengarahkan ke halaman yang lain. Jika user mengklik button “Mulai Deteksi” maka sistem

akan menampilkan halaman deteksi. Selain itu, jika user mengklik button “kunjungi artikel” maka sistem akan menampilkan halaman artikel.

2. *Activity Diagram Deteksi*

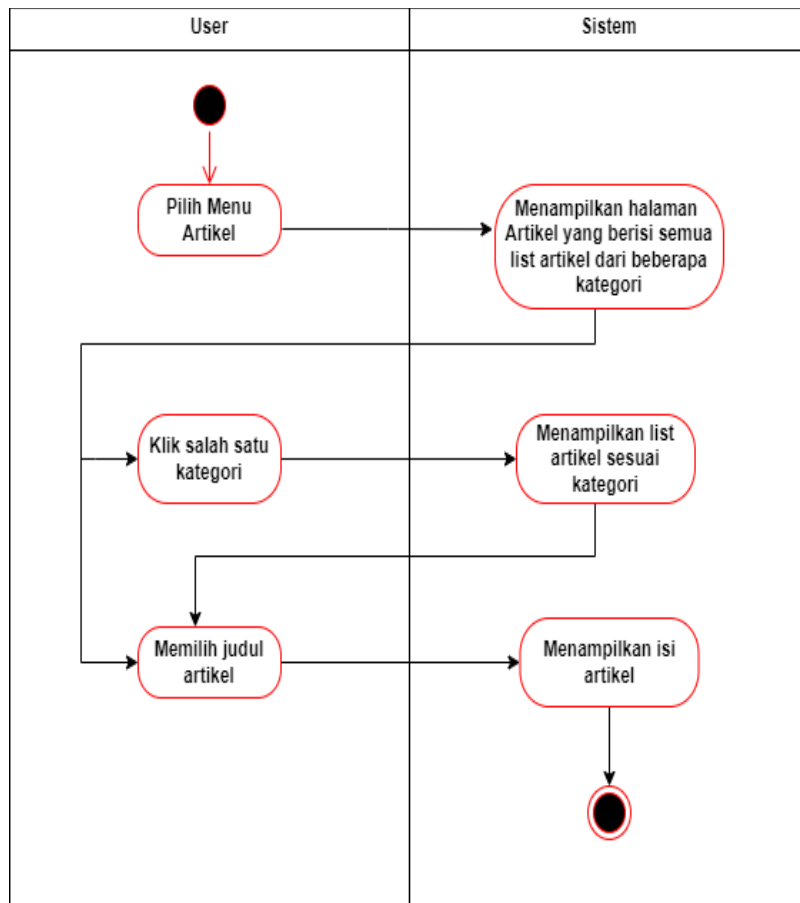


Gambar 4. 18 *Acitivity diagram* deteksi

Berdasarkan Gambar 4.18, dijelaskan bahwa saat *user* memilih menu deteksi maka sistem akan otomatis menampilkan halaman deteksi. *User* dapat menginputkan gambar/video buah kakao yang akan dideteksi dengan mengklik button “*choose file*”. File yang telah dipilih maka

akan ditampilkan kembali di halaman deteksi, jika sudah sesuai maka *user* dapat melakukan upload gambar dengan mengklik *button upload* maka sistem akan menampilkan gambar hasil deteksi.

3. Activity Diagram Artikel

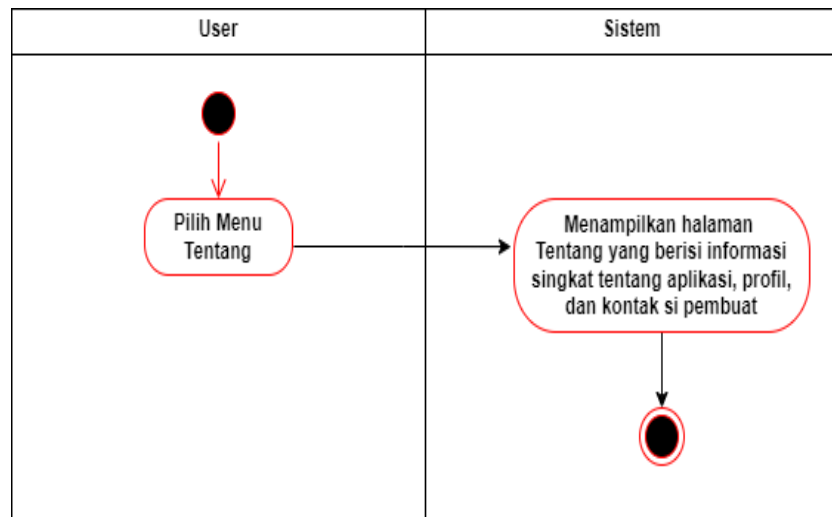


Gambar 4. 19 Activity diagram artikel

Berdasarkan Gambar 4.19, dijelaskan bahwa saat *user* memilih menu artikel, maka sistem akan otomatis menampilkan halaman artikel yang berisikan beberapa *list* artikel yang ada semua kategori. *User* dapat memilih salah satu *button* kategori yang ada, maka sistem akan menampilkan artikel yang sesuai dengan kategori pilihan.

User dapat melihat isi artikel dengan mengeklik judul artikel yang dipilih.

4. Activity Diagram Tentang



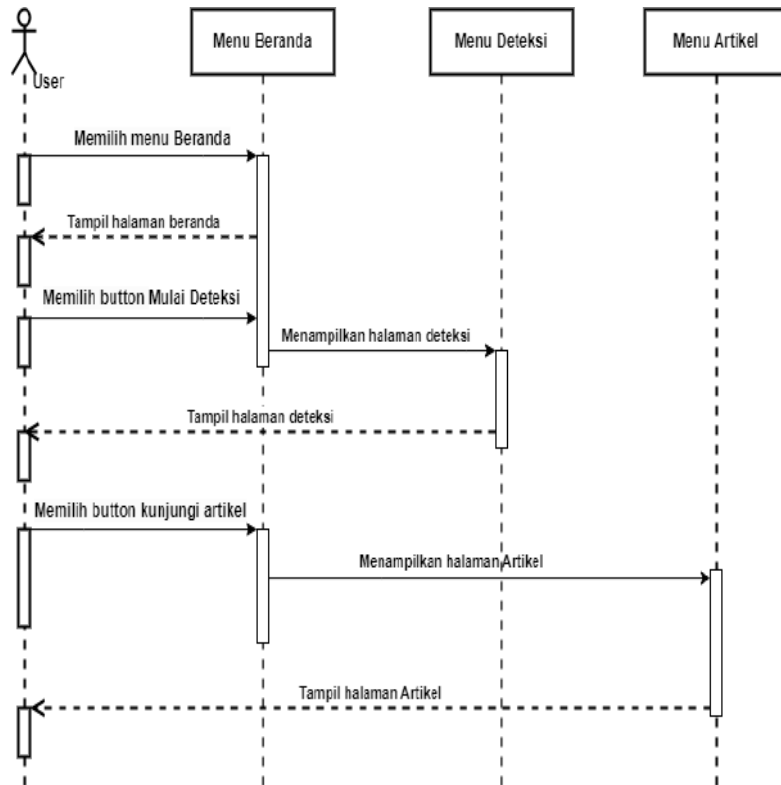
Gambar 4. 20 Acitivity Diagram Tentang

Berdasarkan Gambar 4.20, dijelaskan bahwa saat *user* memilih menu tentang, maka sistem akan otomatis menampilkan halaman tentang yang berisikan beberapa informasi singkat tentang aplikasi, *profil developer*, dan informasi kontak *developer*.

c) Sequence Diagram

Diagram urutan atau *sequence* adalah alat visual yang digunakan untuk menggambarkan interaksi antara objek-objek dalam suatu sistem secara berurutan sepanjang waktu. Dalam diagram ini, objek-objek yang terlibat dalam interaksi diwakili oleh garis vertikal yang disebut "*lifeline*", yang menunjukkan durasi keberadaan objek tersebut selama proses berlangsung.

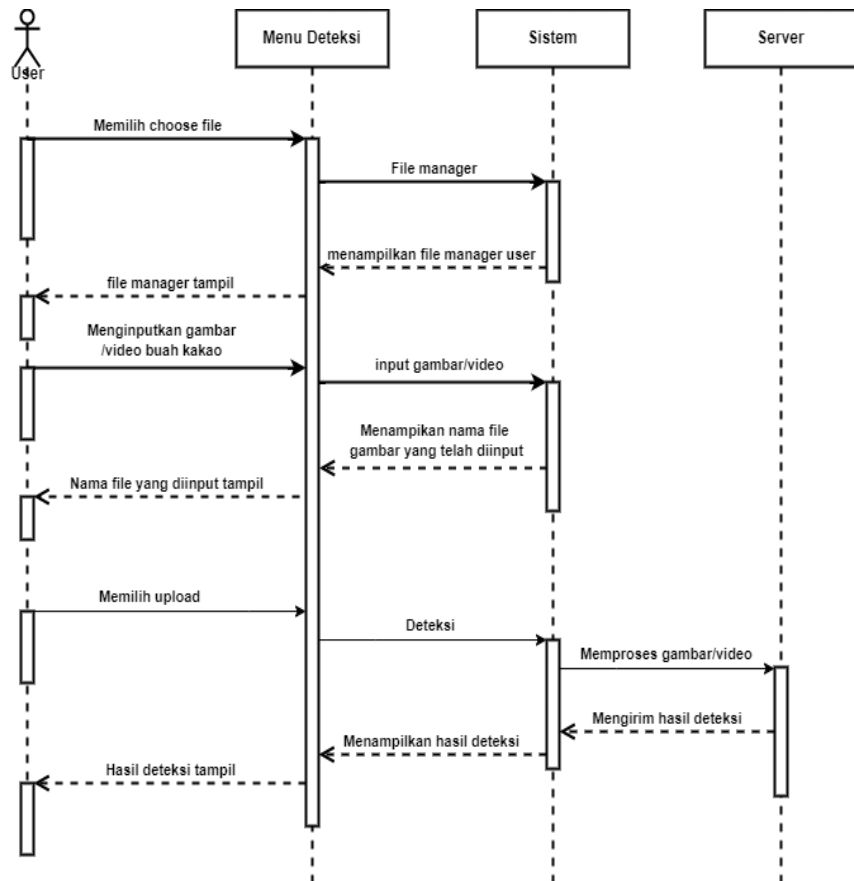
1. Sequence Diagram Beranda



Gambar 4. 21 Sequence diagram beranda

Gambar 4.21 dapat dijelaskan bahwa langkah awalnya adalah *user* memilih menu beranda, maka halaman beranda akan ditampilkan. Didalam menu beranda terdapat *button* “mulai Deteksi”, jika *user* memilih *button* mulai deteksi, maka sistem akan menampilkan halaman menu deteksi. Selain itu, didalam beranda juga terdapat *button* kunjungi artikel, jika *user* memilih *button* kunjungi artikel, maka sistem akan menampilkan halaman artikel.

2. Sequence Diagram Deteksi

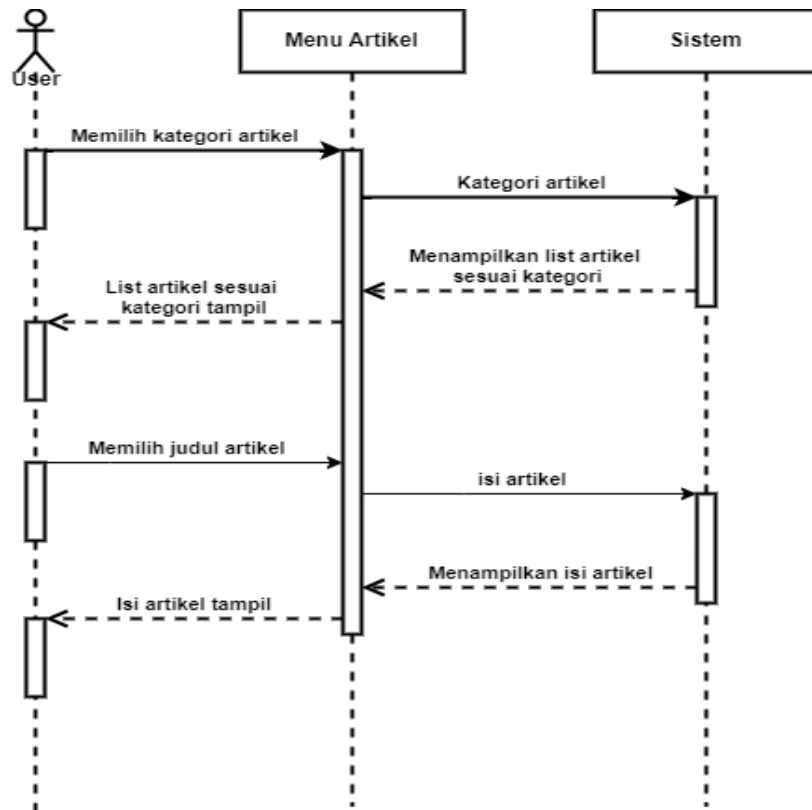


Gambar 4. 22 Sequence diagram deteksi

Gambar 4.22 dapat dijelaskan bahwa langkah awal *user* memilih *choose file* ketika ingin input gambar, maka file manager akan berusaha ditampilkan oleh sistem, saat file manager *user* telah ditampilkan, *user* menginputkan gambar/video buah kakao maka *input* video akan disimpan oleh sistem untuk kemudian ditampilkan ke halaman deteksi dengan menampilkan nama file yang telah *diinput user*. Saat *inputtan* file sudah benar, maka *user* memilih *upload* gambar/video tersebut untuk dilakukan proses deteksi oleh sistem. Server akan memproses gambar/video yang telah diupload untuk dapat mengirim *output* file hasil deteksi ke

sistem untuk kemudian dapat ditampilkan gambar/video hasil deteksi tersebut kedalam halaman deteksi.

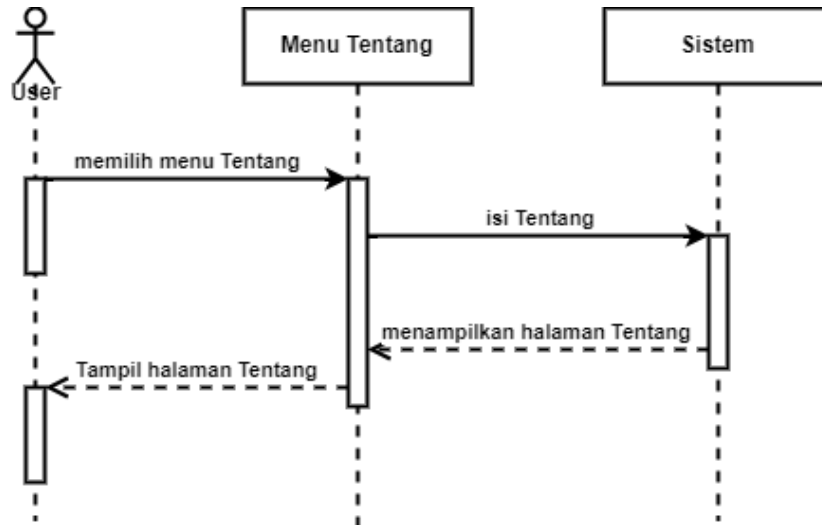
3. *Sequence Diagram Artikel*



Gambar 4. 23 *Sequence diagram* artikel

Berdasarkan gambar 4.23, dapat dijelaskan bahwa langkah awal *user* memilih kategori artikel, sehingga kategori artikel tersebut diproses oleh sistem untuk kemudian dapat menampilkan *list* judul artikel yang sesuai dengan kategori yang dipilih untuk ditampilkan di halaman menu artikel. *User* dapat memilih judul artikel, lalu isi artikel akan diproses oleh sistem untuk dapat menampilkan isi artikel tersebut kepada user.

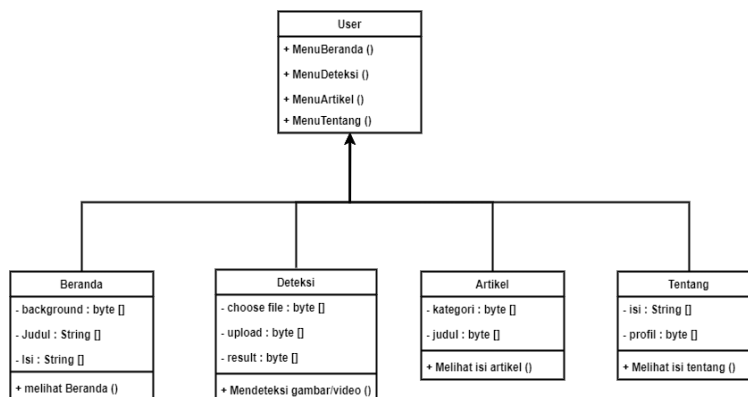
4. Sequence Diagram Tentang



Gambar 4. 24 Sequence diagram tentang

Berdasarkan gambar 4.24, dapat dijelaskan bahwa langkah awal *user* memilih menu tentang, maka isi dari halaman tentang akan diproses oleh sistem untuk kemudian dapat ditampilkan kepada *user* sehingga *user* dapat melihat isi dari halaman tentang tersebut.

d) Class Diagram



Gambar 4. 25 Class diagram

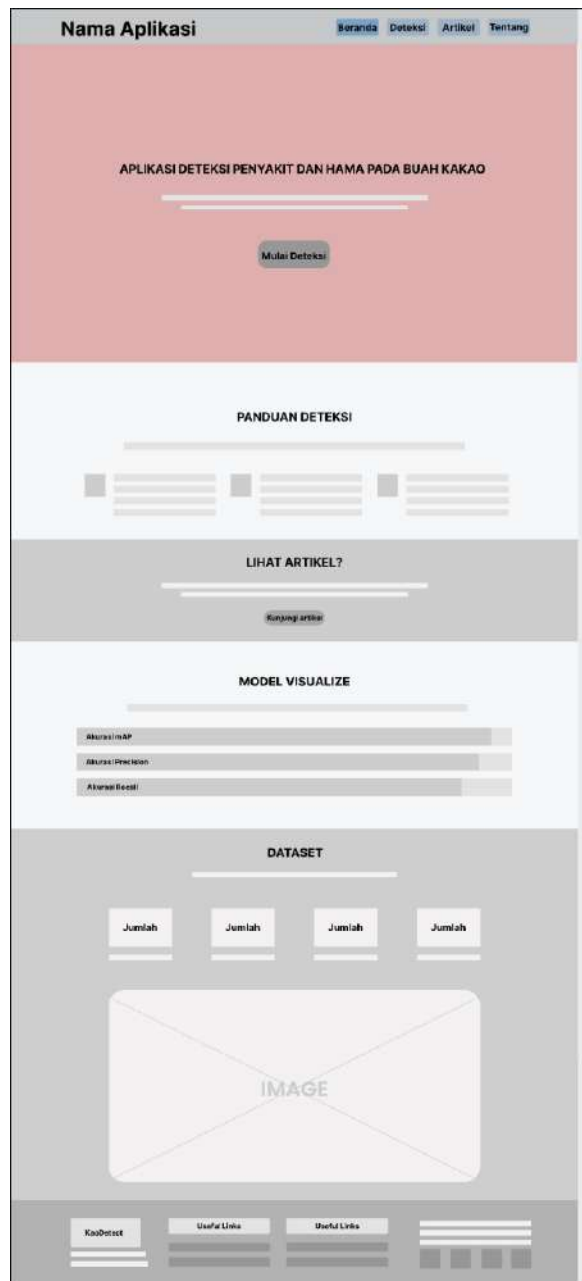
Class diagram adalah diagram struktural dalam UML yang menggambarkan struktur statis suatu sistem atau aplikasi

Diagram ini mengilustrasikan kelas-kelas yang ada dalam sistem, hubungan antara kelas-kelas tersebut, serta atribut dan metode yang dimiliki oleh setiap kelas. Elemen-elemen penting dalam *Class Diagram* mencakup kelas, atribut, metode, dan relasi antar kelas. Tujuan utama dari *class diagram* adalah untuk menggambarkan kelas-kelas dalam sistem, atribut-atribut yang dimiliki oleh setiap kelas, metode-metode yang tersedia, dan hubungan antara kelas-kelas tersebut.

2. Desain User Interface

Dalam *desain User Interface* (UI) untuk aplikasi deteksi penyakit dan hama pada buah kakao, langkah pertama yang diambil oleh penulis adalah merancang desain *wireframe*. *Wireframe* adalah representasi visual sederhana dari antarmuka pengguna yang menunjukkan struktur dasar situs web atau aplikasi tanpa warna, gambar, atau desain yang rumit. *Wireframe* digunakan untuk menunjukkan tata letak, susunan komponen utama, dan fungsi antarmuka pengguna tanpa memperhatikan detail visual yang rumit. Beberapa desain antarmuka pengguna yang disertakan dalam *wireframe* ini meliputi desain beranda, desain halaman deteksi, desain halaman artikel, dan desain halaman tentang.

a) Desain Halaman Beranda

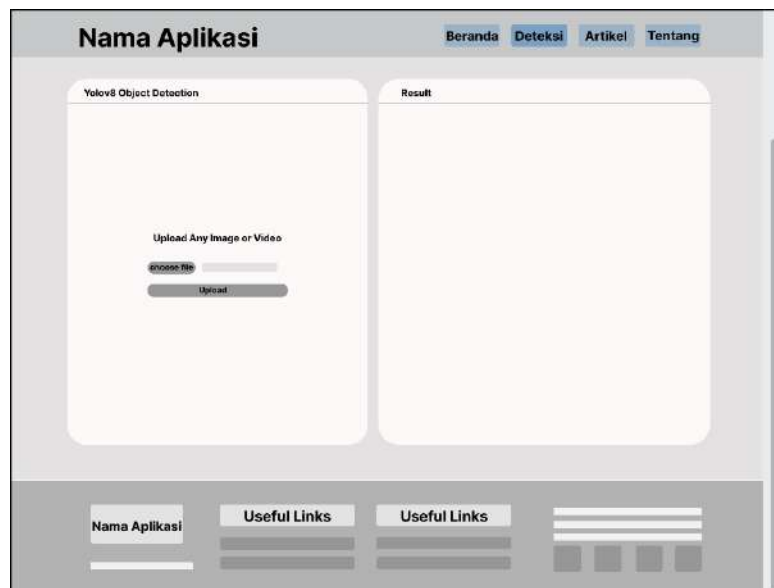


Gambar 4. 26 Desain User Interface Halaman Beranda

Berdasarkan Gambar 4.26, Halaman beranda adalah halaman utama yang pertama kali muncul saat membuka aplikasi. Bagian *header* aplikasi terdapat beberapa menu navbar. Navbar tersebut berisikan navbar menu beranda, deteksi, artikel, dan tentang. Navbar ini akan selalu ada pada setiap header masing-

masing halaman menu yang ada. Di dalam tampilan Beranda ini, nantinya akan menampilkan beberapa informasi mengenai aplikasi termasuk juga dengan beberapa informasi singkat mengenai model pelatihan Yolov8 yang digunakan oleh *developer* (penulis). Dihalaman ini juga terdapat 2 *button* untuk dapat mengarahkan *user* ke fitur-fitur lainnya tanpa memilih menu navbar untuk menambah kemudahan penggunaan aplikasi. Bagian paling akhir dari desain adalah *footer*, yang berisikan beberapa info aplikasi dan *developer* (penulis). *Footer* ini akan selalu ada pada tiap bagian halaman menu navbar yang ada. Dengan demikian, Beranda menjadi pengantar yang sempurna untuk menyambut pengguna saat mereka memulai pengalaman mereka dengan aplikasi deteksi ini.

b) Halaman Deteksi

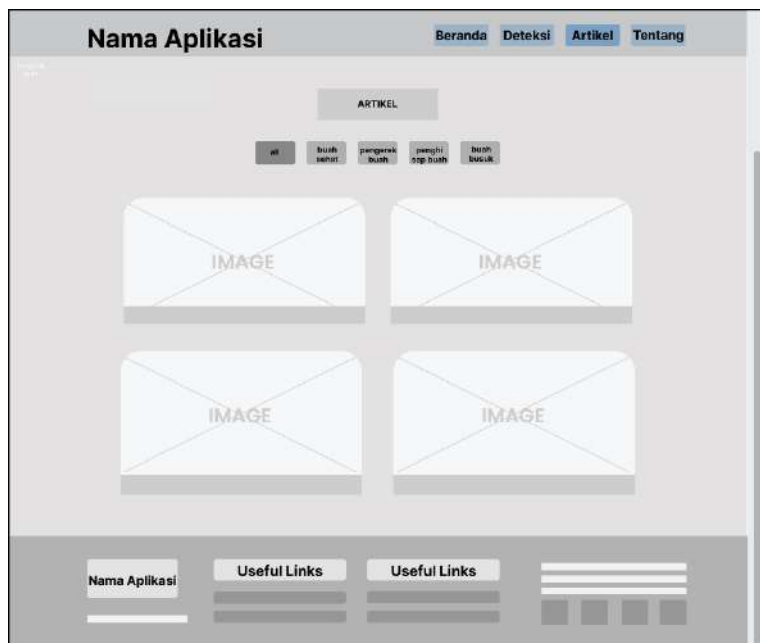


Gambar 4. 27 Desain User Interface Halaman Deteksi

Pada Gambar 4.27 merupakan tampilan dari halaman deteksi. Didalam halaman tersebut, bagian sisi kiri content terdapat fitur untuk proses *upload* gambar yang akan dideteksi. *Content* pada sisi ini berisi tombol *choose file* dan *upload*.

Sedangkan untuk tampilan pada sisi sebelah kanan *content*, merupakan tampilan dari *result*. Tampilan *result* inilah yang nantinya akan menjadi tempat menampilkan gambar hasil deteksi yang sudah diproses oleh sistem. Dengan adanya komponen-komponen ini, pengguna dapat dengan mudah mengakses dan menggunakan fitur deteksi dengan efektif.

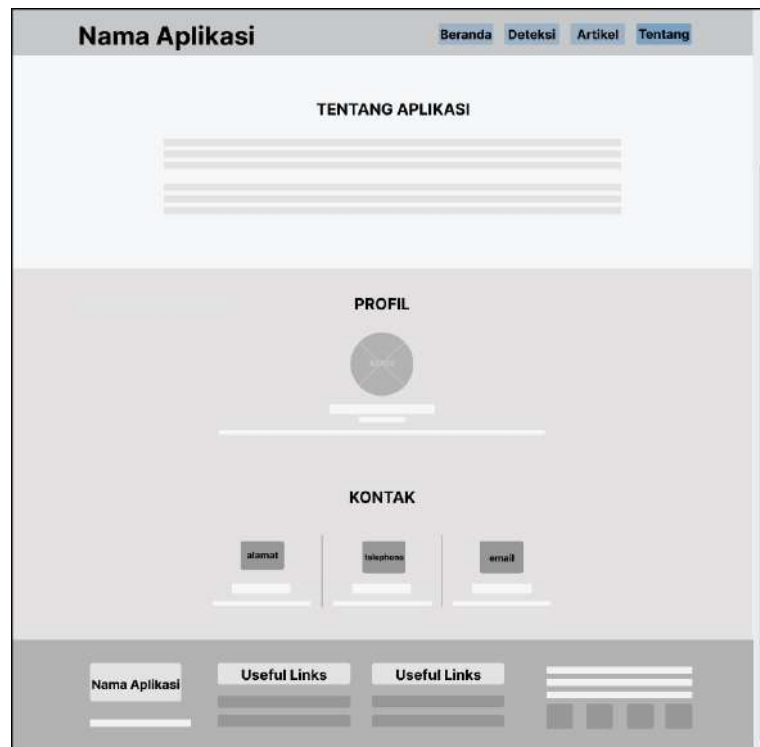
c) Halaman Artikel



Gambar 4. 28 Halaman Artikel

Pada Gambar 4.28 merupakan tampilan dari halaman artikel. Tampilan awal pada halaman ini akan menampilkan semua *list* judul artikel dari semua kategori yang ada. Namun, pada aplikasi ini juga disediakan tombol beberapa kategori mengenai kelas-kelas yang digunakan berupa jenis penyakit dan hama pada buah kakao untuk dapat memudahkan pengguna dalam mencari informasi mengenai kelas-kelas yang akan muncul dalam hasil deteksi penyakit dan hama buah kakao tersebut. Jika pengguna memilih salah satu kategori tersebut, maka sistem akan hanya menampilkan judul artikel yang sesuai dengan kategori yang dipilih.

d) Halaman Tentang



Gambar 4. 29 Halaman Tentang

Pada Gambar 4.39 merupakan tampilan dari halaman tentang. Pada halaman ini nantinya akan berisi beberapa sesi informasi pada bagian bagian yang telah didesain tersebut diatas. Isi dari konten bagian atas, akan diisikan mengenai informasi singkat tentang aplikasi. Pada sisi tengah konten, akan diisikan beberapa informasi mengenai profil diri dari *developer* (penulis), serta bagian sisi bawah konten, akan berisikan mengenai informasi kontak developer yang dapat dihubungi. Dengan demikian, halaman ini tidak hanya berfungsi sebagai sumber informasi yang berharga, tetapi juga memperkuat koneksi antara pengguna dan pengembang aplikasi.

c. Implementasi Sistem

Implementasi sistem pada pembuatan aplikasi ini dibuat menggunakan *text editor* berupa Visual Studio Code. Implementasi sistem dibuat menggunakan *framework* Flask berbasis aplikasi website dengan memasukkan file model *best.pt* dari YOLO v8 untuk diintegrasikan ke dalam aplikasi web tersebut.

Terdapat beberapa langkah penting dalam proses integrasi model kedalam pembuatan aplikasi yaitu sebagai berikut:

1. Memastikan struktur direktori Flask terorganisir dengan baik, termasuk direktori untuk file statis seperti CSS dan JavaScript, direktori untuk template HTML, dan lain-lain beserta dengan requirements yang dibutuhkan dalam pembuatan aplikasi.
2. Menyalin file *best.pt* ke dalam direktori yang sesuai di dalam proyek Flask, dalam hal ini penulis meyalin *file best.pt* kedalam direktori *yolo-Weights*.
3. Dalam file utama Flask (*webapp.py*), melakukan *impor* model YOLO dan menentukan fungsi atau kelas yang dapat memuat model tersebut kedalam aplikasi web dengan menuliskan kode inisialisasi model YOLO v8 serta membuat penulisan kode menggunakan API *method Post* untuk permintaan deteksi.
4. Membuat *endpoint* di Flask untuk menerima permintaan dari *frontend*, memproses gambar menggunakan YOLO v8 untuk deteksi objek, dan memberikan respons yang sesuai kepada pengguna. Setelah itu menguji sistem secara menyeluruh, memastikan bahwa model YOLO dapat diakses dan berfungsi dengan baik, dengan memastikan bahwa aplikasi dapat dirunning dengan mengetikkan kode “python *webapp.py*” pada terminal dan mampu mendeteksi objek sesuai dengan model YOLO v8 yang telah dilatih.

1. Hasil Aplikasi

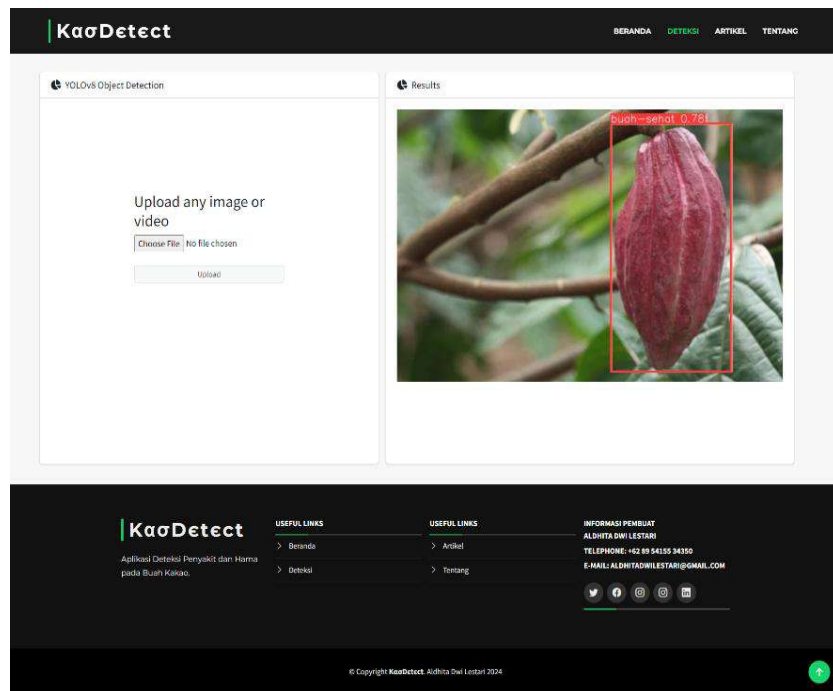
a) Halaman Beranda



Gambar 4. 30 Implementasi halaman beranda

Gambar 4.30, Beranda adalah halaman pertama (utama) yang muncul saat *user* mengakses aplikasi. Pada halaman ini, terdapat beberapa informasi mengenai nama aplikasi, informasi panduan deteksi, informasi model visualize yang menunjukkan persentase akurasi model, dan informasi singkat mengenai dataset serta grafik hasil model pelatihan YOLOv8 yang digunakan oleh *developer* (penulis). Di halaman ini juga terdapat 2 tombol untuk dapat mengarahkan *user* ke fitur-fitur lainnya tanpa memilih menu *navbar* untuk menambah kemudahan penggunaan aplikasi. Yaitu tombol “Mulai Deteksi” untuk mengarah ke halaman menu Deteksi, dan tombol “Kunjungi Artikel” untuk mengarah ke halaman menu Artikel.

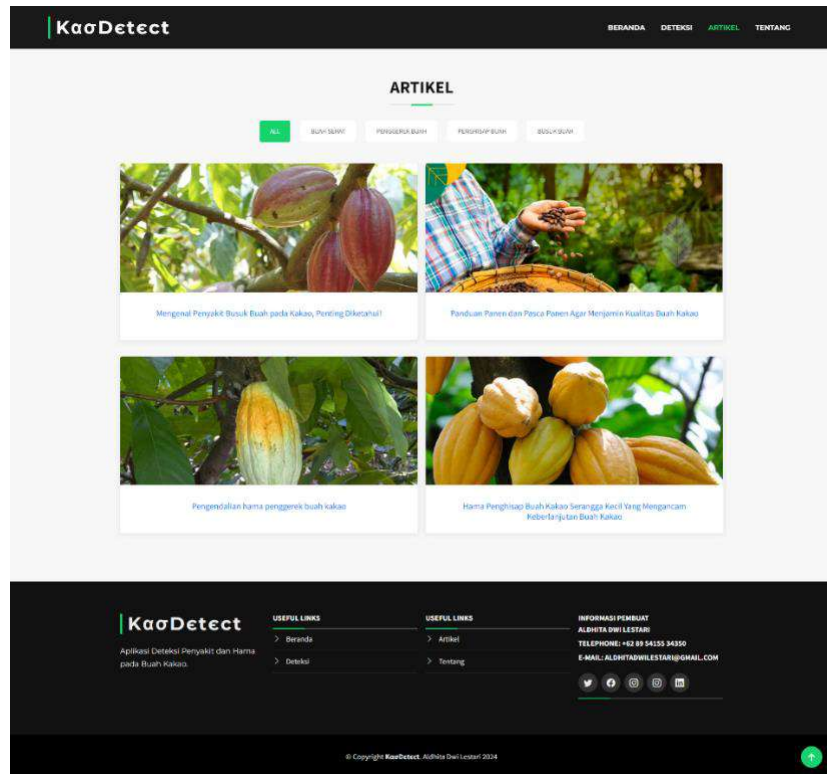
b) Halaman Deteksi



Gambar 4. 31 Implementasi halaman deteksi

Pada Gambar 4.31 merupakan hasil tampilan dari halaman deteksi. Untuk melakukan deteksi, pengguna harus *menginputkan* gambar/video buah kakao yang ingin dideteksi dengan mengklik tombol *choose file*, kemudian pengguna harus melakukan *upload* gambar untuk melakukan proses deteksi. Gambar hasil deteksi akan ditampilkan oleh sistem pada bagian sisi kanan *content (result)*, sehingga pengguna dapat melihat hasil deteksi dengan mudah. Namun untuk hasil deteksi dari file video yang akan dideteksi, akan dimunculkan melalui halaman result tersendiri diluar dari halaman deteksi. Hal ini ditujukan supaya video dapat dilihat dengan lebih jelas jika dapat dilihat secara fullsize.

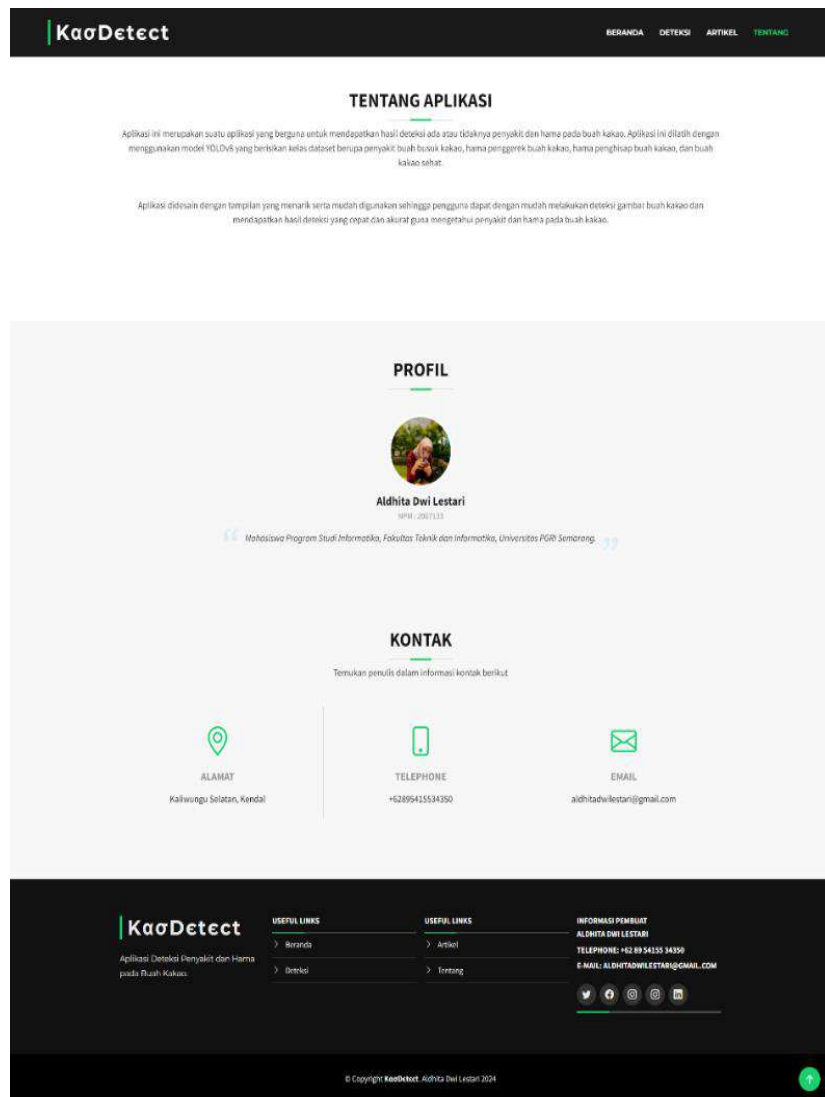
c) Halaman Artikel



Gambar 4. 32 Implementasi halaman artikel

Pada Gambar 4.32 merupakan tampilan dari halaman artikel. Tampilan awal pada halaman ini menampilkan semua *list* judul artikel dari semua kategori yang ada. Namun, pada aplikasi ini juga disediakan tombol beberapa kategori mengenai kelas-kelas yang digunakan berupa jenis penyakit dan hama pada buah kakao untuk dapat memudahkan pengguna dalam mencari informasi mengenai kelas-kelas yang akan muncul dalam hasil deteksi penyakit dan hama buah kakao tersebut. Jika pengguna memilih salah satu kategori tersebut, maka sistem hanya akan menampilkan judul artikel yang sesuai dengan kategori yang dipilih. Pengguna dapat memilih salah satu dari judul artikel yang tersedia untuk dapat melihat isi dari artikel tersebut.

d) Halaman Tentang



Gambar 4. 33 Implementasi halaman tentang

Pada Gambar 4.33 merupakan tampilan dari halaman tentang. Pada halaman terdapat beberapa informasi penting yang meliputi informasi singkat tentang aplikasi, informasi mengenai profil diri dari *developer* (penulis), serta informasi mengenai kontak *developer* yang dapat dihubungi. Dengan demikian, halaman ini diharapkan tidak hanya berfungsi sebagai sumber informasi yang berharga, tetapi juga bertujuan memperkuat koneksi antara pengguna dan pengembang aplikasi.

d. Pengujian

1. Pengujian *Black Box*

Dalam pengujian ini pengujian ini dilakukan oleh 3 dosen Informatika. Proses pengujian *black box* dibagi menjadi 3 tahap yaitu perencanaan pengujian, hasil pengujian, dan kesimpulan pengujian. Setelah merancang pengujian *black box* kemudian rancangan tersebut dicetak menjadi lembar kuesioner dan dibagikan kepada 3 dosen Informatika untuk melakukan pengujian.

a) Hasil Perhitungan Pengujian *Black Box*

Hasil pengujian *black box* dapat dilihat pada tabel 4.10.

Tabel 4. 10 Hasil Pengujian *Black Box*

Nama Pengujian	Test Case	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan					
				Diterima			Ditolak		
				1	2	3	1	2	3
Navigasi	User menekan beberapa tombol pada navigasi.	User dapat masuk ke halaman navigasi yang dituju.	Aplikasi berhasil menampilkan halaman navigasi yang dituju	√	√	√			
Menu Beranda	User melakukan gulir atau scroll ke bawah.	User dapat melihat beberapa informasi yang ada pada halaman beranda.	Aplikasi berhasil menampilkan beberapa informasi dalam halaman beranda.	√	√	√			

	User menekan tombol mulai deteksi.	User dapat masuk ke halaman deteksi.	Aplikasi berhasil menampilkan halaman deteksi.	✓	✓	✓			
	User menekan tombol mulai kunjungi artikel.	User dapat masuk ke halaman artikel.	Aplikasi berhasil menampilkan halaman artikel.	✓	✓	✓			
Menu Deteksi	User menekan tombol deteksi	User dapat masuk ke halaman deteksi.	Aplikasi berhasil menampilkan halaman deteksi.	✓	✓	✓			
	User menekan tombol choose file.	User dapat melihat tampilan file manager dan dapat menginputkan gambar/video.	Aplikasi berhasil menampilkan file manager dan dapat menyimpsn hasil inputtan gambar/video.	✓	✓	✓			
	User menekan tombol Upload.	User dapat mengupload gambar/video inputtan untuk dapat melihat hasil deteksi.	Aplikasi berhasil mengupload gambar/video inputtan dan menampilkan hasil deteksi.	✓	✓	✓			

Menu Artikel	User menekan tombol artikel.	User dapat masuk ke halaman artikel.	Aplikasi berhasil menampilkan halaman artikel.	✓	✓	✓			
	User menekan kategori artikel.	User dapat melihat judul artikel sesuai kategori.	Aplikasi berhasil menampilkan judul artikel sesuai kategori.	✓	✓	✓			
	User memilih judul artikel	User dapat melihat dan membaca isi artikel yang dituju.	Aplikasi berhasil menampilkan isi artikel yang dituju.	✓	✓	✓			
Menu Tentang	User menekan tombol tentang.	User dapat masuk ke dalam halaman tentang.	Aplikasi berhasil menampilkan halaman tentang.	✓	✓	✓			
Footer	User menekan beberapa link navigasi yang ada pada footer.	User dapat masuk ke halaman yang dituju.	Aplikasi berhasil menampilkan halaman navigasi yang dituju.	✓	✓	✓			

b) Kesimpulan Hasil Pengujian *Black Box*

Berdasarkan pengujian *black box*, dari 12 pengujian pada aplikasi yang didapat dari 3 responden, berikut ini hasil pengujian *blackbox*:

- Pengujian pertama
Tercapai $\frac{12}{12} \times 100\% = 100\%$

$$\text{Gagal } \frac{0}{12} \times 100\% = 0\%$$

- Pengujian kedua
Tercapai $\frac{12}{12} \times 100\% = 100\%$

$$\text{Gagal } \frac{0}{12} \times 100\% = 0\%$$

- Pengujian ketiga
Tercapai $\frac{12}{12} \times 100\% = 100\%$

$$\text{Gagal } \frac{0}{12} \times 100\% = 0\%$$

$$\text{Jumlah presentase rata-rata tercapai} = \frac{300\%}{3} = 100\%$$

$$\text{Jumlah presentase rata-rata gagal} = \frac{0\%}{3} = 0\%$$

Berdasarkan analisis tersebut, dari 7 pengujian yang dilakukan oleh 3 responden, hasilnya menunjukkan bahwa tingkat keberhasilan pengujian *black box* mencapai 100%. Sementara kegagalan tidak terjadi sama sekali, sehingga presentasinya adalah 0%. Kesimpulannya, aplikasi berjalan sesuai dengan fungsinya yang diharapkan.

2. Pengujian *White Box*

White box testing adalah teknik pengujian perangkat lunak yang melibatkan pemeriksaan kode sumber dari program untuk mendeteksi adanya kesalahan. Tujuan dari pengujian *white box* adalah memastikan setiap bagian dari kode berfungsi sesuai dengan

spesifikasi yang telah ditetapkan. Hasil pengujian *white box* dapat dilihat pada Tabel 4.11.

Tabel 4. 11 *Whitebox Testing*

Node	Code	Pengertian
1.	<pre>@app.route("/detect") def hello_world(): # return render_template("detect.html") if "image_path" in request.args: image_path = request.args["image_path"] return render_template("detect.html", image_path=image_path) return render_template("detect.html")</pre>	Menampilkan halaman menu deteksi.
2.	<pre>@app.route("/detect", methods=["GET", "POST"]) def predict_img(): if request.method == "POST": if 'file' in request.files: f = request.files['file'] basepath = os.path.dirname(__file__) filepath = os.path.join(basepath, 'uploads', f.filename) print("upload folder is ", filepath) f.save(filepath) predict_img.imgpath = f.filename print("printing predict_img :::::: ", predict_img) file_extension = f.filename.rsplit('.', 1)[1].lower()</pre>	Upload Image

	<pre> if file_extension == 'jpg': # Handle image upload img = cv2.imread(filepath) </pre>	
3.	<pre> elif file_extension == "mp4": # Handle video upload video_path = filepath # replace with your video path cap = cv2.VideoCapture(video_path) # get video dimensions frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)) frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)) # Define the codec and create VideoWriter object fourcc = cv2.VideoWriter_fourcc(*"mp4v") out = cv2.VideoWriter("output.mp4", fourcc, 30.0, (frame_width, frame_height)) </pre>	Upload video
4.	<pre> @app.route("/<path:filename>") def display(filename): folder_path = "runs/detect" subfolders = [f for f in os.listdir(folder_path) </pre>	Memvalidasi apakah menggunakan format .jpg atau .mp4

	<pre> if os.path.isdir(os.path.join(folder_path, f))] latest_subfolder = max(subfolders, key=lambda x: os.path.getctime(os.path.join(folder_path, x))) directory = os.path.join(folder_path, latest_subfolder) print("printing directory: ", directory) files = os.listdir(directory) latest_file = files[0] print(latest_file) image_path = os.path.join(directory, latest_file) file_extension = latest_file.rsplit(".", 1)[1].lower() if file_extension == "jpg": return send_file(image_path, mimetype="image/jpeg") elif file_extension == "mp4": return send_file(image_path, mimetype="video/mp4") </pre>	
5.	<pre> else: return "Invalid file format" </pre>	Format file salah

6.	<pre> # Perform the detection model = YOLO('yolo-Weights/best.pt') detections = model(img, save=True) # Find the latest subdirectory in the 'runs/detect' folder folder_path = os.path.join(basepath, 'runs', 'detect') subfolders = [f for f in os.listdir(folder_path) if os.path.isdir(os.path.join(folder_path, f))] latest_subfolder = max(subfolders, key=lambda x: os.path.getctime(os.path.join(folder_path, x))) # Construct the relative path to the detected image file static_folder = os.path.join(basepath, 'static', 'assets') relative_image_path = os.path.relpath(os.path.join(folder_path, latest_subfolder, f.filename), static_folder) image_path = os.path.join(folder_path, latest_subfolder, f.filename) print("Relative image path:", relative_image_path) # Print the relative_image_path for debugging return render_template('detect.html', image_path=relative_image_path, media_type='image') </pre>	Melakukan proses prediksi gambar berdasarkan model YOLO.
----	---	--

7.	<pre> # initialize the YOLOv8 model here model = YOLO("yolo-Weights/best.pt") while cap.isOpened(): ret, frame = cap.read() if not ret: break # do YOLOv9 detection on the frame here # model = YOLO('yolov9c.pt') results = model(frame, save=True) # working print(results) cv2.waitKey(1) res_plotted = results[0].plot() cv2.imshow("result", res_plotted) # write the frame to the output video out.write(res_plotted) if cv2.waitKey(1) == ord("q"): break return render_template('detect.html', video_path='output.mp4', media_type='video') </pre>	Melakukan proses prediksi video berdasarkan model YOLO.
8.	<pre> @app.route("/<path:filename>") def display(filename): folder_path = "runs/detect" </pre>	Menampilkan hasil deteksi gambar atau

	<pre>subfolders = [f for f in os.listdir(folder_path) if os.path.isdir(os.path.join(folder_path, f))] latest_subfolder = max(subfolders, key=lambda x: os.path.getctime(os.path.join(folder_path, x))) directory = os.path.join(folder_path, latest_subfolder) print("printing directory: ", directory) files = os.listdir(directory) latest_file = files[0] print(latest_file) image_path = os.path.join(directory, latest_file) file_extension = latest_file.rsplit(".", 1)[1].lower() if file_extension == "jpg": return send_file(image_path, mimetype="image/jpeg") elif file_extension == "mp4": return send_file(image_path, mimetype="video/mp4")</pre>	video.
--	---	--------

1. *Basis Path* untuk fungsi Deteksi

Basis path (jalur dasar) untuk pengujian *white box* dari fungsi prediksi yang diimplementasikan di `main.js`. *Basis path testing* adalah teknik yang digunakan untuk memastikan bahwa semua jalur eksekusi yang mungkin dalam kode telah diuji setidaknya sekali. Berikut adalah langkah langkah dalam basis path testing untuk fungsi deteksi.

a) Mengidentifikasi jalur dasar

Mengidentifikasi semua jalur eksekusi dalam kode yang relevan dengan fungsi deteksi. Setiap node yang telah ditetapkan sebelumnya akan menjadi bagian dari jalur eksekusi.

b) Menghitung Kompleksitas Siklomatik

Kompleksitas siklomatik adalah metrik yang digunakan untuk mengukur jumlah jalur *linier independen* dalam program. Rumus untuk menghitung kompleksitas siklomatik($V(G)$) adalah:

$$V(G) = E - N + 2$$

Dimana:

- E adalah jumlah edge (garis alur dalam flow graph)
- N adalah jumlah node (titik keputusan dalam flow graph)

2. *Flow Graph* untuk Fungsi Deteksi

Berikut adalah representasi *flow graph* dari kode fungsi deteksi pada gambar 4.34.

Edges: 10

Nodes: 8

Menggunakan rumus kompleksitas siklomatik:

$$V(G) = E - N + 2 = 10 - 8 + 2 = 4$$

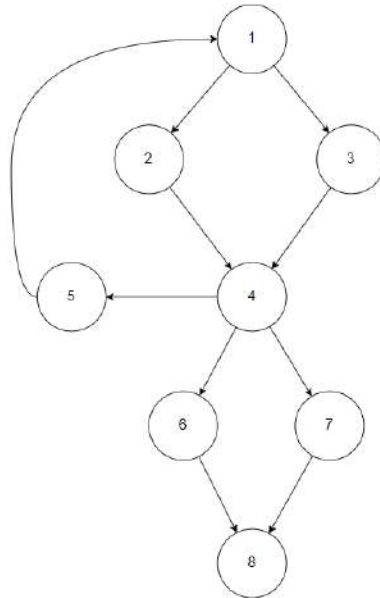
Ini menunjukkan bahwa ada 4 *path* jalur *independen* dalam fungsi prediksi. Jalur *Independen*:

Path 1: 1, 2, 4, 5, 1

Path 2: 1, 3, 4, 5, 1

Path 3: 1, 2, 4, 6, 8

Path 4: 1, 3, 4, 7, 6



Gambar 4. 34 *Flowgraph Basis Path*

3. *Value Test*

Selanjutnya akan dilakukan pengujian *value test* yaitu sebagai pengujian pada setiap independent path. *Value test* dapat dilihat pada Tabel 4.12.

Tabel 4. 12 *Value Test*

No	Path	Input	Output	Keterangan
1.	1, 2, 4, 5, 1	Melakukan upload gambar namun dengan format yang salah (bukan JPG)	Gagal menampilkan hasil deteksi gambar sehingga kembali ke	Tercapai

			halaman awal deteksi	
2.	1, 3, 4, 5, 1	Melakukan upload video namun dengan format yang salah (bukan mp4)	Gagal menampilkan hasil deteksi video sehingga kembali ke halaman awal deteksi	Tercapai
3.	1, 2, 4, 6, 8	Melakukan upload gambar dengan format JPG	Berhasil menampilkan hasil deteksi gambar	Tercapai
4.	1, 3, 4, 7, 6	Melakukan upload video dengan format JPG	Berhasil menampilkan hasil deteksi video	Tercapai

Hasil pengujian pada tabel 4.11 diperoleh hasil sebagai berikut:

- $\text{Tercapai} = \frac{4}{4} \times 100\% = 100\%$
- $\text{Gagal} = \frac{0}{4} \times 100\% = 0\%$

Berdasarkan pengujian *whitebox* yang telah dilakukan dengan menggunakan beberapa tahapan yang ada diatas maka pengujian ini memiliki persentase keberhasilan sebanyak 100% dan kegagalan memiliki persentasi 0%. Dapat disimpulkan hasil dari

pengujian *white box* di atas bahwa aplikasi berjalan dengan baik.

3. Pengujian *User Acceptance Testing* (UAT)

Pengujian UAT pada penelitian ini melibatkan lima responden, terdiri dari masyarakat umum dan para petani kakao. Pengujian ini melibatkan pengguna yang berpotensi menggunakan aplikasi dalam kehidupan sehari-hari mereka, dan memberikan umpan balik yang berharga mengenai kinerja dan kegunaannya. Melalui proses ini, pengembang dapat memastikan bahwa aplikasi tidak hanya berfungsi secara teknis tetapi juga memenuhi kebutuhan dan harapan pengguna, khususnya dalam membantu petani mendeteksi hama dan penyakit pada buah kakao.

1) Hasil Perhitungan Pengujian *User Acceptance Testing* (UAT)

Berikut ini merupakan hasil kuesioner pengujian *User Acceptance Testing* (UAT) yang telah disebarakan kepada 5 responden. Hasil pengujian UAT dapat dilihat pada Tabel 4.13.

Tabel 4. 13 User Acceptence Testing (UAT)

Pertanyaan	Hasil Pengujian				
	Responden 1	Responden 2	Responden 3	Responden 4	Responden 5
1	5	5	4	5	5
2	4	5	5	4	4
3	4	4	5	5	4
4	3	4	5	4	3
5	4	5	4	5	5
6	4	5	5	5	5
7	5	4	5	4	4
8	5	5	5	5	5

9	3	5	5	4	5
10	5	4	5	5	5
Jumlah skor	42	46	48	46	45
Persentase	84%	92%	96%	92%	90%
Total	454%				

2) Kesimpulan Hasil Pengujian *User Acceptance Testing* (UAT)

Dari hasil evaluasi persentase untuk setiap pertanyaan yang mencakup aspek kegunaan, kemudahan penggunaan, dan *User Interface* (UI), yang telah diujikan oleh 5 responden. Nilai-nilai tersebut dijumlah dan kemudian dicari nilai rata-ratanya. Nilai rata-rata ini dapat dihitung dengan menggunakan persamaan sebagai berikut:

Rumus:

$$\text{Persentase rata-rata} = \frac{\text{jumlah total persentase}}{\text{jumlah responden}}$$

$$\text{Persentase rata-rata} = \frac{454\%}{5} = 90,8\%$$

Dengan daftar kategori sebagai berikut:

0% - 20% = Sangat Kurang

21% - 40% = Kurang

41% - 60% = Cukup Baik

61% - 80% = Baik

81% - 100% = Sangat Baik

Dari perhitungan tersebut, diperoleh presentase rata-rata dari tiga aspek sebesar 90,8%. Dengan demikian, dapat

disimpulkan bahwa pengujian UAT pada aplikasi ini memperoleh kategori yang sangat baik.

4. Pengujian Lapangan

Pengujian lapangan aplikasi deteksi hama dan penyakit pada buah kakao ini dilakukan di Desa Nglegi, Kec. Patuk, Kab. Gunung Kidul, Jawa Tengah melalui Ketua UPH Kakao KT. Ngudi Mulyo, dan salah satu petani kakao masyarakat Gunung Kidul. Pengujian lapangan dilakukan dengan petani yang mengetahui pengetahuan luas mengenai budidaya tanaman kakao dengan baik untuk memastikan bahwa segala informasi yang terdapat dalam aplikasi dan juga jenis hama dan penyakit yang dapat dideteksi dalam aplikasi dapat diverifikasi kebenarannya serta mengetahui seberapa bermanfaat aplikasi yang dibuat untuk petani kakao. Pengujian ini bertujuan untuk mengevaluasi kemampuan aplikasi dalam mendeteksi hama dan penyakit secara langsung pada buah kakao di kondisi lapangan.



Gambar 4. 35 Pengujian Lapangan dengan Ketua Kelompok Tani Ngudi Mulyo



Gambar 4. 36 Pengujian Lapangan bersama petani kakao

Selama survei yang dilakukan oleh penulis bersama para petani kakao di daerah Gunung Kidul, penulis mendapati bahwa dalam tempat penelitian hanya ditemukan sampel buah kakao sehat, dan busuk buah kakao yang umum menyerang tanaman kakao di tempat lokasi pengujian. Para petani di sini menjelaskan bahwa umumnya petani mengatasi penyakit busuk buah tersebut dengan cara melakukan pembuangan pada buah yang terinfeksi secara hati-hati dan menerapkan pemangkasan menggunakan kultur teknis yang tepat dalam waktu sebulan sekali. Panen sering pada buah yang sudah masak dilakukan untuk mencegah kehilangan panen untuk menghindari terjadinya infeksi pada buah yang dapat menyebabkan busuk buah.

Penulis mengambil beberapa foto buah kakao di tempat lokasi pengujian melalui handphone kemudian dipindah ke dalam laptop untuk kemudian dilakukan deteksi pada aplikasi. Adapun hasil dari pengujian lapangan dapat dilihat pada gambar 4.37.



Gambar 4. 37 Hasil Pengujian Lapangan

Aplikasi berhasil mendeteksi foto inputan yang didapat dari lokasi survei. *User* (para petani) telah memverifikasi bahwa hasil deteksi hama dan penyakit yang ditampilkan pada aplikasi sudah sesuai dengan akurat, yang mana aplikasi dapat mendeteksi buah sehat kakao dan busuk buah kakao dengan benar.

Dikarenakan untuk jenis hama penggerek buah kakao, dan hama penghisap buah kakao tidak ditemukan dalam lokasi survei, maka penulis melakukan pengujian deteksi menggunakan file gambar yang sudah ada dalam file manager penulis. Untuk hasil dari deteksi penggerek buah dan penghisap buah, juga sudah diverifikasi oleh para petani bahwa ciri gejala dari foto yang dideteksi sudah benar dan sesuai dengan hasil deteksi yang didapatkan.

Dari hasil pengujian, baik dari ketua UPH Kelompok Tani Ngudi Mulyo dan salah satu petani kakao masyarakat Gunung Kidul, menyatakan bahwa dengan adanya aplikasi deteksi hama dan penyakit buah kakao berbasis website ini dinilai sangat bermanfaat khususnya bagi para petani kakao, karena menurut mereka selama ini mereka belum mendapatkan teknologi yang dapat memudahkan petani kakao dalam membantu mendeteksi adanya hama dan penyakit pada buah

kakao. Para petani hanya mengandalkan program penyuluhan dari satu tempat ke tempat lainnya dengan hanya memberikan pengarahan dan pengetahuan jenis hama dan penyakit pada buah kakao melalui teori, sehingga petani dituntut untuk memahami hal tersebut secara teoritis demi menjaga kesuksesan budidaya kakaonya. Namun, dengan adanya aplikasi ini dianggap dapat sangat membantu petani dalam melakukan budidaya kakao, karena jika suatu hama dan penyakit pada buah kakao dapat diketahui sejak dini dengan tepat, maka dapat memudahkan pula dalam hal penanganannya, sehingga tidak terjadi gagal produksi yang menyebabkan kerugian besar, Namun terdapat beberapa saran dari mereka bahwa akan lebih bagus jika aplikasi nantinya dapat dikembangkan kembali menjadi aplikasi yang dapat diakses melalui smartphone dan dapat dilakukan deteksi secara langsung tanpa harus unggah gambar/video terlebih dahulu melalui laptop.

Hasil dari pengujian lapangan ini menunjukkan bahwa aplikasi deteksi hama dan penyakit pada buah kakao memiliki manfaat bagi petani kakao. Dengan kemampuannya untuk melakukan deteksi hama dan penyakit buah kakao, aplikasi ini dapat membantu bagi petani dalam mengidentifikasi jenis hama dan penyakit buah kakao serta dapat mendapatkan informasi terkait penyebab, ciri gejala, beserta cara pengendaliannya.

B. Pembahasan

1. Pemodelan Data

a. *Data Acquisition*

Dalam penelitian ini, pengumpulan data dilakukan dengan pencarian dataset yang diunduh dari situs web Universe Roboflow dengan judul dataset "*Cocoa Detection*", "*Tugas Akhir Hama Penyakit Kakao*", dan "*Cacao*". Dalam beberapa dataset tersebut,

penulis mengambil beberapa kelas data yang dibutuhkan untuk pembuatan model deteksi penyakit dan hama pada buah kakao. Data yang telah terkumpul, kemudian dijadikan empat kelas dataset yang digunakan dalam penelitian, yaitu kelas busuk buah, penggerek buah, penghisap buah, dan buah sehat.

b. *Labelling Data*

Proses *Labelling Data* dilaksanakan melalui penandaan citra menggunakan kotak pembatas (*bounding box*), sambil menetapkan kelas yang relevan pada setiap citra. Metode *labelling data* dilakukan dengan menggunakan website Roboflow pada *tools annotation*, di mana setiap citra diberikan kotak pembatas untuk menandai area yang signifikan untuk kemudian masing-masing diberi label yang menunjukkan informasi mengenai *bounding box* tersebut.

c. *Split Data*

Pada tahap pengelompokan data, dataset disegmentasikan menjadi tiga bagian, yakni dataset pelatihan (*training*) dataset validasi (*validation*) dan dataset pengujian (*testing*). Dataset pelatihan merangkap sebanyak 70% dari total keseluruhan dataset, sementara dataset validasi terdiri dari 15% dari keseluruhan dataset dan dataset pengujian terdiri dari 15% dari keseluruhan dataset.

Adapun Pemilihan pembagian dataset menjadi 70% untuk *training data*, 15% untuk *validation data*, dan 15% untuk *test data* pada saat pemodelan YOLOv8 memiliki alasan-alasan yang spesifik untuk masing-masing persentase sebagai berikut:

1) *Training Data* (70%)

Tujuan menggunakan data *training* sebesar 70% dari total dataset memungkinkan model untuk belajar dengan cukup baik dari variasi yang ada dalam data. Alasan dari pembagian ini yaitu karena semakin besar training set, semakin baik model dapat mempelajari pola-pola yang ada dalam data. Dengan lebih banyak data *training*, model memiliki potensi untuk meningkatkan

generalisasi dan mengurangi kemungkinan *overfitting* terhadap data *training* tersebut.

2) *Valid Data* 15%

Validation set digunakan untuk mengevaluasi kinerja model selama proses pelatihan dan untuk melakukan penyetelan *hyperparameter*. Menggunakan 15% dari dataset sebagai *validation set* memberikan ukuran yang cukup untuk mengukur kinerja model dengan baik tanpa mempengaruhi proses pelatihan secara signifikan. *Validation set* yang lebih besar memberikan evaluasi yang lebih stabil dan dapat diandalkan terhadap model yang sedang dikembangkan.

3) *Test Data* 15%

Test data berperan sebagai simulasi data yang belum pernah dilihat sebelumnya untuk mengevaluasi kinerja akhir model. Menggunakan 15% dari dataset sebagai test set memastikan bahwa evaluasi akhir model dilakukan secara obyektif terhadap data yang benar-benar baru, mirip dengan kondisi di mana model akan digunakan di lingkungan produksi atau di dunia nyata. Dengan memisahkan test set dari awal, kita dapat menghindari bias yang mungkin terjadi jika data test "bocor" ke dalam proses pelatihan atau validasi.

d. *Preprocessing*

Tahap *preprocessing data* (pra-pemrosesan) adalah tahap penting dalam analisis data yang bertujuan untuk membersihkan, mengubah, dan mempersiapkan data mentah agar sesuai untuk analisis lebih lanjut. Pada tahap ini, dilakukan penyaringan data yang tepat untuk disertakan dalam dataset dan dilakukan penyesuaian ukuran setiap citra menjadi 640 x 640 pixel secara otomatis oleh roboflow.

e. *Training Data*

Training data (data pelatihan) dalam konteks *machine learning* adalah subset dari dataset yang digunakan untuk melatih atau mengajar model untuk melakukan prediksi atau tugas tertentu. Data ini berperan penting dalam proses pengembangan model karena model belajar dari pola atau hubungan yang terdapat dalam data ini.

Sebelum melakukan proses *training*, maka dilakukan konfigurasi YOLOv8 terlebih dahulu. Konfigurasi YOLOv8 melibatkan beberapa langkah utama yang meliputi Persiapan lingkungan dengan melakukan instalasi dependensi yang diperlukan, termasuk Python dan pustaka terkait seperti PyTorch, dan *clone* repositori YOLOv8 dari *repository* resmi *Ultralytics*. Persiapan dataset dengan melakukan konfigurasi data “.yaml” yang ada, kemudian melakukan pengaturan parameter pelatihan, dan penyesuaian konfigurasi model. Adapun parameter-parameter yang digunakan pada tahap training ini yaitu menggunakan *epoch* 30, nilai *conf* 0,25, *batch* 16, dan menggunakan konfigurasi model *yolov8n*.

Dalam penelitian ini model dilatih menggunakan 859 citra yang terbagi menjadi empat kelas, antara lain sebagai berikut: busuk buah, penggerek buah, penghisap buah, dan buah sehat. Proses training dilakukan dengan menggunakan model YOLOv8 dengan 30 putaran (*epoch*) melalui Google Colab.

Menggunakan 30 *epoch* dalam proses training model YOLOv8 penting untuk mencapai konvergensi yang optimal dan stabil. *Epoch* yang cukup memberi waktu bagi model untuk belajar secara mendalam dari data *training* dan menyesuaikan parameter-nya untuk meningkatkan akurasi dan keandalan prediksi. Dengan menggunakan 30 *epoch*, kita dapat melakukan eksperimen yang cukup untuk menemukan kombinasi *hyperparameter* yang paling cocok untuk

tugas atau dataset yang spesifik. hal ini membantu meningkatkan kinerja model dan mengurangi risiko *overfitting* atau *underfitting*. Selain itu, menggunakan 30 *epoch* adalah kompromi yang sering digunakan dalam pengembangan model *deep learning* untuk memberikan waktu yang cukup bagi model untuk belajar dari data tanpa memerlukan sumber daya komputasi yang terlalu banyak tanpa terkendala oleh limit waktu pada penyimpanan GPU di Google Colab, karena proses *training* membutuhkan waktu yang lama oleh karena itu, peneliti memilih 30 *epoch* pada proses *training* model ini.

Dalam proses *training* menggunakan YOLOv8 pada google colab dilakukan secara otomatis oleh sistem dengan menentukan ukuran *batch*nya. Adapun disini secara otomatis ditentukan besar ukuran *batch* 16. "Batch 16", mengacu pada pengaturan ukuran *batch* yang digunakan selama proses *training*. *Batch* adalah sejumlah sampel data yang diproses bersama-sama dalam satu iterasi pelatihan. Secara khusus, "batch 16" berarti bahwa dalam setiap iterasi pelatihan, Google Colab akan mengambil 16 sampel data dari dataset *training* dan memprosesnya secara bersamaan. Setelah satu *batch* data diproses, model akan menghitung gradien (*gradient*) dari fungsi kerugian (*loss function*) berdasarkan prediksi yang dihasilkan, dan kemudian melakukan pembaruan parameter berdasarkan gradien tersebut (proses yang disebut *backpropagation*). Beberapa poin penting tentang *batch size* 16 di Google Colab:

- 1) Ukuran *Batch*: *Batch size* 16 adalah ukuran yang cukup kecil, tetapi umum digunakan dalam pelatihan *deep learning*. Ukuran *batch* yang lebih kecil seperti ini dapat membantu mempercepat proses pelatihan karena memungkinkan GPU untuk memproses lebih banyak *batch* dalam waktu yang sama.
- 2) Memori GPU: Google Colab menyediakan akses gratis ke GPU, tetapi memiliki batasan pada penggunaan memori. Dengan memilih *batch size* 16, dapat memastikan bahwa proses pelatihan

dapat berjalan lancar tanpa membebani memori GPU terlalu berat.

- 3) Kompromi Antara Kecepatan dan Stabilitas: Memilih ukuran *batch* 16 sering kali merupakan kompromi antara kecepatan pelatihan dan stabilitas model. *Batch* yang lebih besar dapat mempercepat pelatihan dalam beberapa kasus, tetapi dapat juga membuat pelatihan menjadi lebih tidak stabil atau menyulitkan konvergensi.

Pada tahap training model ditetapkan nilai *confidence* yang menjadi ambang batas (*threshold*) sebesar 0,25. Dalam konteks YOLOv8, *conf* biasanya merujuk pada nilai *threshold* untuk skor kepercayaan prediksi objek. Menggunakan nilai *conf* (*confidence threshold*) sebesar 0,25 yaitu untuk mengatur tingkat kepercayaan minimal yang diperlukan agar sebuah deteksi dianggap *valid*. Alasan menggunakan nilai *conf* sebesar 0,25 adalah untuk memastikan bahwa deteksi objek yang dilakukan oleh model YOLOv8 memiliki tingkat kepercayaan yang cukup tinggi. Nilai ini memfilter prediksi yang memiliki skor kepercayaan rendah, yang mungkin tidak cukup andal untuk diandalkan dalam aplikasi praktis. Dengan menetapkan nilai *threshold* yang lebih tinggi seperti 0,25 maka cenderung dapat mengurangi kemungkinan adanya deteksi palsu (*false positive*). Deteksi yang memiliki skor kepercayaan rendah dapat dianggap sebagai *false positive* dan tidak diambil keputusan tindak lanjut. Selain itu, dengan menentukan nilai *conf* yang lebih tinggi juga membantu dalam meningkatkan efisiensi komputasi.

f. Evaluasi Model

Evaluasi model merujuk pada proses mengukur kinerja atau keefektifan suatu model statistik atau machine learning dalam memprediksi atau mengklasifikasikan data yang belum pernah dilihat sebelumnya. Evaluasi model penting dilakukan untuk memastikan

bahwa model yang dikembangkan dapat diandalkan dan bermanfaat dalam mengambil keputusan atau menyediakan prediksi yang akurat. Pada tahap ini dilakukan perhitungan akurasi model yang didapat dari grafik confusion matrix dengan perolehan hasil akurasi sebesar 94% .

g. *Test Model*

Pada konteks machine learning dan statistik, "*testing a model*" atau pengujian model mengacu pada proses mengevaluasi kinerja model yang telah dilatih dengan menggunakan data yang belum pernah dilihat sebelumnya, yaitu data pengujian (*test set*). Tujuan utama dari pengujian model adalah untuk memastikan bahwa model yang dikembangkan dapat menggeneralisasi dengan baik terhadap data baru yang tidak pernah dilihat sebelumnya, sehingga dapat diandalkan untuk membuat prediksi atau klasifikasi yang akurat.

h. *Upload Model*

upload model adalah titik penting di mana model yang telah diperbaharui dengan pengetahuan dari dataset pelatihan diimplementasikan ke dalam solusi yang nyata. Kita perlu *mengupload* model ke *platform* yang dituju. Pada tahap ini, model yang telah dilatih diunggah ke *platform* roboflow dengan menuliskan source code *upload* model dari dokumentasi yang sudah disediakan oleh roboflow.

2. Pengembangan Sistem

Pengembangan sistem pada penelitian ini menggunakan model waterfall, yang mana terdiri dari beberapa tahapan. Adapun pembahasiantiap tahapan yang telah dilakukan dari penelitian ini yaitu sebagai berikut:

a. Analisis Kebutuhan

Tahap pertama yaitu tahap analisis kebutuhan. Tahap ini diperlukan untuk merinci kebutuhan yang akan digunakan dalam

pengembangan aplikasi. Analisis kebutuhan ini mencakup analisis kebutuhan data, kebutuhan sistem, serta kebutuhan fungsional. Dalam tahapan ini, kebutuhan untuk aplikasi deteksi penyakit dan hama pada buah kakao yang akan dikembangkan dikumpulkan, dipahami, dan diuraikan secara detail.

Analisis kebutuhan data mencakup identifikasi jenis data yang diperlukan, seperti citra jenis penyakit dan hama pada buah kakao. Pengumpulan data yang akurat sangat penting untuk melatih model deteksi hama dan penyakit pada buah kakao dengan tepat.

Analisis kebutuhan sistem melibatkan identifikasi infrastruktur teknis yang diperlukan untuk mendukung aplikasi. Ini mencakup perangkat keras untuk melakukan pengambilan gambar dan pemrosesan data, serta perangkat lunak yang digunakan untuk pengembangan aplikasi. Tahap analisis kebutuhan ini sangat penting karena menjadi dasar bagi desain dan pengembangan aplikasi selanjutnya. Dengan memahami dan merinci kebutuhan secara detail, pengembang dapat memastikan bahwa aplikasi yang dibuat sesuai dengan kebutuhan pengguna dan mampu memberikan solusi yang efektif untuk deteksi penyakit dan hama pada buah kakao.

b. Desain

Setelah mengumpulkan kebutuhan aplikasi, langkah berikutnya adalah tahap desain. Dalam tahap ini, digunakan model perancangan UML, yang mencakup empat jenis diagram UML: *Use Case*, *Activity*, *Sequence*, dan *Class Diagram*. Selain itu, juga digunakan desain antarmuka atau *User Interface* (UI), dengan merancang *Wireframe* sederhana untuk mempermudah pengembangan aplikasi.

Use Case Diagram digunakan untuk mengidentifikasi dan menggambarkan berbagai fungsi yang akan dilakukan oleh aplikasi, serta hubungan antara pengguna (aktor) dengan fungsi-fungsi

tersebut. Ini membantu dalam memahami interaksi pengguna dengan sistem dan memastikan semua kebutuhan fungsional telah tercakup.

Activity Diagram menggambarkan alur kerja atau aktivitas yang terjadi dalam aplikasi, menunjukkan langkah-langkah yang harus diambil dari awal hingga akhir dalam berbagai proses, seperti proses deteksi penyakit. Dengan *Activity Diagram*, setiap proses dijelaskan secara rinci dan efisien.

Sequence Diagram menggambarkan interaksi antara objek dalam urutan waktu tertentu, membantu memahami urutan kejadian dalam sistem. Dengan *Sequence Diagram*, interaksi antar komponen sistem dapat diatur dengan lancar dan sesuai dengan yang diharapkan.

Class Diagram menggambarkan struktur sistem dalam hal kelas-kelas dan hubungan antar kelas tersebut, membantu menentukan atribut dan metode yang dimiliki oleh setiap kelas. Dengan *Class Diagram*, struktur data dan logika aplikasi dapat terorganisir dengan baik sesuai dengan kebutuhan fungsional yang telah diidentifikasi.

Selain model UML, desain antarmuka atau *User Interface* (UI) juga menjadi fokus utama dalam tahap ini. *Wireframe* sederhana berfungsi sebagai blueprint visual untuk tata letak dasar elemen-elemen antarmuka pengguna, memastikan antarmuka pengguna intuitif dan mudah digunakan.

Dengan desain yang baik, aplikasi dapat dikembangkan dengan lebih efisien dan efektif, sesuai dengan kebutuhan pengguna yang telah diidentifikasi.

c. Implementasi

Implementasi yang pertama adalah implementasi model. Model dikembangkan dari mulai pengolahan dataset sampai dengan *upload* model. Setelah model selesai dikembangkan, selanjutnya adalah implementasi kedua yaitu implementasi sistem. Pada implementasi sistem, model yang sudah selesai di training dan menghasilkan *output best.pt*, kemudian menyimpan file *best.pt* tersebut kedalam *directory*.

kemudian file tersebut di *deploy* ke projek aplikasi website untuk pengembangan lebih lanjut. Setelah file *best.pt* model YOLO v8 tersebut berhasil *diintegrasikan* ke dalam aplikasi website, langkah berikutnya adalah melakukan pengujian untuk memastikan bahwa model berfungsi dengan baik di lingkungan website. Pengujian ini meliputi verifikasi akurasi prediksi, performa, dan responsivitas model ketika diakses melalui aplikasi website. Setelah pengujian selesai dan hasilnya memuaskan, aplikasi website yang mengandung model YOLO v8 ini siap untuk digunakan dan diuji oleh pengguna.

d. Pengujian

Pengujian merupakan langkah terakhir dalam proses pengembangan. Pengujian ini dilakukan untuk memverifikasi kualitas dari aplikasi yang dikembangkan. Penulis melakukan empat jenis pengujian: *Black Box Testing*, *White Box Testing*, *User Acceptance Testing* (UAT), dan Pengujian Lapangan. Dalam pengujian yang didapat dari *Black Box Testing* mendapatkan hasil 100% keberhasilan, sedangkan pengujian yang gagal mendapat 0% dari 3 responden dan 12 pengujian dilengkapi dengan masukan dari responden yang tercantum dalam lampiran. Berikutnya untuk pengujian *White Box* mendapatkan hasil yang sangat memuaskan dimana mendapatkan presentase 100% dengan pengujian yang tercapai, sedangkan untuk pengujian gagal mendapatkan presentase 0% dari value test yang terdiri dari 4 *path*. Selain itu, terdapat pengujian *User Acceptance Testing* (UAT) menghasilkan tingkat keberhasilan yang sangat baik, dengan persentase sebesar 90,8% dari lima responden yang menjawab 10 pertanyaan. Terakhir adapula pengujian lapangan yang memberikan hasil kesimpulan bahwa aplikasi dapat mendeteksi hama dan penyakit pada buah kakao dengan baik dan dapat bermanfaat bagi pengguna, khususnya petani.

BAB V

KESIMPULAN DAN SARAN

A. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan oleh penulis, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Dari proses pelatihan dataset hama dan penyakit pada buah kakao menggunakan algoritma YOLO v8 dengan 30 epoch dan jumlah dataset sebanyak 859, yang dibagi menjadi 602 data (70%) untuk pelatihan, 129 data (15%) untuk validasi dan 128 data (15%) untuk *testing*, didapatkan hasil nilai akurasi sebesar 94%. Hasil ini menunjukkan bahwa model yang dikembangkan memiliki kemampuan yang sangat baik dalam mendeteksi hama dan penyakit pada buah kakao. Meskipun demikian, ada ruang untuk perbaikan lebih lanjut guna meningkatkan akurasi dan keandalan model, seperti dengan menambah jumlah data pelatihan, atau menggunakan teknik augmentasi data. Secara keseluruhan, penelitian ini menunjukkan potensi yang sangat baik dari penggunaan YOLO v8 dalam aplikasi deteksi hama dan penyakit pada buah kakao berbasis website.
2. Pengujian *black box* dilakukan oleh 3 responden dengan total 12 pengujian. Beberapa halaman yang diuji meliputi navigasi, menu beranda, menu deteksi, menu artikel, menu tentang, *footer*. Hasil dari pengujian ini menunjukkan tingkat keberhasilan sebesar 100%, dengan tingkat kegagalan 0%.
3. Pengujian *White Box* mendapatkan hasil yang sangat memuaskan dimana mendapatkan presentase 100% dengan pengujian yang tercapai, sedangkan untuk pengujian gagal mendapatkan presentase 0% dari value test yang terdiri dari 4 *path*.
4. Pengujian *user acceptance testing* dilakukan oleh 5 responden dengan 2 diantaranya bersama para petani kakao menggunakan 10 pertanyaan.

Pertanyaan mencakup aspek kegunaan, kemudahan penggunaan, dan *User Interface* (UI). Hasil pengujian ini menunjukkan persentase sebesar 90,8% dengan kategori sangat baik.

5. Pengujian lapangan aplikasi deteksi hama dan penyakit pada buah kakao di salah satu perkebunan warga Desa Nglegi, Kec. Patuk, Kab. Gunung Kidul, Jawa Tengah, menunjukkan bahwa aplikasi yang dikembangkan oleh penulis mampu dengan baik mendeteksi hama dan penyakit buah kakao. Dimana dalam pengujian lapangan, aplikasi berhasil mendeteksi buah sehat dan busuk buah kakao yang umum menyerang tanaman kakao di perkebunan warga Desa Nglegi dengan mengambil foto langsung dari tempat lokasi. Para petani di sini umumnya mengatasi penyakit busuk buah tersebut dengan cara melakukan pembuangan pada buah yang terinfeksi secara hati-hati dan menerapkan pemangkasan menggunakan kultur teknis yang tepat dalam waktu sebulan sekali. Panen sering pada buah yang sudah masak dilakukan untuk mencegah kehilangan panen untuk menghindari terjadinya infeksi pada buah yang dapat menyebabkan busuk buah. Selain itu, tidak ditemukan adanya hama penggerek buah dan hama kepik penghisap buah dalam lokasi pengujian lapangan. Oleh karena itu, hasil dari adanya aplikasi berbasis website ini dinilai dapat bermanfaat bagi pengguna, khususnya petani kakao.
6. Aplikasi berbasis website ini berpotensi besar untuk dapat digunakan oleh petani dan agronomis dalam mendeteksi hama dan penyakit pada buah kakao secara cepat dan akurat, yang dapat membantu dalam pengambilan keputusan terkait penanganan tanaman.
7. Penelitian ini memberikan kontribusi penting dalam bidang pertanian digital, khususnya dalam pengembangan alat bantu berbasis teknologi untuk mendeteksi penyakit tanaman, yang dapat meningkatkan produktivitas dan efisiensi dalam pengelolaan tanaman kakao.

B. SARAN

Berdasarkan penelitian yang telah dilakukan, penulis memberikan beberapa saran sebagai pertimbangan untuk penelitian selanjutnya, yaitu:

1. Mengeksplorasi kemungkinan untuk mengembangkan aplikasi deteksi hama dan penyakit pada buah kakao berbasis android sehingga menambah kemudahan pengguna dalam melakukan akses aplikasi.
2. Melakukan penambahan fitur deteksi secara realtime pada aplikasi untuk penelitian selanjutnya supaya dapat dilakukan deteksi langsung di lapangan tanpa harus upload gambar/video.
3. Mengembangkan kembali jenis format pada file yang dapat diupload selain format jpg dan mp4. Contohnya seperti jpeg, png, dan lain sebagainya untuk kemudahan melakukan upload gambar/video.
4. Melakukan optimasi lebih lanjut terhadap proses training model deteksi hama dan penyakit pada buah kakao menggunakan algoritma YOLO v8 dengan menambahkan beberapa dataset yang lebih beragam untuk dapat meningkatkan akurasi dan keandalan dalam mendeteksi berbagai jenis hama dan penyakit pada buah kakao dengan menambahkan beberapa proses augmentasi data.
5. Melakukan optimasi kembali dalam penyempurnaan kode pemrograman dalam projek aplikasi supaya aplikasi dapat memberikan respon dengan lebih cepat dan lebih presisi dalam menampilkan hasil deteksi dari upload gambar/video.

DAFTAR PUSTAKA

- [1] E. Manurung, H. Marwan, and S. Mulyati, “Keparahan Beberapa Penyakit Pada Buah Kakao di Perkebunan Rakyat Kecamatan Kumpeh Kabupaten Muaro Jambi,” *Jurnal Agroecotania : Publikasi Nasional Ilmu Budidaya Pertanian*, vol. 5, no. 1, pp. 63–74, Dec. 2022, doi: 10.22437/agroecotania.v5i1.22828.
- [2] A. S. Sacita and M. Naim, “Tingkat Serangan Hama *Helopeltis* spp dan Penggerek Buah Kakao (PBK) Pada Beberapa Dosis Pemupukan Tanaman Kakao,” *Perbal: Jurnal Pertanian Berkelanjutan*, vol. 9, no. 3, pp. 202–207, 2021.
- [3] F. Ardiani, D. Nurjanah, and G. Noviana, “Identifikasi dan Teknis Pengendalian Hama pada Tanaman Kakao di Kabupaten Gunungkidul,” *Agritech: Jurnal Fakultas Pertanian Universitas Muhammadiyah Purwokerto*, vol. 25, no. 1, pp. 47–54, 2023.
- [4] U. Khairul, T. Trizelia, and R. Reflin, “Pemberdayaan kelompok tani melalui pelatihan pengendalian hama dan penyakit tanaman kakao di Kanagarian Campago Kabupaten Padang Pariaman,” *Buletin Ilmiah Nagari Membangun*, vol. 1, no. 4, pp. 88–95, 2018.
- [5] A. Hafiz and V. Verawati, “Sistem Pakar Penyakit Buah Kakao Untuk Peningkatan Hasil Panen Kakao Menggunakan Metode Case Base Reasoning (Cbr) Berbasis Web Mobile,” *Jurnal Informasi dan Komputer*, vol. 9, no. 2, pp. 89–94, Oct. 2021, doi: 10.35959/jik.v9i2.226.
- [6] N. J. Hayati, D. Singasatia, and M. R. Muttaqin, “Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 untuk Menghitung Kendaraan,” *Komputa: Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 2, pp. 91–99, Nov. 2023, doi: 10.34010/komputa.v12i2.10654.
- [7] A. Saepulloh and M. Adeyadi, “Aplikasi Scanner Berbasis Android Untuk Menampilkan Data Id Card Menggunakan Barcode,” *Jurnal Manajemen dan Teknik Informatika (JUMANTAKA)*, vol. 3, no. 1, 2019.
- [8] B. Huda and B. Priyatna, “Penggunaan Aplikasi Content Management System (CMS) Untuk Pengembangan Bisnis Berbasis E-commerce,” *SYSTEMATICS*, vol. 1, no. 2, p. 81, Dec. 2019, doi: 10.35706/sys.v1i2.2076.
- [9] H. Rummujib, “Aplikasi Penelitian Dan Pengabdian Masyarakat Pada LPPM Universitas Nurdin Hamzah Berbasis Mobile,” *Jurnal Karya Informatika (KARTIKA)*, vol. 2, no. 2, pp. 44–49, 2021.

- [10] Z. H. Batubara, Y. Hamonangan, M. Arfan, and A. Hidayatno, "Perancangan Sistem Deteksi Pelanggaran Penggunaan Helm Dengan Metode Deep Learning Menggunakan Yolov5 Ultralytic," *Transient: Jurnal Ilmiah Teknik Elektro*, vol. 13, no. 1, pp. 11–20, Mar. 2024, doi: 10.14710/transient.v13i1.11-20.
- [11] P. D. Kusuma, *Machine Learning Teori, Program, dan Studi Kasus*. Deepublish, 2020.
- [12] R. Ardianto and S. K. Wibisono, "Analisis Deep Learning Metode Convolutional Neural Network Dalam Klasifikasi Varietas Gandum," *Jurnal Kolaboratif Sains*, vol. 6, no. 12, pp. 2081–2092, 2023.
- [13] I. Munadhif, D. H. Fathoni, and M. A. Jamiin, "Pengendalian Cctv Menggunakan You Only Look Once (Yolo)," in *Prosiding Seminar Nasional Terapan Riset Inovatif (SENTRINOV)*, 2020, pp. 958–965.
- [14] Z. Yuan *et al.*, "YOLOv8-ACU: improved YOLOv8-pose for facial acupoint detection," *Front Neurobot*, vol. 18, Feb. 2024, doi: 10.3389/fnbot.2024.1355857.
- [15] H. Gibran, B. Purnama, and G. Kosala, "Pengoimanasian Pengukuran Kepadatan Jalan Raya Dengan Cctv Menggunakan Metode Yolov8: Optimizing Highway Density Measurement with CCTV Using the Yolov8 Method," *Technomedia Journal*, vol. 9, no. 1 Juni, pp. 31–45, 2024.
- [16] M. I. Saad, *Otodidak Web Programming: Membuat Website Edutainment*. Elex Media Komputindo, 2020.
- [17] I. Salamah, "Evaluasi usability website polsri dengan menggunakan system usability scale," *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, vol. 8, no. 3, pp. 176–183, 2019.
- [18] Hadi Supriyanto, Sarosa Castrena Abadi, and Aliffa Shalsabilah, "Deteksi Helm Keselamatan Menggunakan Jetson Nano dan YOLOv7," *Journal of Applied Computer Science and Technology*, vol. 5, no. 1, pp. 1–8, Feb. 2024, doi: 10.52158/jacost.v5i1.637.
- [19] R. K. Ngantung and M. A. I. Pakereng, "Model Pengembangan Sistem Informasi Akademik Berbasis User Centered Design Menerapkan Framework Flask Python," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 3, p. 1052, Jul. 2021, doi: 10.30865/mib.v5i3.3054.
- [20] C. Ningki and N. P., "Implementasi Aplikasi Penjualan Produk Tradisional Berbasis Website Menggunakan Metode Waterfall," *Informatik : Jurnal*

- Ilmu Komputer*, vol. 19, no. 2, pp. 107–114, Sep. 2023, doi: 10.52958/iftk.v19i2.6149.
- [21] A. Fadillah, D. I. Mona, and F. Ramadhan, “Rancang Bangun Sistem Informasi Perpustakaan dengan Menggunakan Metode Waterfall di IKesT Muhammadiyah Palembang,” *Jurnal Pengembangan Sistem Informasi dan Informatika*, vol. 4, no. 3, pp. 1–8, Feb. 2024, doi: 10.47747/jpsii.v4i3.1683.
- [22] A. Adensa, K. Raihan, R. Faisal Rafi, I. Richwandi Putra, and F. Azizah, “Pengembangan Web Dinas Perpustakaan Dan Arsip Berbasis Laravel Framework Pada Dpad Kota Tangerang,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 6, pp. 3877–3883, Jan. 2024, doi: 10.36040/jati.v7i6.7840.
- [23] A. F. Prasetya and U. L. D. P. Sintia, “Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modelling,” *Jikti (Jurnal Ilmiah Komputer Terapan dan Informasi)*, vol. 1, no. 1, pp. 14–18, 2022.
- [24] A. Primadewi, M. TI, S. T. Setiya Nugroho, S. S. Maimunah, and M. Kom, “Sistem Informasi Terintegrasi,” Jan. 2022.
- [25] S. Nabila, A. R. Putri, A. Hafizhah, F. H. Rahmah, and R. Muslikhah, “Pemodelan Diagram UML Pada Perancangan Sistem Aplikasi Konsultasi Hewan Peliharaan Berbasis Android (Studi Kasus: Alopel),” *Jurnal Ilmu Komputer dan Bisnis*, vol. 12, no. 2, pp. 130–139, 2021.
- [26] W. Lnu, “Materi 5-Activity Diagram,” 2020.
- [27] D. W. T. Putra and R. Andriani, “Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD,” *Jurnal TeknoIf*, vol. 7, no. 1, p. 32, Apr. 2019, doi: 10.21063/jtif.2019.V7.1.32-39.
- [28] F. A. Novianto and H. Purwanto, “Perancangan Sistem Informasi Land Transportation Assistance Taxi Puskopau Pada Bandara Xyz,” *JSI (Jurnal sistem Informasi) Universitas Suryadarma*, vol. 9, no. 2, pp. 29–40, 2022.
- [29] A. S. Lubis and M. P. A. Ginting, “Pengujian Aplikasi Berbasis Web Data Ska Menggunakan Metode Black Box Testing,” *COSMIC Jurnal Teknik*, pp. 41–48, 2024.
- [30] D. Destiarini, A. Rahman, and K. Sumartayasa, “Analisa Kualitas Website BPJS Kesehatan Dengan Metode WebQual 4.0 Dan User Acceptance Testing Di Wilayah Kabupaten Ogan Komering Ulu.,” *JURNAL MEDIA INFOTAMA*, vol. 19, no. 2, pp. 237–243, Oct. 2023, doi: 10.37676/jmi.v19i2.3911.

- [31] R. F. Wijaya and R. B. Utomo, “Metode Waterfall Dalam Rancang Bangun Sistem Informasi Manajemen Kegiatan Masjid Berbasis Web,” *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 3, no. 5, pp. 563–571, 2023.
- [32] S. Putra, Y. Ferry, and R. Harni, “Pengendalian penyakit busuk buah kakao menggunakan Trichoderma dan pupuk Kalium,” *Kultivasi*, vol. 21, no. 2, Aug. 2022, doi: 10.24198/kultivasi.v21i2.36807.
- [33] V. C. Lea, “Hama Dan Penyakit Penting Tanaman Kakao Di Kabupaten Nagekeo, Provinsi Ntt,” *Jurnal Agrotek Tropika*, vol. 10, no. 4, p. 509, Oct. 2022, doi: 10.23960/jat.v10i4.5860.
- [34] J. Rimbing and R. Engka, “Pengenalan Hama-Hama Tanaman Kakao dan Pengendaliannya,” 2022, *CV. Patra Media Grafindo Bandung*.
- [35] Y. Suherlina, Y. Yaherwandi, and S. C. Efendi, “Sebaran dan tingkat serangan hama penggerek buah kakao (*Conopomorpha cramerella* Snellen) pada lahan bukaan baru di kabupaten dharmasraya,” *Jurnal agronida*, vol. 6, no. 1, pp. 44–54, 2020.
- [36] J. J. Lawalata, D. M. D. Tatuhey, H. Masbaitubun, M. I. A. Linggi, and P. A. Merahabia, “Kerusakan Hama Penghisap Buah Kakao (*Helopeltis* Spp) Pada Tanaman Kakao Di Kampung Sereh Dan Yahim Distrik Sentani,” *Jurnal Pertanian Terpadu Santo Thomas Aquinas*, vol. 3, no. 1, pp. 31–34, 2024.
- [37] D. Supriyatdi, D. R. Lovantineya, and B. Utoyo, “Potensi Ekstrak Serai Wangi dan Daun Mengkudu dalam Pengendalian Hama Penghisap Buah Kakao (<i>Helopeltis</i> spp.),” *Jurnal AGROSAINS dan TEKNOLOGI*, vol. 8, no. 1, p. 11, Jul. 2023, doi: 10.24853/jat.8.1.11-19.

LAMPIRAN

Lampiran 1 Lembar Bimbingan Dosen Pembimbing



UNIVERSITAS PGRI SEMARANG

FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidodadi Timur Nomor 24 Dr. Cipto, Semarang – Indonesia 50125

Telp. (024)8316377, Faks. (024)8448217, E-mail : upgrisng@gmail.com, Homepage : www.upgrisng.ac.id

LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : Aldhita Dwi Lestari
NPM : 20670133
Program Studi : Informatika
Judul Skripsi : Aplikasi Deteksi Hama dan Penyakit pada Buah Kakas Menggunakan YOLO V8 Berbasis Website
Dosen Pembimbing I : Febrian Murti Dewanto, SE., M.Kom.
Dosen Pembimbing II : Noora Qotrun Nada,

No.	Hari Tanggal	Uraian Bimbingan	Paraf
1.	Senin/12-02-2024	Konsultasi judul skripsi	<i>[Signature]</i>
2.	Selasa/20-02-2024	Acc judul skripsi	<i>[Signature]</i>
3.	Kamis/14-03-2024	Bab 1	<i>[Signature]</i>
4.	Rabu/20-03-2024	Bab 2 dan revisian bab 1	<i>[Signature]</i>
5.	Selasa/8-05-2024	Bab 3 dan revisian bab 2	<i>[Signature]</i>
6.	Jumat/28-06-2024	Bab 4 dan revisian bab 3	<i>[Signature]</i>
7.	Rabu/09-07-2024 10/7	Revisi bab 4 & bimbingan proyek	<i>[Signature]</i>
8.		Acc Skripsi	<i>[Signature]</i>

Dosen Pembimbing I,

[Signature]
Febrian Murti Dewanto, SE., M.Kom.
NIDN. 0606027801

Mahasiswa,

[Signature]
Aldhita Dwi Lestari
NPM. 20670133



UNIVERSITAS PGRI SEMARANG

FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidodadi Timur Nomor 24 Dr. Cipto, Semarang – Indonesia 50125

Telp. (024)8316377, Faks. (024)8448217, E-mail : upgrismg@gmail.com, Homepage : www.upgrismg.ac.id

LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : Aldhita Dwi Lestari
NPM : 20670133
Program Studi : Informatika
Judul Skripsi : Aplikasi Deteksi Hama dan Penyakit pada Buah Kakao Menggunakan YOLO v8 Berbasis Website

Dosen Pembimbing I : Febrian Murti Dewanto, SE., M. Kom.
Dosen Pembimbing II : Noora Qotrun Nada, S.T., M. Eng.

No.	Hari Tanggal	Uraian Bimbingan	Paraf
1.	Senin / 19-02-2024	Konsultasi Judul Skripsi	f
2.	Kamis / 20-02-2024	ACC Judul skripsi	f
3.	Kamis / 28-03-2024	Bab 1	f
4.	Kamis / 21-05-2024	Bab 2 dan revisian bab 1	f
5.	Selasa / 07-05-2024	Bab 3 dan revisian bab 2	f
6.	Kamis / 09-07-2024	Bab 4 dan revisian bab 3	f
7.	Selasa / 09-07-2024	Bimbingan Bab 5 & penuntasan ke 4	f
8.	Rabu / 10-07-2024	ACC. dilanjutkan ke sidang TA	f

Dosen Pembimbing II,

Noora Qotrun Nada, S.T., M.Eng.
NIDN. 0626028201

Mahasiswa,

Aldhita Dwi Lestari
NPM. 20670133

Lampiran 2 Kuesioner Pengujian *Black Box*

Kuesioner Pengujian Black Box pada Aplikasi Deteksi Hama dan Penyakit pada Buah Kakao menggunakan YOLO v8 Berbasis Website

Nama Penguji : *Bambang Agus H.*

Tanggal Pengujian : *3/7/2024.*

Nama Pengujian	Test Case	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Diterima	Ditolak
Navigasi	User menekan beberapa tombol pada navigasi.	User dapat masuk ke halaman navigasi yang dituju.	Aplikasi berhasil menampilkan halaman navigasi yang dituju.	✓	
Menu Beranda	User melakukan gulir atau scrolling kebawah.	User dapat melihat beberapa informasi yang ada pada halaman beranda.	Aplikasi berhasil menampilkan beberapa informasi dalam halaman beranda.	✓	
	User menekan tombol mulai deteksi.	User dapat masuk ke halaman deteksi.	Aplikasi berhasil menampilkan halaman deteksi.	✓	
	User menekan tombol kunjungi artikel.	User dapat masuk ke halaman artikel.	Aplikasi berhasil menampilkan halaman artikel.	✓	
Menu Deteksi	User menekan tombol deteksi.	User dapat masuk ke halaman deteksi.	Aplikasi berhasil menampilkan halaman deteksi.	✓	
	User menekan	User dapat melihat tampilan file	Aplikasi berhasil menampilkan	✓	

	tombol choose file.	manager dan dapat menginputkan gambar/video.	file manager dan dapat menyimpan hasil inputan gambar/video.	✓	
	User menekan tombol upload.	User dapat mengupload gambar/video inputan untuk dapat melihat hasil deteksi.	Aplikasi berhasil mengupload gambar/video inputan dan menampilkan hasil deteksi.	✓	
Menu Artikel	User menekan tombol artikel.	User dapat masuk ke halaman artikel.	Aplikasi berhasil menampilkan halaman artikel.	✓	
	User menekan kategori artikel	User dapat melihat judul artikel sesuai kategori	Aplikasi berhasil menampilkan judul artikel sesuai kategori	✓	
	User memilih judul artikel	User dapat melihat dan membaca isi artikel yang dituju	Aplikasi berhasil menampilkan halaman isi artikel yang dituju	✓	
Menu Tentang	User menekan tombol tentang.	User dapat masuk ke dalam halaman tentang.	Aplikasi berhasil menampilkan halaman tentang.	✓	
Footer	User dapat menekan beberapa link navigasi yang ada di footer	User dapat masuk ke halaman navigasi yang dituju.	Aplikasi berhasil menampilkan halaman navigasi yang dituju.	✓	

Saran dari Penguji:

- Beberapa link detail disemakan.

Semarang, 03 Juli 2024



NIDN 0601088201.

Kuesioner Pengujian Black Box pada Aplikasi Deteksi Hama dan Penyakit pada Buah Kakao menggunakan YOLO v8 Berbasis Website

Nama Penguji : *Aris Thi Jalen H.*

Tanggal Pengujian : *3-7-24.*


Nama Pengujian	Test Case	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Diterima	Ditolak
Navigasi	<i>User menekan beberapa tombol pada navigasi.</i>	<i>User dapat masuk ke halaman navigasi yang dituju.</i>	<i>Aplikasi berhasil menampilkan halaman navigasi yang dituju.</i>	✓	
Menu Beranda	<i>User melakukan gulir atau scrolling kebawah.</i>	<i>User dapat melihat beberapa informasi yang ada pada halaman beranda.</i>	<i>Aplikasi berhasil menampilkan beberapa informasi dalam halaman beranda.</i>	✓	
	<i>User menekan tombol mulai deteksi.</i>	<i>User dapat masuk ke halaman deteksi.</i>	<i>Aplikasi berhasil menampilkan halaman deteksi.</i>	✓	
	<i>User menekan tombol kunjungi artikel.</i>	<i>User dapat masuk ke halaman artikel.</i>	<i>Aplikasi berhasil menampilkan halaman artikel.</i>	✓	
Menu Deteksi	<i>User menekan tombol deteksi.</i>	<i>User dapat masuk ke halaman deteksi.</i>	<i>Aplikasi berhasil menampilkan halaman deteksi.</i>	✓	
	<i>User menekan</i>	<i>User dapat melihat tampilan file</i>	<i>Aplikasi berhasil menampilkan</i>	✓	

	tombol choose file.	manager dan dapat menginputkan gambar/video.	file manager dan dapat menyimpan hasil inputan gambar/video.	✓	
	User menekan tombol upload.	User dapat mengupload gambar/video inputan untuk dapat melihat hasil deteksi.	Aplikasi berhasil mengupload gambar/video inputan dan menampilkan hasil deteksi.	✓	
Menu Artikel	User menekan tombol artikel.	User dapat masuk ke halaman artikel.	Aplikasi berhasil menampilkan halaman artikel.	✓	
	User menekan kategori artikel	User dapat melihat judul artikel sesuai kategori	Aplikasi berhasil menampilkan judul artikel sesuai kategori	✓	
	User memilih judul artikel	User dapat melihat dan membaca isi artikel yang dituju	Aplikasi berhasil menampilkan halaman isi artikel yang dituju	✓	
Menu Tentang	User menekan tombol tentang.	User dapat masuk ke dalam halaman tentang.	Aplikasi berhasil menampilkan halaman tentang.	✓	
Footer	User dapat menekan beberapa link navigasi yang ada di footer	User dapat masuk ke halaman navigasi yang dituju.	Aplikasi berhasil menampilkan halaman navigasi yang dituju.	✓	

Saran dari Penguji:

- bisa di kembangkan real time dgn laporan
- bisa tambahkan lebih dari dua jenis data

Semarang, 03 - Juli 2024


Ant. H. Jolani.....
NIDN ~~15820619048202~~

Kuesioner Pengujian Black Box pada Aplikasi Deteksi Hama dan Penyakit pada Buah Kakao menggunakan YOLO v8 Berbasis Website

Nama Penguji : Ramadhan Renaldy, S.kom, M.kom

Tanggal Pengujian : 2 Juli 2024

Nama Pengujian	Test Case	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Diterima	Ditolak
Navigasi	User menekan beberapa tombol pada navigasi.	User dapat masuk ke halaman navigasi yang dituju.	Aplikasi berhasil menampilkan halaman navigasi yang dituju.	✓	
Menu Beranda	User melakukan gulir atau scrolling kebawah.	User dapat melihat beberapa informasi yang ada pada halaman beranda.	Aplikasi berhasil menampilkan beberapa informasi dalam halaman beranda.	✓	
	User menekan tombol mulai deteksi.	User dapat masuk ke halaman deteksi.	Aplikasi berhasil menampilkan halaman deteksi.	✓	
	User menekan tombol kunjungi artikel.	User dapat masuk ke halaman artikel.	Aplikasi berhasil menampilkan halaman artikel.	✓	
Menu Deteksi	User menekan tombol deteksi.	User dapat masuk ke halaman deteksi.	Aplikasi berhasil menampilkan halaman deteksi	✓	
	User menekan	User dapat melihat tampilan file	Aplikasi berhasil menampilkan	✓	

	tombol choose file.	manager dan dapat menginputkan gambar/video.	file manager dan dapat menyimpan hasil inputan gambar/video.	✓	
	User menekan tombol upload.	User dapat mengupload gambar/video inputan untuk dapat melihat hasil deteksi.	Aplikasi berhasil mengupload gambar/video inputan dan menampilkan hasil deteksi.	✓	
Menu Artikel	User menekan tombol artikel.	User dapat masuk ke halaman artikel.	Aplikasi berhasil menampilkan halaman artikel.	✓	
	User menekan kategori artikel	User dapat melihat judul artikel sesuai kategori	Aplikasi berhasil menampilkan judul artikel sesuai kategori	✓	
	User memilih judul artikel	User dapat melihat dan membaca isi artikel yang dituju	Aplikasi berhasil menampilkan halaman isi artikel yang dituju	✓	
Menu Tentang	User menekan tombol tentang.	User dapat masuk ke dalam halaman tentang.	Aplikasi berhasil menampilkan halaman tentang.	✓	
Footer	User dapat menekan beberapa link navigasi yang ada di footer	User dapat masuk ke halaman navigasi yang dituju.	Aplikasi berhasil menampilkan halaman navigasi yang dituju.	✓	

Saran dari Penguji:

Akan lebih bagus jika deteksi dapat dilakukan secara realtime.

Semarang, 2 Juli 2024



Ramadhan Renaldi

NIDN NPP. 299901659

Lampiran 3 Pengujian UAT

Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi Deteksi Hama Dan Penyakit Pada Buah Kakao Menggunakan YOLOv8 Berbasis Website

Nama Penguji : *Paيران*

Tanggal Pengujian : *5 Juli 2024*

No	Pertanyaan	Skor				
		1	2	3	4	5
Aspek Kegunaan						
1.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat bermanfaat bagi pengguna (<i>user</i>), khususnya petani kakao?					✓
2.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat membantu pengguna dalam melakukan deteksi jenis hama dan penyakit pada buah kakao secara efektif?				✓	
3.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao memberikan informasi mengenai jenis hama dan penyakit pada buah kakao				✓	

	lengkap dengan gejala beserta cara pengendalian yang berguna bagi pengguna?					
Aspek Kemudahan Pengguna						
4.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao mudah dioperasikan?			√		
5.	Apakah menu-menu yang ada pada aplikasi deteksi hama dan penyakit pada buah kakao mudah dipahami oleh pengguna?				√	
6.	Apakah setiap menu yang ada dalam aplikasi deteksi hama dan penyakit pada buah kakao dapat diakses dengan mudah?				√	
7.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat berjalan sesuai yang diharapkan?					√
Aspek Tampilan (User Interface)						
8.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao memiliki tampilan yang menarik?					√
9.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao					

	memiliki tema warna yang enak dilihat dan jelas?			V		.
10.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao perlu dikembangkan lagi?					V

Keterangan:

- 1 = Tidak Setuju
- 2 = Kurang Setuju
- 3 = Cukup Setuju
- 4 = Setuju
- 5 = Sangat Setuju

Gumung Kidul 5 Juli 2024



.....
Pak Páran
(Ketua Kelompok tani Ngudi Malyo)

Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi Deteksi Hama Dan Penyakit Pada Buah Kakao Menggunakan YOLOv8 Berbasis Website

Nama Penguji : Kasiyanti
 Tanggal Pengujian : 6 Juli 2024

No	Pertanyaan	Skor				
		1	2	3	4	5
Aspek Kegunaan						
1.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat bermanfaat bagi pengguna (<i>user</i>), khususnya petani kakao?					✓
2.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat membantu pengguna dalam melakukan deteksi jenis hama dan penyakit pada buah kakao secara efektif?					✓
3.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao memberikan informasi mengenai jenis hama dan penyakit pada buah kakao				✓	

	lengkap dengan gejala beserta cara pengendalian yang berguna bagi pengguna?					
Aspek Kemudahan Pengguna						
4.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao mudah dioperasikan?				✓	
5.	Apakah menu-menu yang ada pada aplikasi deteksi hama dan penyakit pada buah kakao mudah dipahami oleh pengguna?					✓
6.	Apakah setiap menu yang ada dalam aplikasi deteksi hama dan penyakit pada buah kakao dapat diakses dengan mudah?					✓
7.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat berjalan sesuai yang diharapkan?				✓	
Aspek Tampilan (<i>User Interface</i>)						
8..	Apakah aplikasi deteksi hama dan penyakit pada buah kakao memiliki tampilan yang menarik?					✓
9.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao					✓

	memiliki tema warna yang enak dilihat dan jelas?					
10.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao perlu dikembangkan lagi?				✓	

Keterangan:

1 = Tidak Setuju

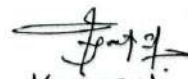
2 = Kurang Setuju

3 = Cukup Setuju

4 = Setuju

5 = Sangat Setuju

Gunung Kidul 6 Juli 2024



Kasiyanti

Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi Deteksi Hama Dan Penyakit Pada Buah Kakao Menggunakan YOLOv8 Berbasis Website

Nama Penguji : Nanta Angraeni

Tanggal Pengujian : 8 Juli 2024

No	Pertanyaan	Skor				
		1	2	3	4	5
Aspek Kegunaan						
1.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat bermanfaat bagi pengguna (<i>user</i>), khususnya petani kakao?				✓	
2.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat membantu pengguna dalam melakukan deteksi jenis hama dan penyakit pada buah kakao secara efektif?					✓
3.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao memberikan informasi mengenai jenis hama dan penyakit pada buah kakao					✓


	lengkap dengan gejala beserta cara pengendalian yang berguna bagi pengguna?						✓
Aspek Kemudahan Pengguna							
4.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao mudah dioperasikan?						✓
5.	Apakah menu-menu yang ada pada aplikasi deteksi hama dan penyakit pada buah kakao mudah dipahami oleh pengguna?					✓	
6.	Apakah setiap menu yang ada dalam aplikasi deteksi hama dan penyakit pada buah kakao dapat diakses dengan mudah?					.	✓
7.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat berjalan sesuai yang diharapkan?						✓
Aspek Tampilan (User Interface)							
8..	Apakah aplikasi deteksi hama dan penyakit pada buah kakao memiliki tampilan yang menarik?						✓
9.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao						

	memiliki tema warna yang enak dilihat dan jelas?					✓
10.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao perlu dikembangkan lagi?					✓

Keterangan:

- 1 = Tidak Setuju
- 2 = Kurang Setuju
- 3 = Cukup Setuju
- 4 = Setuju
- 5 = Sangat Setuju

Semarang
08 Juli 2024


Winda Angraeni

Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi Deteksi Hama Dan Penyakit Pada Buah Kakao Menggunakan YOLOv8 Berbasis Website

Nama Penguji : Faridatul Zuhriyah

Tanggal Pengujian : 8 Juli 2024

No	Pertanyaan	Skor				
		1	2	3	4	5
Aspek Kegunaan						
1.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat bermanfaat bagi pengguna (<i>user</i>), khususnya petani kakao?					✓
2.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat membantu pengguna dalam melakukan deteksi jenis hama dan penyakit pada buah kakao secara efektif?				✓	
3.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao memberikan informasi mengenai jenis hama dan penyakit pada buah kakao				✓	

	lengkap dengan gejala beserta cara pengendalian yang berguna bagi pengguna?					
Aspek Kemudahan Pengguna						
4.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao mudah dioperasikan?			✓		
5.	Apakah menu-menu yang ada pada aplikasi deteksi hama dan penyakit pada buah kakao mudah dipahami oleh pengguna?					✓
6.	Apakah setiap menu yang ada dalam aplikasi deteksi hama dan penyakit pada buah kakao dapat diakses dengan mudah?					✓
7.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat berjalan sesuai yang diharapkan?				✓	
Aspek Tampilan (User Interface)						
8..	Apakah aplikasi deteksi hama dan penyakit pada buah kakao memiliki tampilan yang menarik?					✓
9.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao					

	memiliki tema warna yang enak dilihat dan jelas?					✓
10.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao perlu dikembangkan lagi?					✓

Keterangan:

- 1 = Tidak Setuju
- 2 = Kurang Setuju
- 3 = Cukup Setuju
- 4 = Setuju
- 5 = Sangat Setuju

Kendal, 8 Juli 2024

[Handwritten Signature]
Faridatul Zahriyah

Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi Deteksi Hama Dan Penyakit Pada Buah Kakao Menggunakan YOLOv8 Berbasis Website

Nama Penguji : Faizlotun Nikmah

Tanggal Pengujian : 8 Juli 2024

No	Pertanyaan	Skor				
		1	2	3	4	5
Aspek Kegunaan						
1.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat bermanfaat bagi pengguna (<i>user</i>), khususnya petani kakao?					✓
2.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat membantu pengguna dalam melakukan deteksi jenis hama dan penyakit pada buah kakao secara efektif?				✓	
3.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao memberikan informasi mengenai jenis hama dan penyakit pada buah kakao					✓

	lengkap dengan gejala beserta cara pengendalian yang berguna bagi pengguna?					
Aspek Kemudahan Pengguna						
4.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao mudah dioperasikan?				✓	
5.	Apakah menu-menu yang ada pada aplikasi deteksi hama dan penyakit pada buah kakao mudah dipahami oleh pengguna?					✓
6.	Apakah setiap menu yang ada dalam aplikasi deteksi hama dan penyakit pada buah kakao dapat diakses dengan mudah?					✓
7.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao dapat berjalan sesuai yang diharapkan?				✓	
Aspek Tampilan (User Interface)						
8.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao memiliki tampilan yang menarik?					✓
9.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao					

	memiliki tema warna yang enak dilihat dan jelas?				✓	
10.	Apakah aplikasi deteksi hama dan penyakit pada buah kakao perlu dikembangkan lagi?					✓

Keterangan:

- 1 = Tidak Setuju
- 2 = Kurang Setuju
- 3 = Cukup Setuju
- 4 = Setuju
- 5 = Sangat Setuju

Semarang, 8 Juli 2024

[Handwritten Signature]

Faidlotun Nikmah


Lampiran 4 Lembar Revisi Ujian Skripsi/Tugas Akhir

LEMBAR REVISI UJIAN SKRIPSI

Nama Mahasiswa : Aldhita Dwi Lestari
N P M : 20670133
Judul : Aplikasi Deteksi Hama dan Penyakit pada Buah Kakao
Menggunakan YOLOv8 Berbasis Website

No	Uraian Revisi	Keterangan
1.	Tahap Pre Processing dikembangkan	
2.	Evaluasi Model ?	
3.	Hasil Pengujian ?	
4.	Hasil Train ?	
5.	Konfigurasi Yolo ?	

Pengesahan Penguji I


Khoirya Latifah, S. Kom., M. Kom.
NIP/NPP. 147801434

*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi

LEMBAR REVISI UJIAN SKRIPSI/TUGAS AKHIR

Nama Mahasiswa : Aldhita Dwi Lestari
N P M : 20670133
Judul : Aplikasi Deteksi Hama dan Penyakit pada Buah Kakao
Menggunakan YOLOv8 Berbasis Website

No	Uraian Revisi	Keterangan
1	Pembahasan perbaikan : analisis awal. Selamat !	Revisi acc <i>[Signature]</i> 22/9/2024.

Pengesahan Penguji II

[Signature]
Noora Qotrun Nada, S.T., M.ENG
NIP/NPP. 158201485

*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi

LEMBAR REVISI UJIAN SKRIPSI/TUGAS AKHIR

Nama Mahasiswa : Aldhita Dwi Lestari
 N P M : 20670133
 Judul : Aplikasi Deteksi Hama dan Penyakit pada Buah Kakao Menggunakan YOLOv8 Berbasis Website

No	Uraian Revisi	Keterangan
1	Kerangka Berfikir ✓	ae
2	Use case ✓	22/7/2024 ✓
3	White Box Test ? ✓	
4	Belum Ada Pembahasan Pemodelan Delta.	
5	Hama — Pemodelan Data ✓ — Pengembangan Sistem	
6	Alasan 70% , 15% , 15% ? ✓	
7	Epoch 30 ? Alasan . ✓	

Pengesahan Penguji III



Bambang Agus H. S. Kom, M. Kom
 NIP/NPP. 148201433

*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi