APLIKASI DETEKSI PENYAKIT PADA DAUN TANAMAN SINGKONG MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO) V8 BERBASIS ANDROID

SKRIPSI

TEGAR RAMADHANI

NPM 20670038

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK DAN INFORMATIKA

UNIVERSITAS PGRI SEMARANG

2024

APLIKASI DETEKSI PENYAKIT PADA DAUN TANAMAN SINGKONG MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO) V8 BERBASIS ANDROID

SKRIPSI

Diajukan kepada Fakultas Teknik dan Informatika Universitas PGRI Semarang untuk Penyusunan Skripsi

TEGAR RAMADHANI

NPM 20670038

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK DAN INFORMATIKA

UNIVERSITAS PGRI SEMARANG

2024

SKRIPSI

APLIKASI DETEKSI PENYAKIT PADA DAUN TANAMAN SINGKONG MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO) V8 BERBASIS ANDROID

Disusun dan diajukan oleh

TEGAR RAMADHANI

NPM 20670038

Telah disetujui oleh pembimbing untuk dilanjutkan untuk menempuh ujian skripsi pada tanggal 03 Juni 2024

Pembimbing Utama,	Pembimbing Pendamping,
Febrian Murti D, SE., M. Kom	Aris Tri Jaka H, S.Kom., M.Kom
NIDN. 0606027801	NIDN. 0619048202

SKRIPSI

APLIKASI DETEKSI PENYAKIT PADA DAUN TANAMAN SINGKONG MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO) V8 BERBASIS ANDROID

Disusun dan diajukan oleh

TEGAR RAMADHANI

NPM 20670038

Telah dipertahankan di depan Dewan Penguji pada tanggal 01 Juli 2024 dan dinyatakan telah memenuhi syarat Dewan Penguji

Ketua,	**Sekertaris,**
Ihnu Toto Husodo, S.T., M.T	**Bambang Agus H, S.Kom., M.Kom**
NIDN. 0602126902	**NIDN. 0601088201**
Penguji I,	**Penguji II,**
Febrian Murti D, SE., M. Kom	**Aris Tri Jaka H, S.Kom., M.Kom**
NIDN. 0606027801	**NIDN. 0619048202**

Penguji III,

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

NIDN. 0601088201

MOTTO DAN PERSEMBAHAN

"It always seems impossible until it's done."

-Nelson Mandela-

"A ship is always safe at the shore, but that's not what it's built for."

-Albert Einstein-

"I am a slow walker, but I never walk back."

-Abraham Lincoln-

Persembahan:

Saya persembahkan skripsi / tugas akhir ini untuk:

1. Kedua orang tua saya.
2. Adik saya.
3. Para guru yang telah memberikan ilmu kepada saya.
4. Teman-teman yang memberi semangat kepada saya.

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	:	Tegar Ramadhani
NPM	:	20670038
Program Studi	:	Informatika
Fakultas	:	Teknik dan Informatika

Menyatakan dengan sebenarnya bahwa skripsi yang saya buat ini benar-benar merupakan hasil karya sendiri, bukan plagiarisme. Apabila pada kemudian hari skripsi ini terbukti hasil plagiarisme, saya bersedia menerima sanksi atas perbuatan tersebut.

Semarang, 02 Juni 2024
Yang membuat pernyataan,

Tegar Ramadhani
20670038

ABSTRAK

Singkong, atau *Manihot esculenta*, merupakan salah satu tanaman penting dalam pertanian tropis, terutama di wilayah Asia, Afrika, dan Amerika Latin. Daun singkong sering kali menjadi sasaran penyakit yang dapat menyebabkan penurunan hasil panen yang signifikan. Di era modern ini, pengetahuan dapat dicari dengan mudah melalui internet, terlebih lagi dengan pesatnya perkembangan teknologi. Salah satu teknologi yang sedang populer adalah *Artificial Intelligence* (AI) atau kecerdasan buatan. AI tidak hanya populer, tetapi juga membawa dampak besar dalam berbagai aspek kehidupan, termasuk dalam deteksi penyakit tanaman. Dengan kehadiran AI, masyarakat dapat lebih efektif dalam deteksi penyakit pada tanaman khususnya tanaman singkong. Teknologi ini mempermudah dalam mengidentifikasi berbagai penyakit yang menyerang daun singkong, membantu petani untuk mengambil tindakan pencegahan atau pengendalian yang tepat. Dalam rangka mengatasi masalah ini, sebuah aplikasi deteksi penyakit pada daun tanaman singkong dikembangkan menggunakan algoritma *You Only Look Once* (YOLO) v8 berbasis Android. Metode pengembangan yang digunakan adalah metode *Waterfall.* Dengan tingkat akurasi mencapai 77%, Aplikasi ini telah melalui tiga jenis pengujian, yaitu *Black Box*, *User Acceptance Testing* (UAT), dan Pengujian Lapangan. Hasil pengujian *Black Box* menunjukkan tingkat keberhasilan 100%, UAT mencapai 86%, sementara hasil Pengujian Lapangan menunjukkan bahwa penyakit tanaman singkong yang mendominasi pada lokasi pengujian adalah *Green Mottle* dan *Bacterial Blight*. Pengujian aplikasi di lapangan berhasil mendeteksi penyakit dengan baik dan akurat secara *realtime*, ini memberikan harapan untuk pengendalian penyakit yang lebih efektif dalam pertanian singkong.

Kata Kunci: *You Only Look Once*, Deteksi *Realtime*, Penyakit Daun Singkong, Android, *Artificial Intelligence*.

PRAKATA

Dengan rasa syukur yang tak terhingga kepada Allah SWT atas berkah dan rahmat-Nya, penulis berhasil menyelesaikan skripsi ini dengan lancar. Skripsi berjudul "APLIKASI DETEKSI PENYAKIT PADA DAUN TANAMAN SINGKONG MENGGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO) V8 BERBASIS ANDROID" disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer.

Proses penyusunan skripsi 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 ijin 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 Utama Program Studi Informatika Fakultas Teknik dan Informatika Universitas PGRI Semarang yang telah membimbing penulis saat menyusun skripsi dengan penuh dedikasi yang tinggi.
5. Bapak Aris Tri Jaka Harjanta, S.Kom., M.Kom, selaku Dosen Pembimbing Pendamping Program Studi Informatika Fakultas Teknik dan Informatika Universitas PGRI Semarang yang telah membimbing penulis saat menyusun skripsi 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, dan adik penulis yang senantiasa mendoakan penulis demi terselesaikannya skripsi ini.
8. Teman-teman dekat penulis yang tidak bisa disebutkan satu persatu yang telah memberikan semangat dalam penyusunan skripsi ini.
9. Teman-teman Program Studi Informatika Universitas PGRI Semarang.

Akhir kata penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dan memberikan dukungan dalam penyusunan skripsi ini. Penulis berharap skripsi ini dapat memberikan manfaat bagi banyak orang khususnya pada bidang Informatika.

Semarang, 02 Juni 2024

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

A. Latar Belakang

Singkong merupakan salah satu tanaman yang banyak ditemukan di Indonesia dan merupakan tanaman dengan banyak manfaat. Di Indonesia, singkong merupakan produksi hasil pertanian pangan ke dua terbesar setelah padi dengan menyumbang lebih dari 18 juta produksi setiap tahunnya. Sebagaimana tanaman lainnya, tanaman singkong juga tidak luput dari serangan hama dan penyakit tanaman. Untuk mengidentifikasi penyakit tanaman singkong bisa dilihat dari gejala-gejala dan munculnya perubahan warna pada daun. Daun yang diserang oleh penyakit akan mempengaruhi hasil dari tanaman singkong karena daun merupakan bagian vital dari tanaman yang berfungsi sebagai tempat berlangsungnya proses fotosintesis[1]. Oleh karena itu, pengembangan teknologi deteksi penyakit pada daun tanaman singkong menjadi penting dalam upaya meningkatkan produktivitas pertanian. Salah satu teknologi yang dapat digunakan adalah YOLO (You Only Look Once), sebuah algoritma deteksi objek dalam bidang *deep learning*. YOLO merupakan salah satu teknik dalam *Convolutional Neural Network* (CNN) yang memungkinkan deteksi objek dengan tingkat akurasi yang tinggi.

YOLO merupakan salah satu metode yang paling cepat dan akurat pada pendeteksian objek bahkan mampu melebihi hingga 2 kali kemampuan algoritma lain[2]. Hal ini berbeda dengan pendekatan lain seperti CNN yang memerlukan beberapa proses untuk menghasilkan prediksi. Dengan penerapan YOLO dalam sebuah aplikasi berbasis Android, para petani atau ahli pertanian dapat dengan cepat dan mudah mendeteksi penyakit pada daun tanaman singkong hanya dengan menggunakan perangkat *smartphone* mereka.

B. Batasan Masalah

1. Penelitian ini membatasi jenis penyakit pada tanaman singkong yang hanya sebanyak 4 jenis penyakit yaitu: *Bacterial Blight*, *Brown Streak Disease*, *Green Mottle*, dan *Mosaic Disease*.
2. Aplikasi yang dikembangkan untuk penelitian ini hanya untuk platform Android.
3. Algoritma *You Only Look Once* (YOLO) versi 8 akan digunakan sebagai metode utama untuk deteksi penyakit pada daun tanaman singkong.
4. Penelitian ini menggunakan metode pengembangan sistem *Waterfall* dan hanya sampai dengan tahapan testing.

C. Perumusan Masalah

Bagaimana mengembangkan aplikasi berbasis Android yang mampu mendeteksi penyakit pada daun tanaman singkong dengan menggunakan algoritma YOLO v8?

D. Tujuan Penelitian

1. Mengimplementasikan metode YOLO dalam aplikasi untuk mendeteksi penyakit pada daun tanaman singkong. Algoritma YOLO dipilih karena kemampuannya untuk melakukan deteksi objek dengan tingkat akurasi yang tinggi. Mengurangi kerusakan tanaman serta penurunan hasil panen dengan memberikan solusi deteksi dini terhadap penyakit pada tanaman singkong.
2. Memfasilitasi petani atau ahli pertanian dalam mengidentifikasi penyakit tanaman singkong secara tepat dan akurat, sehingga memungkinkan pengambilan tindakan pencegahan atau penanganan yang tepat waktu.
3. Menyediakan solusi yang dapat diakses oleh semua petani, termasuk yang berada di daerah pedesaan atau terpencil, melalui pengembangan aplikasi berbasis Android yang mudah digunakan dan tidak memerlukan infrastruktur teknologi yang kompleks.

E. Manfaat Penelitian

1. Peningkatan Produktivitas Pertanian: Penelitian ini diharapkan dapat meningkatkan produktivitas pertanian, khususnya dalam budidaya tanaman singkong, dengan memberikan solusi deteksi penyakit yang cepat dan akurat. Hal ini dapat mengurangi kerugian akibat kerusakan tanaman serta penurunan hasil panen.
2. Aksesibilitas Teknologi: Pengembangan aplikasi berbasis Android untuk deteksi penyakit pada tanaman singkong akan memungkinkan akses yang lebih mudah bagi semua petani, termasuk yang berada di daerah pedesaan atau terpencil. Ini akan membantu mengatasi keterbatasan akses teknologi dan meningkatkan adopsi teknologi dalam pertanian.
3. Peningkatan Kesejahteraan Petani: Dengan kemampuan untuk mendeteksi penyakit tanaman secara dini, petani dapat mengambil tindakan pencegahan atau penanganan yang tepat waktu, mengurangi kerugian dan meningkatkan hasil panen. Hal ini akan berdampak positif pada kesejahteraan petani dan keberlanjutan usaha pertanian mereka.

BAB II
KAJIAN PUSTAKA/TEORI

A. Tinjauan Pustaka

Sebelumnya, beberapa penelitian telah dilakukan mengenai perancangan sistem menggunakan berbagai metode. Setiap penelitian memiliki kriteria dan pola yang berbeda satu sama lain. Berikut adalah tabel perbandingan penelitian sebelumnya yang mengadopsi beberapa metode dalam prosesnya:

Tabel 2. 1 Perbandingan Penelitian Sebelumnya

No	Nama Peneliti dan Tahun Penelitian	Judul Penelitian	Metode	Hasil Penelitian
1.	Mohammad Syarief, Amirul Mukminin, Novi Prastiti, Wahyudi Setiawan (2017)	Penerapan Metode *Naive Bayes Classifier* Untuk Deteksi Penyakit pada Tanaman Jagung	*Naive Bayes Classifier*	Ujicoba pertama dan kedua menunjukkan deteksi 18 dari 30 kasus dan 17 dari 30 kasus masing-masing, dengan penyakit paling umum adalah Bulai dan Hama Ulat Grayak. Kesimpulannya, Metode *Naive Bayes* kurang efektif, perlu perbaikan dengan *Certainty Factor*

				untuk bobot keyakinan pada gejala[3].
2	Lutfi Hakim, Sepyan Purnama Kristanto, Dianni Yusuf, Aditya Roman Asyari, Khoirul Umam. (2023)	Sistem Deteksi Penyakit dan *Crawling* Informasi Pada Tanamam Buah Naga Berbasis Web dan Android	*Extreme Programming*	Sistem deteksi penyakit pada batang buah naga mencapai akurasi yang diharapkan, namun masih terdapat kesalahan dalam proses *crawling* data dari Tokopedia dan http://cybex.pertanian.go.id yang memerlukan perbaikan, serta perlu pengumpulan informasi artikel terkait penyakit dan obatnya untuk pengembangan lebih lanjut[4].
3	Yohani Setiya Rafika Nur, Auliya Burhanuddin, Dasril Aldo, Widya Lelisa Army	Sistem Pakar Deteksi Penyakit Bawang Merah dengan Metode *Case*	*Case Based Reasoning*	Dari penelitian dan analisis, sistem pakar mampu mengidentifikasi penyakit bawang merah berdasarkan gejala yang dipilih

	(2022)	*Based Reasoning*		pengguna, menyimpulkan bahwa penyakit yang dialami adalah Bercak Ungu dengan *similarity* 100%[5].
4	Vita Indri Safitri (2017)	Sistem Pakar Deteksi Penyakit Tanaman Kentang Menggunakan Metode *Certainty Factor*	*Certainty Factor*	Kesimpulan dari perancangan sistem pakar menggunakan metode *Certainty Factor* adalah bahwa sistem ini berhasil menghasilkan aplikasi untuk mendiagnosa penyakit tanaman kentang dengan akurasi tinggi, ditunjukkan oleh hasil pengujian yang menunjukkan persentase *error* rata-rata sebesar 0.14% dan sebagian besar pakar menilai aplikasi ini baik[6].

B. Landasan Teori

1. Singkong

Tanaman singkong (*Manihot esculenta*) adalah tanaman umbi-umbian yang banyak dibudidayakan di berbagai belahan dunia, terutama di daerah tropis. Singkong dikenal karena umbinya yang kaya karbohidrat dan bisa diolah menjadi berbagai jenis makanan seperti tepung, kue, dan makanan pokok lainnya. Tanaman ini tumbuh subur di tanah yang subur dan memiliki toleransi terhadap kondisi lingkungan yang kurang menguntungkan. Singkong merupakan bahan makanan komoditi pangan yang melimpah serta banyak sekali dijumpai di daerah pedesaan dengan harga yang relatif terjangkau[7]. Tanaman singkong juga sama seperti tanaman lainnya yaitu mempunyai penyakit. Diantaranya adalah: *Bacterial Blight, Brown Streak Disease, Green Mottle, dan Mosaic Disease*[8].

Bacterial Blight pada daun singkong disebabkan oleh bakteri *Xanthomonas campestris pv. manihotis*. Penyakit ini biasanya mempengaruhi daun muda singkong, menyebabkan bercak-bercak kecoklatan pada permukaan daun yang kemudian berkembang menjadi lesi dengan tepi kuning. Infeksi yang parah dapat mengakibatkan daun mengering dan gugur, yang pada akhirnya dapat menyebabkan penurunan hasil panen yang signifikan jika tidak ditangani dengan tepat[9].

Gambar 2. 1 Penyakit Daun Singkong – *Bacterial Blight*

Brown Streak Disease, atau yang dikenal sebagai *Cassava Brown Streak Disease* (CBSD), adalah penyakit virus yang menyerang daun, batang, dan umbi singkong. Gejalanya meliputi bercak coklat pada daun, serta pucat dan keriting pada daun yang terinfeksi. Penyakit ini dapat menyebabkan kerugian ekonomi yang serius bagi petani singkong jika tidak dikendalikan dengan baik[10].

Gambar 2. 2 Penyakit Daun Singkong – *Brown Streak Disease*

Green Mottle pada daun singkong biasanya disebabkan oleh infeksi virus, seperti *Cassava Green Mottle Virus* (CGMV). Gejalanya termasuk daun yang menguning, keriting ke atas, dan pembengkakan pada batang. Infeksi yang parah dapat menyebabkan penurunan hasil panen dan kualitas umbi singkong yang buruk[11].

Gambar 2. 3 Penyakit Daun Singkong – *Green Mottle*

Mosaic Disease pada daun singkong juga disebabkan oleh virus dan ditandai dengan pola *mosaic* atau bercak-bercak pada daun, disertai dengan klorosis dan deformasi daun. Penyakit ini dapat menurunkan produktivitas tanaman dan kualitas umbi singkong jika tidak dikelola dengan baik[12].

Gambar 2. 4 Penyakit Daun Singkong – *Mosaic Disease*

2. *Machine Learning*

Machine learning merupakan cabang dari kecerdasan buatan *(Artificial Intelligence)* dan ilmu komputer yang berfokus pada penggunaan data dan algoritma untuk meniru cara manusia belajar secara bertahap dan dapat meningkatkan akurasinya. Semakin baik algoritma *machine learning* yang digunakan, maka akan semakin baik pula keputusan yang dihasilkan[13]. Dalam *machine learning*, terdapat beberapa tipe utama dari algoritma, masing-masing dengan karakteristik dan kegunaannya sendiri. Algoritma-algoritma ini dapat dibagi menjadi beberapa kategori utama, seperti:

a. Supervised Learning

Supervised Learning, atau yang juga dikenal sebagai pembelajaran terawasi, adalah jenis algoritma pembelajaran di mana komputer dilatih menggunakan data *input* yang telah diberi label untuk mencapai *output* tertentu. Dalam *supervised learning*, setiap objek dalam data memiliki kelas yang sudah diketahui.

Biasanya, *Supervised Learning* digunakan untuk tugas seperti klasifikasi, seperti mengidentifikasi jenis bunga iris, dan regresi, seperti memprediksi harga rumah[14].

b. Unsupervised Learning

Unsupervised Learning merupakan salah satu jenis algoritma *machine learning* yang dimanfaatkan untuk mengekstrak informasi dari kumpulan data. Model akan mengidentifikasi pola yang tersirat dalam data yang diberikan. *Unsupervised Learning* sering digunakan untuk tugas seperti pengelompokan (*clustering*), seperti membagi pelanggan dalam suatu bisnis ke dalam segmen yang berbeda[15].

c. Reinforcement Learning

Reinforcement Learning adalah jenis algoritma yang mempelajari perilaku terbaik dalam suatu lingkungan dengan tujuan memperoleh manfaat maksimal. Algoritma *Reinforcement Learning* didesain untuk beradaptasi terhadap perubahan yang terjadi dalam lingkungan. Fokus utama *Reinforcement Learning* adalah pada penelusuran tindakan yang menghasilkan imbalan kumulatif tertinggi dari lingkungan[16].

3. *Deep Learning*

Deep learning merupakan percabangan dari bidang *machine learning* yang menggunakan saraf tiruan atau disebut dengan *artificial neural networks* yang memiliki beberapa *layers* dalam memproses atau mempelajari suatu data. *Deep learning* memungkinkan suatu komputer untuk belajar secara otomatis melalui pengalaman yang diberikan agar sistem tersebut dapat mengidentifikasi pola yang kompleks dalam data. *Deep learning* memiliki model jaringan saraf tiruan yang mirip dengan struktur jaringan saraf manusia, dimana model tersebut terdapat beberapa lapisan neuron buatan yang terhubung satu sama lain[17].

4. *Artificial Intelligence*

Artificial Intelligence atau kecerdasan buatan merupakan pengembangan dan integrasi dari bidang elektronika, ilmu komputer, dan matematika. Secara sederhana, sistem dengan kecerdasan buatan dapat melakukan pekerjaan seperti yang dilakukan oleh manusia, seperti berpikir, mengambil keputusan, melakukan klasifikasi terhadap suatu keadaan, atau mengestimasi keadaan di masa yang akan datang[18]. Dengan kecerdasan buatan, berbagai aplikasi dapat dibuat untuk membantu manusia dalam menyelesaikan tugas-tugas yang kompleks dan berulang, meningkatkan efisiensi dan efektivitas dalam berbagai bidang kehidupan. AI dapat digunakan dalam berbagai sektor seperti kesehatan, transportasi, keuangan, pertanian, dan banyak lagi, memungkinkan pengambilan keputusan yang lebih baik, analisis data yang lebih cepat dan akurat, serta otomatisasi proses yang sebelumnya membutuhkan intervensi manusia. Selain itu, kecerdasan buatan juga memungkinkan pengembangan teknologi baru yang dapat mengubah cara kita hidup dan bekerja, dari mobil otonom hingga asisten virtual yang cerdas. Potensinya yang luas menjadikan kecerdasan buatan sebagai salah satu inovasi teknologi paling signifikan di era modern.

5. *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) adalah jenis arsitektur *neural network* yang paling umum digunakan dalam tugas-tugas pengolahan citra, termasuk pengenalan gambar. CNN memiliki keunggulan dalam mengekstraksi fitur-fitur penting dari gambar melalui proses konvolusi, di mana *filter* atau *kernel* digunakan untuk mendeteksi pola-pola lokal dalam citra. Proses ini membantu dalam mengidentifikasi tepi, tekstur, dan elemen-elemen penting lainnya yang membentuk dasar dari pengenalan gambar.

Struktur CNN terdiri dari beberapa lapisan konvolusi yang diikuti oleh lapisan *pooling*. Lapisan konvolusi bertanggung jawab untuk menerapkan filter yang berbeda pada gambar *input*, menghasilkan peta fitur yang mencerminkan keberadaan pola-pola spesifik. Lapisan *pooling*, di sisi lain, berfungsi untuk mengurangi dimensi dari peta fitur dengan cara melakukan *downsampling*, seperti *max pooling* atau *average pooling*. Hal ini tidak hanya mengurangi ukuran data, tetapi juga membantu dalam mengurangi *overfitting* dan meningkatkan efisiensi komputasi.

Di akhir arsitektur CNN, terdapat lapisan-lapisan *fully connected* yang menghubungkan seluruh neuron dari lapisan sebelumnya ke setiap neuron di lapisan berikutnya. Lapisan ini berperan dalam menggabungkan fitur-fitur yang telah diekstraksi oleh lapisan konvolusi dan *pooling*, serta melakukan klasifikasi akhir berdasarkan fitur-fitur tersebut. Dengan menggunakan fungsi aktivasi seperti *Softmax*, lapisan *fully connected* menghasilkan *output* berupa probabilitas dari setiap kelas yang mungkin, yang kemudian digunakan untuk menentukan kelas akhir dari gambar *input.*

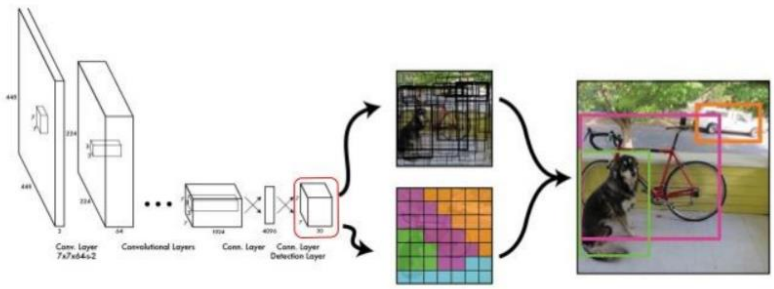
Keseluruhan proses ini menjadikan CNN sangat efektif dalam berbagai aplikasi pengolahan citra, termasuk deteksi objek, segmentasi citra, dan pengenalan wajah, menjadikannya pilihan utama dalam bidang *computer vision*[19].

6. *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. Dengan perkembangan versi-versi terbarunya, seperti YOLOv5 dan YOLOv8, algoritma ini terus dikembangkan untuk meningkatkan kecepatan dan akurasi deteksi objek pada berbagai aplikasi praktis[20].



Gambar 2. 5 Proses Algoritma YOLO dalam Deteksi Objek

7. *Object Detection*

Object detection adalah suatu teknik yang memungkinkan komputer untuk melihat dan mengidentifikasi objek yang ada baik dalam gambar maupun video. Hal ini juga dapat diartikan sebagai perancangan suatu sistem agar sistem mampu melihat dan mengidentifikasi objek seperti manusia. Dalam proses pembuatannya, diperlukan data pendukung yang nantinya akan dilatih agar sistem mampu mengenali suatu objek[21]. Data pendukung tersebut mencakup gambar-gambar yang sudah diberi label objek yang ingin dideteksi, serta koordinat lokasi objek tersebut dalam gambar. Proses pelatihan kemudian dilakukan dengan menggunakan algoritma *machine learning* seperti *Convolutional Neural Networks* (CNN), di mana model akan belajar dari data tersebut untuk mengenali pola-pola yang terkait dengan objek yang ingin dideteksi. Setelah proses pelatihan selesai, model dapat digunakan dalam tahap inferensi untuk melakukan deteksi objek pada data baru dengan memberikan prediksi lokasi dan kelas objek yang terdeteksi.

8. *Computer Vision*

Computer Vision adalah sistem otomatis yang digunakan untuk menganalisis citra dan video oleh komputer guna memperoleh informasi dan pemahaman tentang suatu objek. Ini merupakan kemampuan mesin atau komputer dalam melihat atau mengenali citra dengan tingkat kemampuan yang setara atau bahkan melebihi kemampuan penglihatan manusia asli[22]. *Computer Vision* memiliki beragam aplikasi, mulai dari pengenalan wajah dan identifikasi objek dalam gambar hingga navigasi kendaraan otonom dan pemantauan industri. Dengan teknologi ini, komputer dapat melakukan tugas-tugas yang sebelumnya memerlukan kehadiran manusia secara manual, membantu meningkatkan efisiensi dan akurasi dalam berbagai bidang.

9. Pengolahan Citra

Pengolahan citra adalah teknik untuk memanipulasi gambar digital agar sesuai dengan kebutuhan tertentu. Dalam aplikasi ini, pengolahan citra digunakan untuk mempersiapkan dan memproses gambar daun singkong sebelum dimasukkan ke dalam model YOLO. Pengolahan citra (*image processing*) mempunyai keterikatan yang erat dengan disiplin ilmu yang jika sebuah disiplin ilmu dinyatakan dalam bentuk proses suatu *input* menjadikan *output*, maka pengolahan citra memiliki *input* berupa citra serta *output* berupa citra. Kata "Citra" mengacu pada gambar digital dari objek di dunia nyata, yang dihasilkan oleh sensor seperti kamera, pemindai, atau perangkat medis seperti MRI dan *CT Scan*[23].

10. *Preprocessing* Data

Sebelum memberikan data gambar ke model YOLO, perlu dilakukan proses *preprocessing* seperti *resizing* gambar, normalisasi intensitas *pixel*, dan lain-lain untuk mempersiapkan data agar sesuai dengan kebutuhan model. *Preprocessing* adalah tahapan untuk menghilangkan beberapa permasalahan yang bisa mengganggu saat pemrosesan data. Hal tersebut karena banyak data yang formatnya tidak konsisten. *Preprocessing* data merupakan teknik paling awal sebelum melakukan data *mining*. Namun terdapat beberapa proses juga dalam data *preprocessing* seperti membersihkan, mengintegrasikan, mentransformasikan dan mereduksi data[24].

11. *Dataset*

Dataset adalah kumpulan data yang terorganisir yang digunakan untuk analisis atau penelitian dalam ilmu komputer, statistik, dan bidang lainnya. Pengembangan model YOLO memerlukan *dataset* yang mencakup gambar-gambar daun singkong yang terkena berbagai jenis penyakit seperti antraknosa, hawar daun, cendawan tepung, dan lain-lain. *Dataset* ini digunakan untuk melatih dan menguji model YOLO.

12. *Training Model*

Training model adalah proses untuk mengolah dan mengoptimalkan algoritma pembelajaran mesin menggunakan kumpulan data yang besar. Model YOLO dilatih menggunakan *dataset* yang telah disiapkan. Proses pelatihan melibatkan proses pembelajaran mesin di mana model secara iteratif memperbarui bobotnya untuk meminimalkan kesalahan prediksi antara keluaran yang dihasilkan dan label yang sebenarnya dari *dataset*.

13. Android

Android adalah sebuah sistem operasi perangkat *mobile* berbasis Linux yang mencangkup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka[25]. Sejak diluncurkan pada tahun 2008, Android telah menjadi sistem operasi *mobile* yang paling banyak digunakan di dunia, dengan jutaan aplikasi tersedia di Google Play Store.

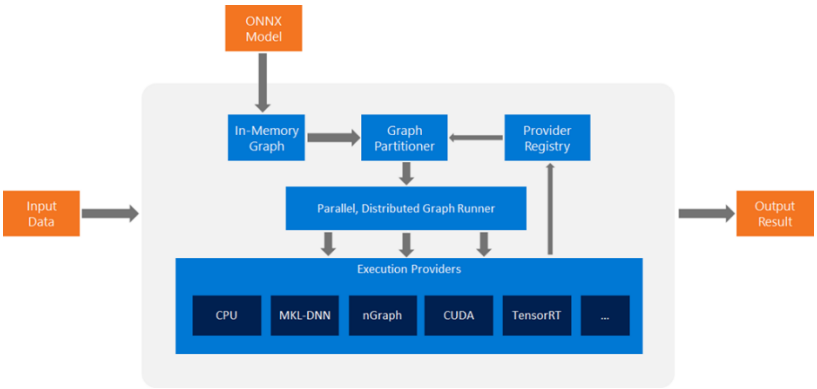
14. ONNX

Open Neural Network Exchange (ONNX) adalah format terbuka yang dibuat untuk mewakili model pembelajaran mesin (*machine learninng*). ONNX mendefinisikan serangkaian operator umum elemen penyusun pembelajaran mesin dan model pembelajaran mendalam serta format *file* umum yang memungkinkan pengembang AI menggunakan model dengan beragam kerangka kerja, alat, waktu proses, dan *kompiler*.

ONNX memiliki keuntungan utama dalam portabilitas model antara berbagai platform dan kerangka kerja *machine learning*. Dengan ONNX, pengembang dapat melatih model dengan satu kerangka kerja, seperti PyTorch atau TensorFlow, lalu mentransfernya ke kerangka kerja lain seperti Microsoft Cognitive Toolkit (CNTK) atau Apache MXNet untuk dideploy di lingkungan produksi. Hal ini memungkinkan

kolaborasi yang lebih besar antara tim yang menggunakan berbagai alat pembelajaran mesin dan mempercepat siklus pengembangan model.

ONNX juga menawarkan fleksibilitas dalam eksekusi model di berbagai perangkat keras, termasuk CPU, GPU, dan perangkat AI khusus lainnya. Dengan ekosistem yang terus berkembang dan dukungan dari berbagai vendor dan komunitas *open-source*, ONNX terus menjadi pilihan utama dalam penerapan model *machine learning*[26].



Gambar 2. 6 *ONNX Runtime Architecture*

15. Bahasa Pemrograman Python

Python merupakan salah satu dari bahasa pemrograman yang sering digunakan oleh programmer atau pembuat program dalam membuat program mereka. Python memiliki karakteristik *sintaks* yang tidak terlalu rumit. Sehingga Python menjadi salah satu bahasa pemrograman tingkat tinggi yang mudah untuk digunakan. Dalam menulis sebuah kode program menggunakan bahasa pemrograman Python, terdapat beberapa aturan yang harus dipenuhi. Hal ini untuk mengantisipasi terjadinya *error* atau masalah pada program yang dibuat. Aturan *sintaks* Python yang pertama adalah dalam penulisan *Statement* atau perintah[27].

16. Bahasa Pemrograman Kotlin

Kotlin adalah bahasa pemrograman berbasis Java Virtual Machine (JVM) yang dikembangkan oleh JetBrains. Bahasa pemrograman ini bersifat pragmatis untuk android yang mengkombinasikan *Object Oriented* (OO) dan pemrograman fungsional. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dengan bahasa pemrograman Java pada suatu *project* yang sama. Kotlin juga dapat digunakan untuk mengembangkan aplikasi berbasis desktop, web dan bahkan untuk *backend*[28].

17. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah sebuah bahasa standar yang digunakan untuk mendokumentasikan, mendesain, dan menggambarkan struktur dan perilaku sistem perangkat lunak. UML menyediakan seperangkat notasi grafis yang digunakan untuk merepresentasikan berbagai aspek dari sistem perangkat lunak, seperti konsep desain, struktur kelas, interaksi antar objek, dan alur kerja (*workflow*). UML menjadi alat yang populer di dunia rekayasa perangkat lunak karena dapat digunakan untuk berkomunikasi dengan jelas antara pengembang, pemangku kepentingan, serta memfasilitasi pemahaman dan dokumentasi yang konsisten dari sistem yang sedang dikembangkan. Setiap sistem yang kompleks seharusnya dapat dianalisis dari berbagai perspektif untuk mencapai pemahaman menyeluruh, dan untuk tujuan ini UML menyediakan berbagai jenis diagram yang dapat dikelompokkan berdasarkan sifatnya, baik statis maupun dinamis. Berikut adalah diantaranya:

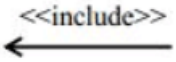
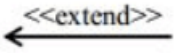
A. *Use Case Diagram*

Use Case Diagram adalah representasi model dari perilaku sistem informasi yang akan dikembangkan. Diagram ini berfungsi dengan menggambarkan interaksi tipikal antara pengguna sistem dengan sistem itu sendiri

melalui narasi atau cerita yang menjelaskan bagaimana sistem tersebut digunakan.

Adapun simbol-simbol yang digunakan dalam *Use Case Diagram* dapat dilihat pada Tabel 2.2:

Tabel 2. 2 Simbol *Use Case Diagram*

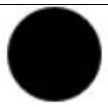
Simbol	Keterangan
	Aktor: Mewakil peran orang, *system* yang lain, atau alat ketika berkomunipasi dengan *use case*.
	*Use case*: Abstraksi dan interaksi antara sistem dan actor.
	*Association*: Abstraksi dari penghubung antara aktor dengan *use case*.
	Generalisasi: Menunjukkan spesialisasi *actor* untuk dapat berpartisipasi dengan *use case*.
	Menunjukkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya.
	Menunjukkan bahwa suatu *use case* seluruhnya merupakan tambahan fungsional dari *use case* lainnya jika suatu kondsi terpenuhi.

B. *Activity Diagram*

Activity Diagram adalah representasi visual yang digunakan dalam rekayasa perangkat lunak untuk menampilkan alur kerja atau aktivitas dari sebuah sistem. Dalam diagram ini, tindakan-tindakan konkret seperti

pengolahan data atau pengambilan keputusan direpresentasikan dengan menggunakan simbol-simbol tertentu yang dihubungkan dengan garis alir, menciptakan gambaran yang jelas tentang urutan langkah-langkah dalam proses tersebut. Adapun simbol-simbol yang digunakan dalam *Activity Diagram* dapat dilihat pada Tabel 2.3:

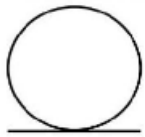
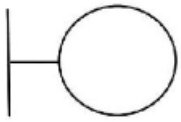
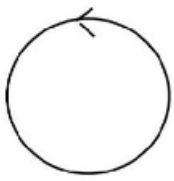
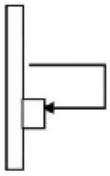
Tabel 2. 3 Simbol *Activity Diagram*

Simbol	Keterangan
	Status awal: Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas: Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / *Decision*: Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / *Join*; Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir: Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	*Swimlane*: *Swimlane* Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

C. *Sequence Diagram*

Sequence Diagram mengilustrasikan perilaku objek dalam skenario penggunaan dengan menggambarkan rentang waktu hidup objek serta pesan yang dikirimkan dan diterima di antara objek-objek tersebut. Gambaran diagram urutan dibuat minimal dengan jumlah definisi *use case* yang memiliki proses tersendiri atau yang paling penting, memastikan bahwa semua interaksi pesan yang dijelaskan dalam *use case* tercakup dalam diagram urutan. Sehingga, semakin banyak jumlah *use case* yang didefinisikan, semakin banyak juga diagram urutan yang harus dibuat untuk mencakup seluruh interaksi pesan yang terjadi. Adapun simbol-simbol yang digunakan dalam *Sequence Diagram* dapat dilihat pada Tabel 2.4:

Tabel 2. 4 Simbol *Sequence Diagram*

Simbol	Keterangan
	*Entity Class*: Gambaran sistem sebagai landasan dalam menyusun basis data.
	*Boundary Class*: Menangani komunikasi antar lingkungan sistem .
	*Control Class*: Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika.
	*Recursive*: Pesan untuk dirinya.

Simbol	Keterangan
	*Activation*: Mewakili proses durasi aktivasi sebuah operasi.
	Life Line: Komponen yang digambarkan garis putus terhubung dengan objek.

D. *Class Diagram*

Class diagram merupakan gambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* terdiri dari atribut dan operasi dengan tujuan pembuat pembuat program dapat membuat hubungan antara dokumentasi perancangan dan perangkat lunak sesuai[29]. Simbol-simbol yang digunakan dalam *Class Diagram* dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Simbol *Class Diagram*

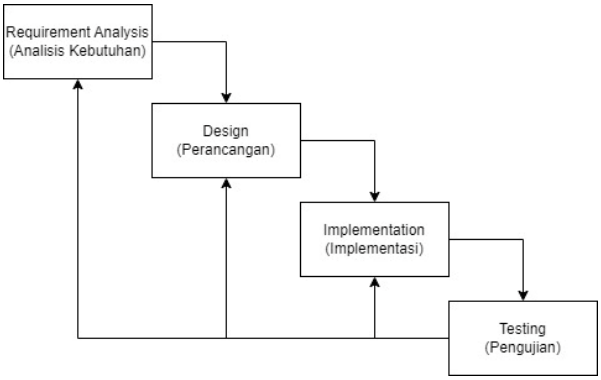
Simbol	Keterangan
	Generalization: Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Nary Association*: Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

	*Class*: Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	*Collaboration*: Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	Realization: Operasi yang benar-benar dilakukan oleh suatu objek.
	*Dependency*: Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	Association: Apa yang menghubungkan antara objek satu dengan objek lainnya.

18. Metode *Waterfall*

Metode *Waterfall* adalah metode yang melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding, testing* / *verification*, dan *maintenance*. Disebut dengan *Waterfall* karena tahap demi tahap yang dilalui pada metode ini harus menunggu selesainya tahap sebelumnya[30]. Metode *Waterfall* sering digunakan dalam proyek-proyek pengembangan perangkat lunak yang membutuhkan pendekatan yang terstruktur dan dengan setiap tahap yang dikerjakan secara berurutan tanpa mundur ke tahap sebelumnya[31]. Metode ini juga memberikan dokumentasi yang sangat baik karena setiap tahap harus didokumentasikan sebelum

melanjutkan ke tahap berikutnya. Dengan demikian, metode *Waterfall* sangat cocok digunakan untuk proyek-proyek yang memiliki persyaratan yang jelas dan tidak berubah-ubah selama proses pengembangan, serta untuk tim yang mungkin terdistribusi karena dokumen dapat digunakan sebagai acuan yang solid sepanjang proyek berlangsung.



Gambar 2. 7 Metode *Waterfall*

19. Pengujian *Black Box*

Pengujian *Black Box* testing disebut sebagai pengujian perilaku. Dimana struktur interior, logika perangkat lunak yang diuji tidak diketahui oleh penguji. Penguji didasarkan kepada spesifikasi kebutuhan dan tidak perlu dilakukannya analisis kode. Pengujian *Black Box* testing pengujian ini dilakukan dari sudut pandang pengguna akhir.

Ada beberapa jenis pengujian *Black Box* testing, diantaranya seperti: partisi, analisis nilai batas, grafik penyebab efek, pengujian *orthogonal array*, dan *fuzzing*. Selain itu *Black Box* testing memiliki keuntungan dan kekurangan dalam implementasinya. Salah satu kelebihannya yaitu membantu dalam hal penemuan aspek yang tidak terpenuhi dari spesifikasi kebutuhan yang diberikan dalam pengembangan perangkat lunak. Dan kekurangan dari *Black Box* testing adalah pengujian tidak bisa dilalukan sepenuhnya dikarenakan pengetahuan penguji terbatas tentang perangkat lunak yang diuji[32]. Pengujian *Black Box* testing melibatkan pengujian perangkat lunak

tanpa memperhatikan struktur internal atau logika program. Sebagai contoh, dalam pengujian partisi, *input* perangkat lunak dibagi menjadi kelompok-kelompok yang setara, dan setiap kelompok diuji secara terpisah untuk memastikan bahwa setiap kelompok *input* menghasilkan *output* yang diharapkan. Sedangkan dalam pengujian analisis nilai batas, penguji menguji perangkat lunak pada nilai-nilai batas dan di sekitar nilai-nilai tersebut untuk mengidentifikasi potensi masalah.

Keuntungan dari pendekatan *Black Box* testing meliputi kemampuannya untuk mengidentifikasi cacat fungsional yang mungkin terlewatkan selama pengembangan perangkat lunak dan kemudahan dalam penerapannya, karena tidak memerlukan pengetahuan detail tentang implementasi internal. Namun, kelemahan utamanya adalah keterbatasan dalam menemukan *bug* yang mungkin terkait dengan struktur internal perangkat lunak, karena penguji tidak memiliki akses ke logika program.

Penerapan *Black Box* testing sangat penting dalam memastikan kualitas perangkat lunak sebelum dirilis ke pengguna akhir. Dengan menggunakan berbagai teknik dan pendekatan *Black Box* testing, pengembang dapat memastikan bahwa perangkat lunak dapat berfungsi dengan baik dan memenuhi kebutuhan pengguna.

20. Pengujian UAT

Pengujian *User Acceptance Testing* (UAT) adalah tahap akhir dalam proses pengujian perangkat lunak sebelum diserahkan kepada pengguna, ini bertujuan memastikan aplikasi memenuhi kebutuhan bisnis dan persyaratan pengguna. Dilakukan oleh pengguna dengan skenario dunia nyata, UAT memverifikasi bahwa aplikasi siap digunakan dalam lingkungan produksi dan mendeteksi *bug* yang belum ditemukan sebelumnya. Proses UAT mencakup perencanaan, desain skenario uji, persiapan data uji, eksekusi, pelaporan hasil, perbaikan jika diperlukan, dan mendapatkan persetujuan pengguna akhir. UAT

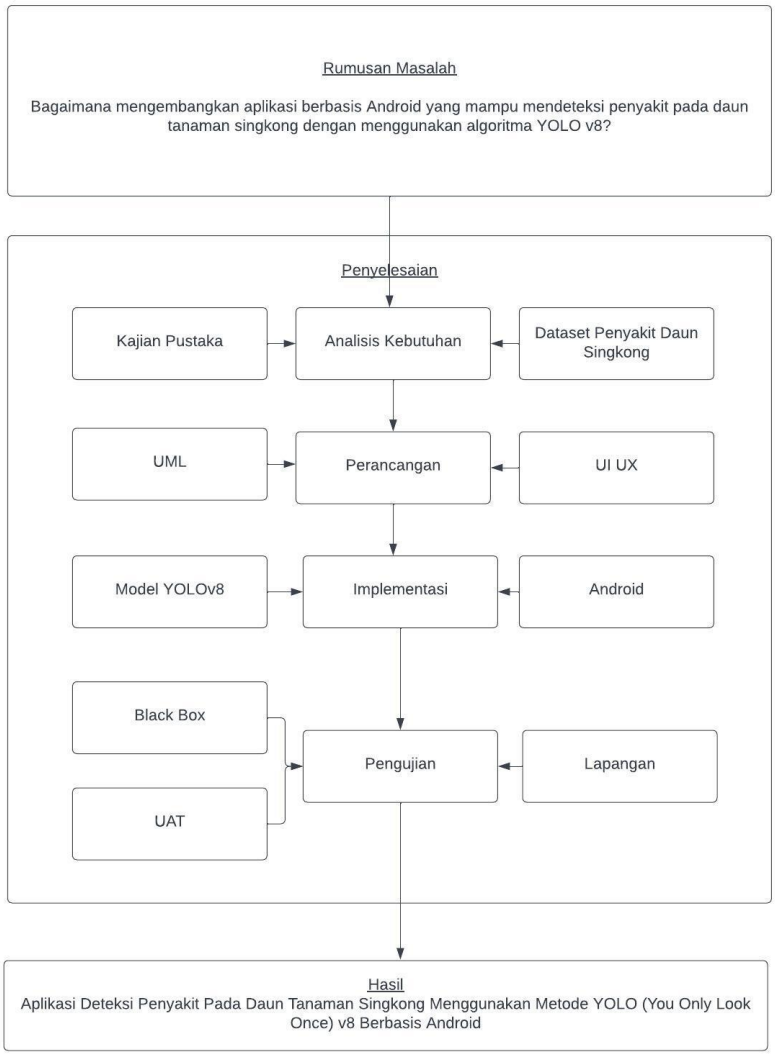
memastikan aplikasi berfungsi secara teknis dan memenuhi harapan pengguna akhir sebelum diluncurkan. Hasil dari UAT adalah dokumen yang menunjukkan bukti pengujian, berdasarkan bukti pengujian inilah dapat diambil kesimpulan, apakah sistem yang diuji telah dapat diterima atau tidak[33].

21. Pengujian Lapangan

Pengujian lapangan dalam pengembangan aplikasi adalah tahap di mana aplikasi diuji di lingkungan nyata oleh pengguna untuk memastikan bahwa ia berfungsi sesuai dengan harapan dalam kondisi sebenarnya. Proses ini melibatkan memantau bagaimana aplikasi tersebut digunakan dalam kehidupan sehari-hari. Pengguna memberikan umpan balik mengenai kinerja, kegunaan, dan stabilitas aplikasi, serta melaporkan masalah yang mungkin tidak terdeteksi selama pengujian internal atau di lingkungan pengembangan. Pengujian lapangan memungkinkan pengembang untuk memahami bagaimana aplikasi berinteraksi dengan berbagai perangkat keras, sistem operasi, dan jaringan yang berbeda, serta bagaimana aplikasi tersebut menangani situasi tak terduga atau beban kerja yang bervariasi. Melalui proses ini, pengembang dapat melakukan penyesuaian dan perbaikan yang diperlukan untuk meningkatkan kinerja dan pengalaman pengguna sebelum aplikasi diluncurkan secara luas. Pengujian lapangan memastikan bahwa aplikasi siap untuk dihadapi oleh berbagai tantangan dan situasi yang mungkin ditemui pengguna akhir, sehingga dapat memberikan performa yang optimal dan kepuasan yang tinggi.

C. Kerangka Berpikir

Sebuah kerangka berpikir telah dibuat untuk memudahkan proses penelitian dalam deteksi penyakit pada daun singkong menggunakan algoritma YOLO v8 berbasis Android. Permasalahan yang dihadapi adalah kesulitan dalam mendeteksi penyakit pada daun singkong secara tepat. Dengan tujuan tersebut, peneliti bertujuan untuk mengembangkan sebuah aplikasi yang dapat mendeteksi penyakit pada daun singkong menggunakan algoritma YOLO v8 pada platform Android. Kerangka berpikir untuk penelitian ini dapat dilihat dalam Gambar 2.8.



Gambar 2. 8 Kerangka Berpikir

BAB III
METODE PENELITIAN

A. Pendekatan Penelitian

Penelitian ini bertujuan untuk mengembangkan aplikasi deteksi penyakit pada daun tanaman singkong menggunakan algoritma YOLO v8 dengan pendekatan *Waterfall*. Tahapan penelitian dimulai dengan Analisis Kebutuhan, di mana akan dilakukan identifikasi kebutuhan pengguna dan analisis terhadap fitur-fitur yang dibutuhkan dalam aplikasi. Selanjutnya, tahap Desain akan dilakukan untuk merancang arsitektur sistem, antarmuka pengguna, dan algoritma. Setelah itu, Implementasi akan dilaksanakan dengan mengimplementasikan desain menjadi aplikasi Android yang fungsional. Pengujian akan menjadi tahap terakhir dalam pendekatan *Waterfall* ini, di mana aplikasi akan diuji secara menyeluruh untuk memastikan kinerja dan keandalannya dalam mendeteksi penyakit pada daun tanaman singkong. Dengan pendekatan ini, diharapkan dapat menghasilkan aplikasi yang efektif dan dapat diandalkan dalam mendukung pengendalian penyakit pada tanaman singkong.

B. Fokus Penelitian

Fokus Penelitian difokuskan pada pembahasan sebagai berikut :

1. Aplikasi berjalan dengan baik dan berbasis Android
2. Aplikasi mengidentifikasi penyakit pada daun tanaman Singkong berdasarkan foto yang diambil oleh pengguna.
3. Target pengguna aplikasi adalah petani tanaman Singkong.

C. Desain Penelitian

Desain penelitian untuk Aplikasi Deteksi Penyakit pada Daun Tanaman Singkong menggunakan algoritma YOLO v8 berbasis Android akan mengikuti pendekatan *Waterfall* dengan tahapan-tahapan berikut:

1. Analisis Kebutuhan: Analisis Kebutuhan akan dimulai dengan identifikasi kebutuhan pengguna melalui survei dengan petani tanaman

Singkong untuk memahami kebutuhan dan preferensi mereka terhadap aplikasi ini. Selanjutnya, akan dilakukan analisis fitur untuk menentukan fitur-fitur utama yang diperlukan dalam aplikasi, seperti kemampuan deteksi penyakit, antarmuka pengguna yang gampang dipahami, dan kemampuan untuk mengunggah foto daun tanaman Singkong.

2. Desain: Pada tahap Desain, akan dirancang arsitektur sistem secara keseluruhan, termasuk algoritma deteksi penyakit menggunakan metode YOLO v8 dan integrasi dengan platform Android. Selain itu, akan dibuat desain antarmuka pengguna yang mudah dipahami dan digunakan oleh petani tanaman Singkong, dengan fitur-fitur yang terorganisir dengan baik.
3. Implementasi: Setelah selesai merancang, tahap Implementasi akan dimulai dengan pengembangan aplikasi, yaitu menerjemahkan desain menjadi kode nyata, termasuk implementasi algoritma deteksi penyakit menggunakan YOLO v8, pengembangan antarmuka pengguna, dan integrasi dengan Android. Selama proses ini, akan dilakukan uji coba berkala untuk memastikan aplikasi berjalan dengan baik dan sesuai dengan spesifikasi yang telah ditetapkan.
4. Pengujian: Pengujian akan menjadi tahap terakhir dalam desain penelitian ini. Pengujian akan mencakup pengujian: *Black Box*, UAT, dan lapangan. Pada pengujian lapangan aplikasi akan diuji dengan menggunakan daun tanaman singkong secara langsung dan juga dengan foto daun yang diunggah oleh pengguna untuk memastikan keakuratannya dalam mendeteksi penyakit. Untuk pengujian *Black Box* dan UAT akan diujikan kepada pengguna secara langsung serta mengisi kuiseoner untuk selanjutnya dilakukan pendataan. Berikut adalah form pengujian yang nantinya akan digunakan dapat dilihat pada Tabel 3.1 dan Tabel 3.2.

Tabel 3. 1 Form Pengujian *Black Box*

Nama Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Diterima	Ditolak
Splash Screen	*User* menekan tombol mulai.	*User* dapat menekan tombol mulai dan dapat masuk ke halaman artikel.	Aplikasi akan menampilkan halaman artikel.		
Navigasi	*User* menekan beberapa tombol pada navigasi.	*User* dapat masuk ke halaman yang dituju dari navigasi aplikasi.	Aplikasi akan menampilkan halaman dituju.		
Halaman Artikel	*User* melakukan gulir atau *scrolling* kebawah.	*User* dapat melihat daftar artikel.	Aplikasi akan menampilkan daftar artikel.		
	User melakukan menekan pada *thumbnail* gambar atau judul artikel.	*User* dapat melihat isi artikel.	Aplikasi akan menampilkan keseluruhan isi artikel.		
Halaman Deteksi	*User* mengarahkan	*User* dapat melihat hasil	Aplikasi akan menampilkan		

	kamera ke objek yang akan dideteksi.	deteksi *realtime*.	hasil deteksi *realtime*.		
	User menekan tombol 'Galeri' pada halaman deteksi.	*User* dapat memilih gambar dari galeri untuk dideteksi.	Aplikasi akan menampilkan galeri dan akan menampilkan hasil deteksi ketika *user* telah memilih gambar.		
Halaman Tentang	*User* menekan tombol 'Tentang' pada navigasi aplikasi.	*User* dapat melihat isi halaman tentang aplikasi.	Aplikasi akan menampilkan halaman tentang aplikasi.		

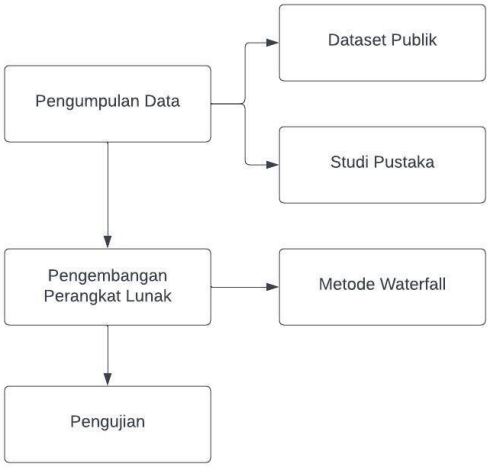
Tabel 3. 2 Form Pengujian UAT

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Aspek Kegunaan						
1.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong dapat bermanfaat bagi					

	pengguna, khususnya petani?					
2.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memberikan informasi tentang penyakit pada daun tanaman singkong dan cara mengatasinya?					
Aspek Kemudahan Pengguna						
3.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong mudah dioperasikan?					
4.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai yang diharapkan?					
5.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong berisi informasi yang dibutuhkan?					
6.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai keperluan Anda?					
Aspek *User Interface* (UI)						
7.	Apakah aplikasi deteksi penyakit pada daun					

	tanaman singkong memiliki tampilan yang mudah dipahami?					
8.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tampilan yang menarik?					
9.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tema warna yang enak dilihat?					
10.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong perlu dikembangkan lagi?					

Desain penelitian yang sudah dideskripsikan di atas kemudian dibentuk menjadi bentuk diagram dan dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Skema Desain Penelitian

D. Teknik Pengumpulan Data

1. *Dataset* Publik

   Pada teknik pengumpulan data ini yang pertama dilakukan peneliti adalah mengumpulkan data berupa gambar dari internet. Data yang digunakan dalam penelitian ini berupa *dataset* sebanyak 718 gambar yang diperoleh dari situs web Kaggle.com. *Dataset* ini dibagi beberapa *folder/class* yaitu: *Bacterial Blight, Brown Streak Disease, Green Mottle,* dan *Mosaic Disease*. Setelah penulis mengunduh *dataset*, lalu penulis melakukan anotasi secara manual pada *dataset* yang telah diunduh dan memindahkan *dataset* ke Google Drive untuk selanjutnya dilakukan pelatihan model. Penulis membagi 718 *dataset* gambar yang telah dianotasi dengan pembagian class: *Bacterial Blight*, *Brown Streak Disease*, *Green Mottle*, dan *Mosaic Disease* ke *training data* sebanyak 72% dan *validation* data sebanyak 28%.

2. Studi Pustaka

   Teknik ini melibatkan eksplorasi, pengumpulan, serta analisis dari literatur atau karya ilmiah yang relevan dengan topik penelitian yang sedang diteliti. Dalam proses ini, peneliti dapat menggali wawasan mendalam tentang subjek penelitian dan memperkuat landasan teoritis dengan merujuk pada penelitian sebelumnya. Langkah-langkah dalam teknik ini meliputi:

   a. Meneliti jurnal-jurnal yang berkaitan dengan penelitian memungkinkan peneliti untuk mendapatkan wawasan terbaru dan temuan dalam bidang identifikasi penyakit tanaman menggunakan teknologi seperti pengolahan citra, kecerdasan buatan, dan pemrosesan sinyal. Dengan mempelajari penelitian-penelitian sebelumnya, peneliti dapat memahami metode yang telah digunakan, kelemahan yang mungkin ada, dan berkontribusi pada pengembangan solusi yang lebih baik.

b. Memanfaatkan sumber informasi dari internet untuk mendapatkan data yang relevan memungkinkan peneliti untuk mengakses informasi terkini, data lapangan, dan sumber daya lainnya yang mungkin diperlukan dalam penelitian. Ini termasuk data gambar tanaman singkong yang sakit, informasi tentang gejala penyakit, dan teknik pengolahan citra yang dapat digunakan dalam pengidentifikasi penyakit.

E. Teknik Analisis Data

1. Analisis Kebutuhan Pengguna: Langkah pertama adalah memahami kebutuhan pengguna atau tujuan dari proyek ini. Ini melibatkan identifikasi masalah yang ingin dipecahkan, pemahaman tentang informasi yang dibutuhkan, dan kebutuhan spesifik lainnya yang akan mempengaruhi proses pengumpulan dan analisis data.
2. Pengumpulan *Dataset* dari Kaggle: *Dataset* dikumpulkan dari Kaggle.com, yang mungkin melibatkan pencarian, penelitian, dan pemilihan *dataset* yang sesuai dengan kebutuhan proyek. Ini juga dapat melibatkan verifikasi kualitas *dataset* dan pemastian bahwa *dataset* memenuhi syarat untuk analisis yang direncanakan.
3. *Preprocessing:* Pra-pemrosesan data adalah langkah penting untuk memastikan kualitas data sebelum digunakan dalam pembuatan model. Ini meliputi analisis terhadap *dataset* untuk memeriksa apakah ada masalah seperti kesalahan penempatan *folder/class,* gambar yang hilang, atau kesalahan penamaan *file*.

BAB IV
HASIL PENGEMBANGAN DAN PEMBAHASAN

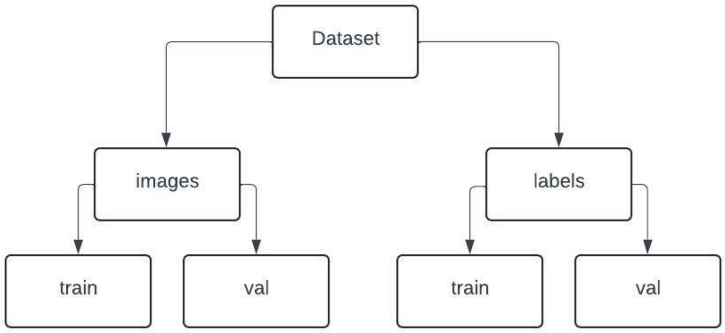
A. Hasil

Penulis menggunakan metode *Waterfall* dalam pengembangan aplikasi ini. Tahapan pengembangannya mengikuti urutan yang ada dalam metode ini adalah sebagai berikut:

1. Analisis Kebutuhan

A. Analisis Kebutuhan Data

Data yang diperlukan untuk penelitian ini adalah data gambar yang akan digunakan untuk melatih model deteksi. Data tersebut diunduh dari situs web Kaggle dan disimpan di dalam komputer oleh penulis. Proses pengolahan *dataset* melibatkan anotasi *dataset* secara manual setelah itu pembagian *dataset* ke dalam folder untuk selanjutnya digunakan dalam proses *training* model YOLO v8. Setelah anotasi *dataset* selesai, kemudian folder dibagi untuk kemudian dilakukan *training* model YOLO v8. Berikut gambaran struktur *dataset* di dalam folder:



Gambar 4. 1 Struktur Folder *Dataset*

Data *train* digunakan untuk melatih model, yang berarti model belajar dari pola-pola yang terdapat dalam data tersebut untuk membuat prediksi yang akurat. Proses ini mirip dengan pembelajaran manusia di mana individu belajar dari pengalaman. Namun, penting untuk memiliki data yang berbeda untuk memvalidasi model yang telah dilatih, sehingga dapat mengukur seberapa baik model tersebut dapat memprediksi data yang belum pernah dilihat sebelumnya.

Data *val* digunakan untuk menguji kinerja model secara objektif. Dengan menggunakan data yang terpisah dari data pelatihan, model dapat mengevaluasi kemampuan untuk menggeneralisasi pola-pola yang ada dalam data yang belum pernah dilihat sebelumnya. Dengan demikian, dapat dipastikan bahwa model tidak hanya menghafal data pelatihan, tetapi juga dapat bekerja dengan baik pada data baru. Dengan pendekatan ini, pengembang dapat memastikan bahwa model yang dihasilkan adalah model yang andal dan dapat diandalkan untuk digunakan dalam berbagai aplikasi.

Anotasi dalam metode YOLO v8 merujuk pada proses menandai atau memberi label kepada objek dalam gambar dengan menggunakan kotak pembatas (*bounding box*) dan klasifikasi kelas objek. Anotasi ini penting dalam proses pelatihan model deteksi objek untuk memungkinkan model belajar mengenali dan membedakan berbagai objek dalam gambar. Proses anotasi biasanya dilakukan secara manual oleh manusia, di mana setiap objek dalam gambar diberi label dan dikelilingi dengan kotak pembatas yang sesuai. Setelah proses anotasi selesai, data tersebut dapat digunakan untuk melatih model deteksi objek agar dapat mengenali objek-objek tersebut dengan akurasi yang lebih baik. Hasil anotasi pada pengembangan kali ini disimpan dengan nama *labels*.

Dalam *dataset* pada penelitian kali ini, penulis membagi *dataset* menjadi 2 folder yaitu folder *labels* dan *images*. Di dalam 2 folder tersebut terdapat pembagian kembali yaitu folder *train* dan *val*. Presentasi pembagian seluruh *dataset* dari 100%, 72% untuk folder *train* dan 28% untuk folder *val*. Berikut adalah tabel data dari *dataset* yang sudah siap untuk *training model*:

Tabel 4. 1 Data *Dataset Images*

No	Nama Folder	Jumlah Data
1.	*train*	518 gambar
2.	*val*	200 gambar

Tabel 4. 2 Data *Dataset Labels*

No	Nama Folder	Jumlah Data
1.	*train*	518 label
2.	*val*	200 label

B. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah langkah dalam memahami, mendokumentasikan, dan menetapkan persyaratan yang harus dipenuhi oleh suatu sistem perangkat lunak. Tujuannya adalah untuk memahami masalah yang akan dihadapi atau kebutuhan yang akan diimplementasikan dalam sistem tersebut. Berikut adalah beberapa data analisis kebutuhan dalam bentuk tabel yang akan digunakan oleh peneliti untuk mengembangkan aplikasi deteksi penyakit daun tanaman singkong dari mulai *Computer* untuk pengembangan dan *Mobile Smartphone* untuk pengujian dan minimum spesifikasi pemakaian aplikasi:

1) Kebutuhan Perangkat Keras *Mobile Smartphone*

Tabel 4. 3 Kebutuhan Perangkat *Mobile Smartphone*

RAM	Sistem Operasi	Penyimpanan	Resolusi
4GB (Minimal)	Android 7 Nougat (Minimal)	3GB (Minimal Tersedia)	1280x720 (Minimal)

2) Kebutuhan Perangkat Keras *Computer*

Tabel 4. 4 Kebutuhan Perangkat Keras Computer

RAM	GPU	CPU	Penyimpanan	Resolusi
8GB (Minimal)	NVIDIA GPU dengan dukungan CUDA (8GB Minimal)	Intel Core i5 / AMD Ryzen 3 atau yang lebih tinggi	20GB (Minimal Tersedia)	1280x720 (Minimal)

3) Kebutuhan Perangkat Lunak *Computer*

Tabel 4. 5 Kebutuhan Perangkat Lunak *Computer*

Sistem Operasi	Aplikasi
- Windows 10 / 11 (Atau diatasnya) - Linux	- Google Colaboratory - Android Studio - Google Drive - Visual Studio Code - Chrome Browser - Figma - Draw.io - Kaggle - Labelimg

C. Analisis Kebutuhan Fungsional

1) Halaman awal pada aplikasi yaitu *splash screen* yang berfungsi untuk memberikan informasi bahwa aplikasi siap digunakan.
2) Aplikasi dapat menampilkan halaman artikel untuk memberikan pemahaman kepada pengguna tentang penyakit pada daun tanaman singkong.
3) Aplikasi dapat menampilkan menu untuk deteksi penyakit daun tanaman singkong secara *realtime*.
4) Aplikasi dapat menampilkan menu untuk deteksi penyakit daun tanaman singkong secara tidak langsung menggunakan gambar yang sudah dimiliki oleh pengguna.
5) Aplikasi dapat menampilkan halaman tentang aplikasi untuk menambah pengetahuan kepada pengguna.

2. Desain

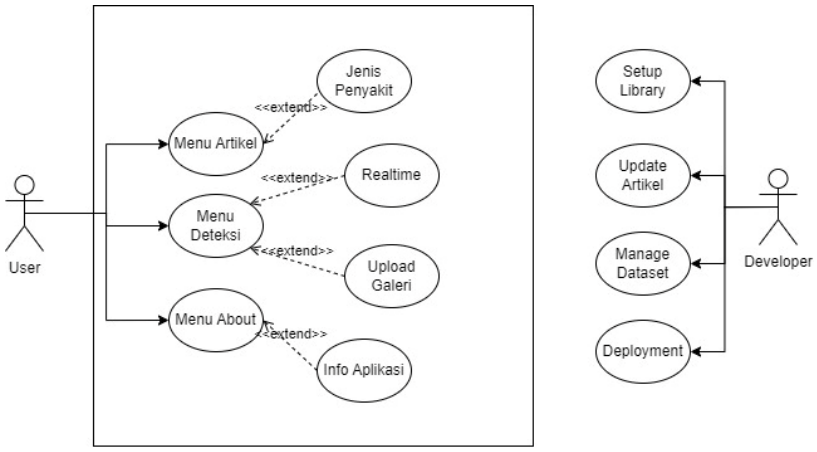
Tujuan dari desain sistem ini adalah untuk memudahkan proses implementasi atau penulisan kode dengan menyediakan panduan yang jelas sebelum pembuatan dimulai. Desain yang disusun oleh penulis untuk pengembangan aplikasi deteksi penyakit pada daun tanaman singkong adalah sebagai berikut:

a) Desain Unified Modeling Language (UML)

Dalam desain menggunakan *Unified Modeling Language* (UML), berbagai jenis diagram digunakan untuk mengilustrasikan berbagai aspek dari aplikasi atau sistem yang sedang dikembangkan. Berikut beberapa jenis diagram UML yang digunakan oleh penulis:

1) *Use Case Diagram*

Use Case Diagram digunakan untuk mengilustrasikan hubungan antara aktor-aktor dengan sistem yang sedang dikembangkan. Dengan memanfaatkan diagram ini, penulis bisa mendapatkan gambaran yang terperinci tentang fitur-fitur yang dibutuhkan oleh sistem dan bagaimana interaksi antara aktor dengan sistem tersebut akan terjadi. Diagram ini dapat dilihat pada Gambar 4.2.



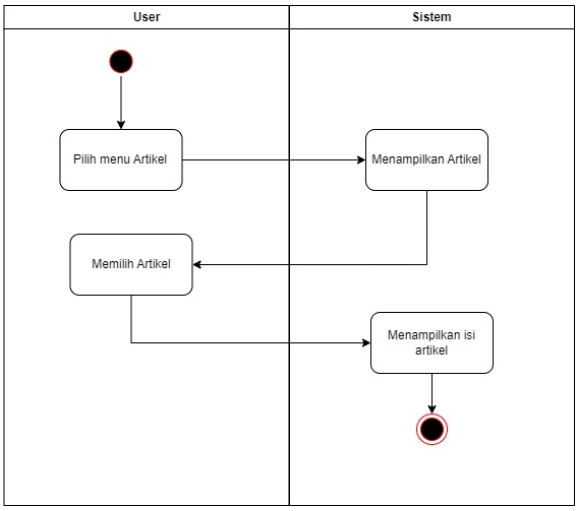
Gambar 4. 2 *Use Case Diagram*

Berdasarkan Diagram 4.2, dapat diuraikan bahwa dalam sistem, terdapat dua peran utama, yaitu *User* dan *Developer*. Pengguna memiliki akses penuh terhadap aplikasi, dimulai dari proses membuka aplikasi hingga menutupnya. Mereka juga dapat membaca artikel tentang penyakit pada daun tanaman singkong dan melakukan deteksi secara langsung (*realtime*) atau dengan opsi lain yaitu memasukkan gambar dari galeri foto. Di sisi lain, pengembang memiliki hak akses untuk memperbarui data artikel dan mengelola *model dataset*.

2) *Activity Diagram*

Diagram ini adalah representasi visual dari alur kerja atau proses yang menunjukkan serangkaian aktivitas sebagai simpul dan hubungan antara aktivitas tersebut dalam bentuk panah. Biasanya digunakan dalam pengembangan perangkat lunak untuk memodelkan langkah-langkah yang diperlukan untuk menyelesaikan tugas tertentu atau rangkaian aktivitas dalam sistem. Ini membantu pengembang dan pemangku kepentingan untuk memahami urutan aktivitas, pengambilan keputusan, dan aliran kontrol dalam proses yang didefinisikan.

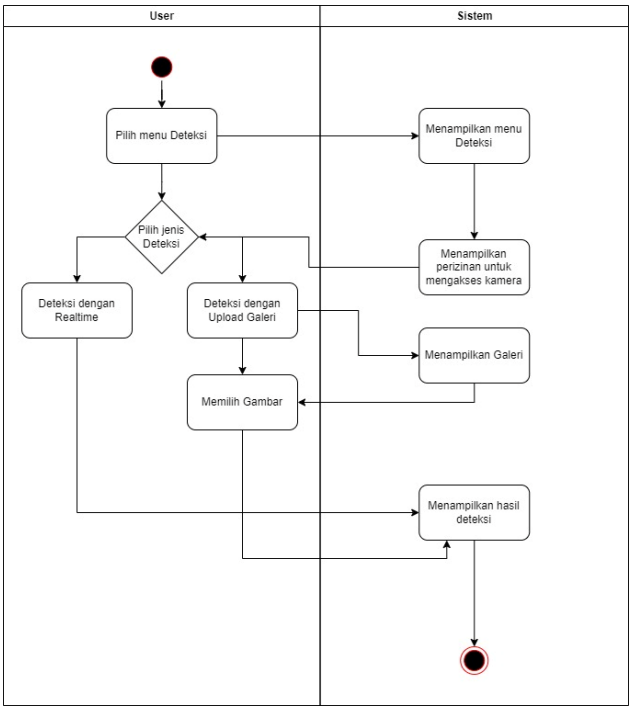
a) *Activity Diagram* Menu Artikel



Gambar 4. 3 *Activity Diagram* Menu Artikel

Berdasarkan gambar 4.3 ketika pengguna membuka aplikasi akan muncul halaman utama terdapat sejumlah artikel mengenai penyakit daun tanaman singkong. Pengguna dapat memilih artikel yang ingin dibaca, dan kemudian sistem akan menampilkan konten artikel yang dipilih oleh pengguna.

b) *Activity Diagram* Menu Deteksi

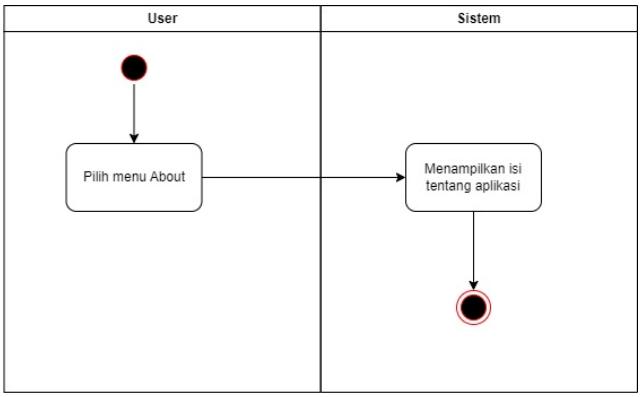


Gambar 4. 4 *Activity Diagram* Menu Deteksi

Diagram pada Gambar 4.4 digunakan untuk mendeteksi jenis penyakit pada daun tanaman singkong, yang merupakan salah satu menu penting dalam aplikasi yang sedang dikembangkan oleh penulis. Pada menu ini, pengguna diminta untuk memilih menu deteksi yaitu secara langsung (*realtime*) atau mengunggah dari galeri. Jika pengguna membuka menu deteksi setelah memasang aplikasi, sistem akan meminta izin pengguna terlebih dahulu untuk menggunakan kamera. Setelah itu pengguna bisa

melakukan deteksi secara *realtime* ataupun menggunakan metode *upload* dengan galeri. Setelah menu deteksi diklik pengguna langsung bisa menggunakan fungsi *realtime* dan akan langsung memunculkan hasil deteksi. Jika ingin menggunakan deteksi dari galeri, pengguna harus klik terlebih dahulu tombol galeri, setelah gambar dipilih sistem akan langsung mendeteksi dan menampilkan hasil deteksi.

c) *Activiy Diagram* Menu *About*



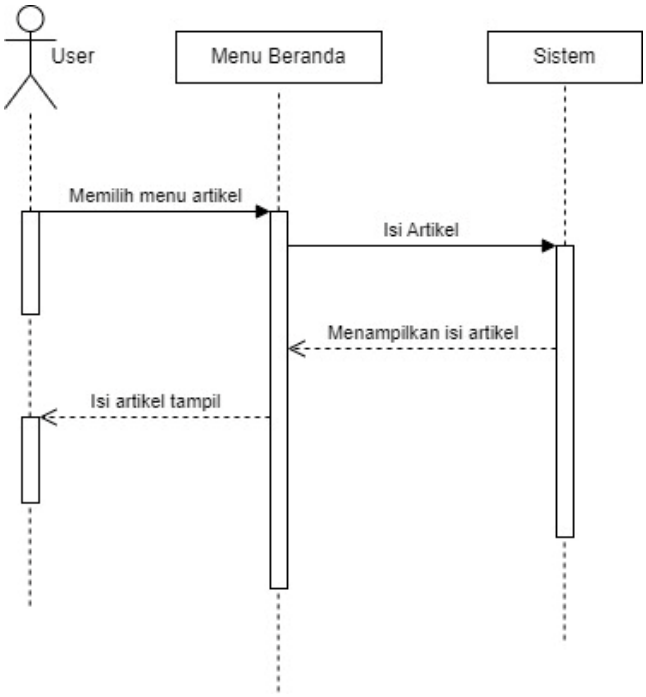
Gambar 4. 5 *Activity Diagram* Menu *About*

Berdasarkan gambar 4.5 ketika pengguna membuka aplikasi dan memilih menu *About* maka sistem akan menampilkan isi menu *About*, dimana di menu ini terdapat keterangan tentang aplikasi deteksi penyakit daun tanaman singkong sedang dikembangkan oleh penulis.

3) *Sequence Diagram*

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

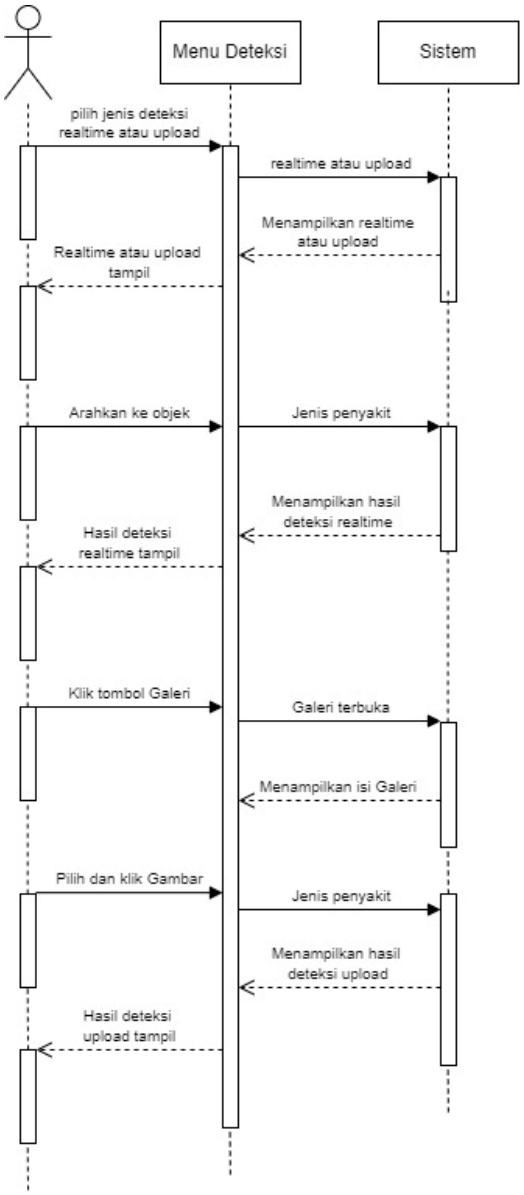
a) *Sequence Diagram* Menu Artikel



Gambar 4. 6 *Sequence Diagram* Menu Artikel

Menurut diagram pada Gambar 4.6, langkah awalnya adalah *user* memilih jenis artikel yang ingin dibaca, kemudian sistem akan menampilkan konten artikel yang dipilih oleh *user* tersebut.

b) *Sequence Diagram* Menu Deteksi

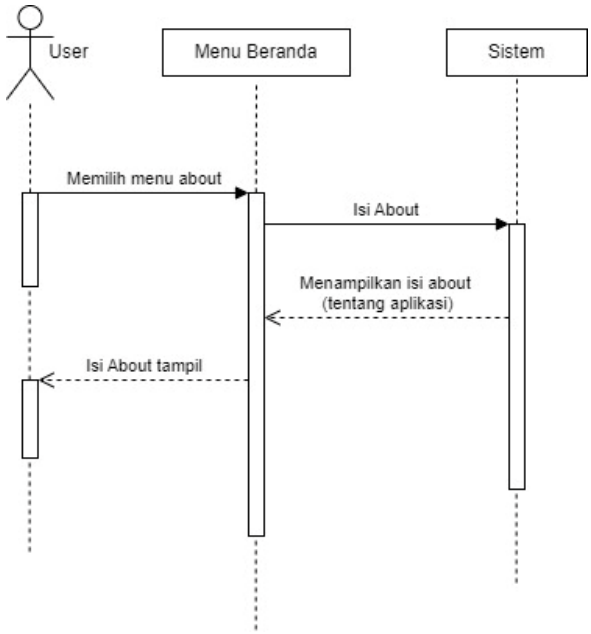


Gambar 4. 7 *Sequence Diagram* Menu Deteksi

Berdasarkan diagram pada Gambar 4.7, langkah awalnya adalah pengguna memilih jenis deteksi baik dari *realtime* atau *upload*, setelah itu sistem akan memperhatikan pilihan pengguna dan menampilkan metode yang telah dipilih. Selanjutnya, pengguna akan melakukan deteksi *realtime* atau *upload*. Untuk *realtime* pengguna hanya perlu

mengarahkan kamera ke objek, sedangkan *upload* pengguna perlu memilih gambar dari galeri terlebih dahulu. Setelah itu proses deteksi akan berlangsung. Dalam *realtime,* hasil deteksi akan langsung terlihat sedangkan metode *upload* akan terdeteksi setelah gambar telah dipilih.

c) *Sequence Diagram* Menu *About*



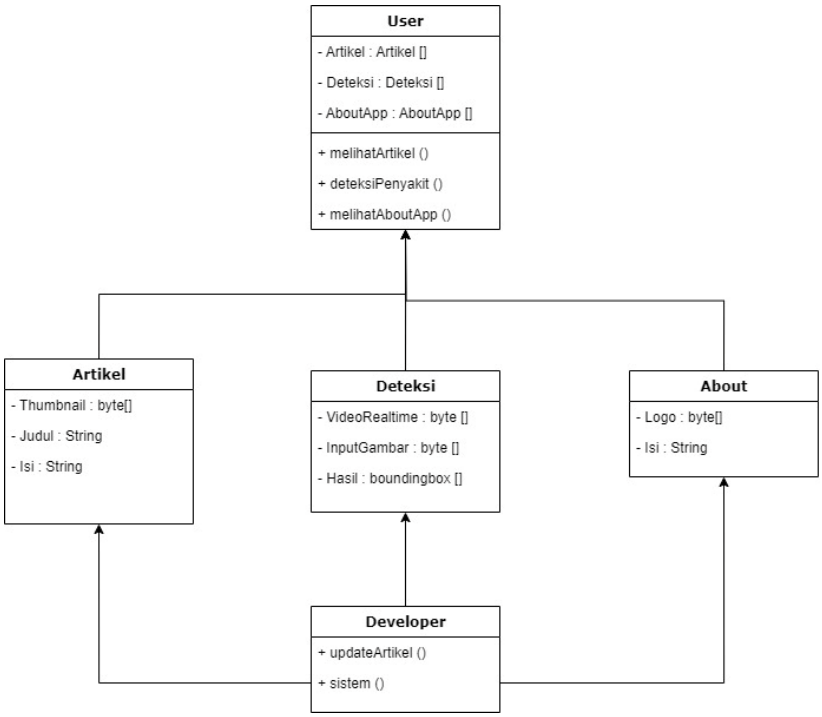
Gambar 4. 8 *Sequence Diagram* Menu *About*

Berdasarkan Gambar 4.8, proses pertama pengguna memilih menu *About*, berikutnya sistem akan menampilkan isi menu *About*.

4) *Class Diagram*

Class Diagram adalah jenis diagram dalam *Unified Modeling Language* (UML) yang berfungsi untuk mengilustrasikan struktur statis suatu sistem atau aplikasi. Diagram ini memvisualisasikan kelas-kelas yang ada dalam sistem, hubungan antara kelas-kelas tersebut, serta atribut dan metode yang dimiliki oleh setiap kelas. Beberapa elemen kunci dalam *Class Diagram* meliputi kelas, atribut, metode, dan

hubungan antar kelas. Tujuan dari *Class Diagram* adalah menyediakan gambaran yang jelas tentang struktur kelas dalam sistem, sehingga mempermudah pemahaman tentang bagaimana sistem beroperasi dan bagaimana komponen-komponen dalam sistem berinteraksi satu sama lain. Berikut *Class Diagram* yang menggambarkan deteksi penyakit pada daun tanaman singkong menggunakan metode YOLO v8 Berbasis Android:



Gambar 4. 9 *Class Diagram*

b) Desain User Interface

1) Desain Halaman *Splash Screen*

Gambar 4. 10 Desain *User Interface* Halaman *Splash Screen*

Berdasarkan Gambar 4.10, *splash screen* adalah halaman pertama yang muncul saat membuka aplikasi. Pada *splash screen* ini, terdapat beberapa komponen utama yang mencakup logo aplikasi, *tagline* aplikasi, dan tombol mulai untuk memulai penggunaan aplikasi. Logo aplikasi biasanya ditempatkan di bagian atas halaman untuk memberikan identitas visual kepada pengguna. Di bawah logo, terdapat *tagline* aplikasi yang singkat namun informatif, menjelaskan apa yang dapat dilakukan oleh aplikasi tersebut. Sedangkan di bagian bawah halaman, terdapat tombol mulai yang memungkinkan pengguna untuk memulai penggunaan aplikasi dengan sekali klik. Dengan demikian, *splash screen* menjadi pengantar yang sempurna untuk menyambut pengguna saat mereka memulai pengalaman mereka dengan aplikasi.

2) Desain Halaman Artikel

Gambar 4. 11 Desan *User Interface* Halaman Artikel

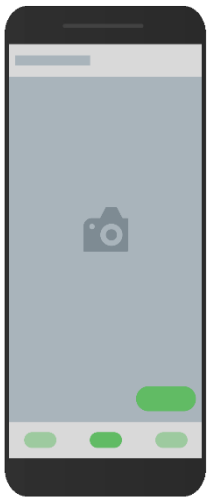
Berdasarkan Gambar 4.11, Halaman tersebut menampilkan beberapa komponen yang memudahkan pengguna dalam membaca artikel yang diinginkan. Di bagian atas halaman, terdapat judul menu halaman. Di bawahnya, terdapat *thumbnail* artikel yang memberikan gambaran visual tentang konten artikel. Di bawah *thumbnail*, terdapat judul dan *preview* isi artikel yang memberikan cuplikan tentang apa yang akan dibaca pengguna. Untuk mempermudah navigasi, terdapat menu navigasi yang memungkinkan pengguna untuk berpindah antara menu-menu yang tersedia. Warna hijau gelap pada desain menandakan menu navigasi yang sedang aktif, memberikan pengalaman pengguna yang terarah saat berinteraksi dengan aplikasi tersebut. Dengan demikian, pengguna dapat dengan mudah menavigasi dan memilih artikel yang sesuai dengan minat mereka.

3) Desain Halaman Isi Artikel

Gambar 4. 12 Desain *User Interface* Halaman Isi Artikel

Berdasarkan Gambar 4.12, Halaman tersebut dirancang untuk memfasilitasi pengguna dalam membaca artikel yang diminati. Di bagian atas halaman terdapat judul artikel yang menarik perhatian pengguna. Selain itu, terdapat tombol kembali ke halaman utama yang memungkinkan pengguna untuk dengan mudah kembali ke menu utama aplikasi. *Thumbnail* artikel juga disediakan sebagai representasi visual dari konten artikel yang akan dibaca. Di bawah *thumbnail*, pengguna dapat menemukan isi artikel yang lengkap, memungkinkan mereka untuk membaca artikel dengan nyaman tanpa harus meninggalkan halaman tersebut. Dengan adanya komponen-komponen tersebut, pengguna dapat dengan mudah menavigasi dan menikmati konten artikel yang tersedia dalam aplikasi.

4) Desain Halaman Deteksi



Gambar 4. 13 Desain *User Interface* Halaman Deteksi

Berdasarkan Gambar 4.13, Halaman tersebut menyajikan beberapa komponen yang memungkinkan pengguna untuk menggunakan fitur deteksi secara langsung. Di bagian atas halaman, terdapat judul menu. Fitur utama dari halaman ini adalah kamera *realtime* yang memungkinkan pengguna untuk melakukan deteksi langsung melalui kamera perangkat mereka. Pengguna dapat melihat hasil deteksi secara langsung pada layar kamera. Selain itu, terdapat juga tombol *upload* dari galeri yang memungkinkan pengguna untuk mengunggah gambar atau konten dari galeri perangkat mereka untuk kemudian dilakukan deteksi. Ini memberikan fleksibilitas kepada pengguna dalam memilih sumber gambar untuk analisis. Terakhir, terdapat navigasi yang aktif yang memungkinkan pengguna untuk berpindah antara menu atau fitur yang tersedia dalam aplikasi tersebut. Dengan adanya komponen-komponen tersebut, pengguna dapat dengan mudah menggunakan fitur deteksi yang disediakan dengan baik.

5) Desain Halaman *About*

Gambar 4. 14 Desain *User Interface* Halaman *About*

Berdasarkan Gambar 4.14, Halaman tersebut menampilkan beberapa komponen yang penting untuk memberikan informasi kepada pengguna tentang aplikasi tersebut. Di bagian atas halaman, terdapat judul menu. Di bawahnya, terdapat logo aplikasi yang memberikan identitas visual kepada pengguna, disertai dengan nama aplikasi yang jelas. Selanjutnya, terdapat bagian "Tentang Aplikasi" yang berisi teks yang menjelaskan informasi tentang aplikasi tersebut, seperti deskripsi singkat, fitur utama, atau tujuan dari aplikasi tersebut. Hal ini membantu pengguna untuk memahami lebih lanjut tentang apa yang dapat mereka harapkan dari penggunaan aplikasi tersebut. Terakhir, terdapat navigasi yang aktif yang memungkinkan pengguna untuk berpindah antara menu atau fitur yang tersedia dalam aplikasi. Dengan adanya komponen-komponen tersebut, pengguna dapat dengan mudah memahami dan mengakses informasi yang disediakan oleh aplikasi.

3. Implementasi Model

a) Data *Acquisition*

Pada penelitian ini, pengumpulan data dilakukan melalui *scrapping* data dari Google dengan mencari informasi terkait dan mengunduh *dataset* dari situs web Kaggle dengan *keyword* "*cassava leaf disease*". Data yang dikumpulkan terdiri dari empat kelas yang berbeda, yakni: *Bacterial Blight, Brown Streak Disease, Green Mottle,* dan *Mosaic Disease*. Gambar 4. 15 menampilkan contoh dari setiap kelas di dalam *dataset*.

Gambar 4. 15 *Dataset Class*

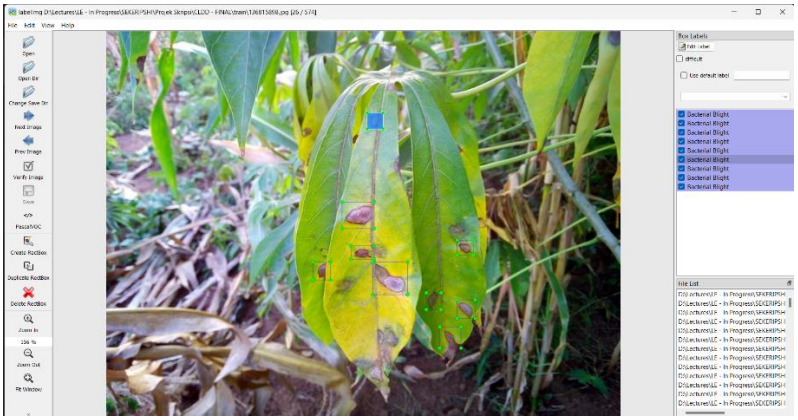
b) *Pre-processing* Data

Tahap *preprocessing* bertujuan untuk mempermudah proses pengolahan citra pada tahapan selanjutnya. Pada tahap ini, dilakukan penyaringan data yang tepat untuk disertakan dalam *dataset* dan dilakukan penyesuaian ukuran setiap citra menjadi 640 x 640 *pixel*.

c) *Labelling* Data

Proses *labeling* data dilakukan dengan memberikan anotasi pada citra tersebut menggunakan *bounding box*, serta menetapkan kelas pada setiap citra. Proses anotasi dilakukan menggunakan *software* Labelimg, di mana setiap citra diberi tanda *bounding box* untuk menandai area yang relevan, seperti daun yang terinfeksi atau bagian yang ingin dianalisis. Proses ini memungkinkan *dataset* untuk dilengkapi dengan informasi yang diperlukan untuk pelatihan

dan evaluasi model dalam pengolahan citra lebih lanjut. Proses menambahkan label pada citra memerlukan waktu tergantung banyaknya *dataset* yang ada, pada penelitian ini penulis membutuhkan waktu kurang lebih 3 hari untuk melakukan anotasi dengan total 718 data citra. Untuk dokumentasi proses *labeling* menggunakan LabelImg bisa dilihat pada Gambar 4.16.



Gambar 4. 16 Proses *Anotasi Labeling Dataset*

d) *Split* Data

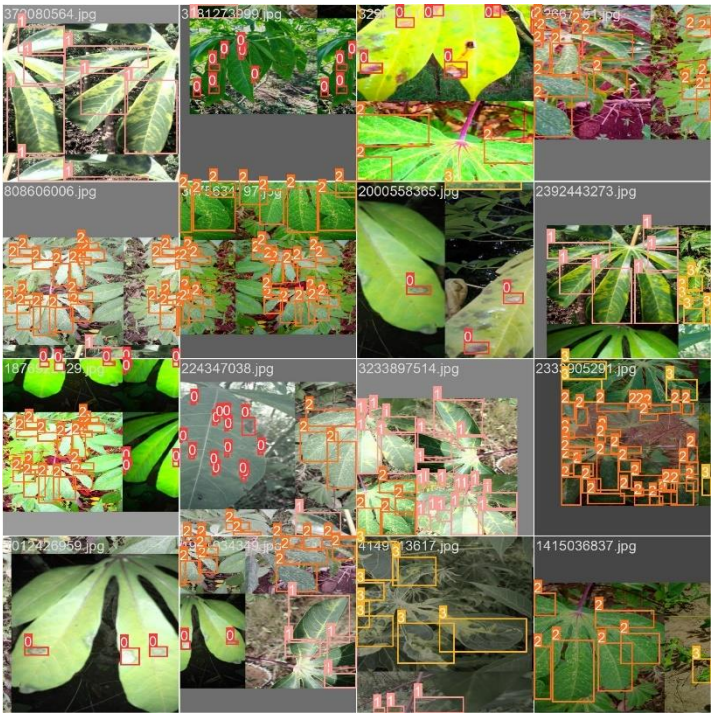
Dalam tahap pemisahan data, *dataset* dibagi menjadi dua bagian, yaitu data pelatihan (*training*) dan data validasi (*validation*). Data *training* merupakan 72% dari keseluruhan *dataset*, sementara data *validation* terdiri dari 28% dari keseluruhan *dataset*. Data *training* dan *validation* ini akan digunakan dalam proses pelatihan dan pembuatan model. Pembagian ini penting untuk mengoptimalkan performa model dengan memastikan bahwa model telah dipelajari dari sejumlah besar data yang cukup bervariasi dan juga dapat diuji dengan data yang tidak pernah dilihat sebelumnya.

e) *Training* Model

Proses *training* model pada penelitian kali ini menggunakan YOLO v8. *Dataset* yang telah siap untuk *training* kemudian di unggah ke platform Google Drive untuk selanjutnya dilakukan

proses *training* dengan terhubung ke platform Google Colaboratory. Model dilatih menggunakan 718 gambar yang terbagi menjadi empat kelas yaitu: *Bacterial Blight, Brown Streak Disease, Green Mottle, Mosaic Disease.* Proses *training* menggunakan 30 putaran (*epochs*) dengan *batch* 16.

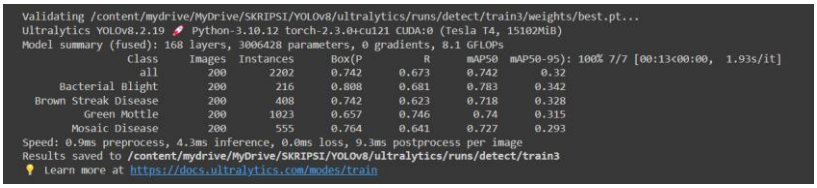
Proses augmentasi pada versi YOLO sebelumnya dilakukan diluar proses *training* model sedangkan dalam versi YOLO yang digunakan pada penelitian kali ini yaitu versi 8, proses augmentasi dilakukan secara otomatis oleh YOLO v8, berikut hasil dari *auto* augmentasi ketika proses *training* dapat dilihat pada Gambar 4.17.



Gambar 4. 17 *Auto* Augmentasi YOLO v8

Ada 4 jenis augmentasi yang diterapkan dalam proses *auto* augmentasi ini, diantaranya adalah: *Lighting Variation* (Variasi Pencahayaan), *Color Transformation* (Transformasi Warna), *Crop* (Pemangkasan), dan *Flip* (Pembalikan). Variasi pencahayaan melibatkan pengubahan intensitas, sudut, atau bayangan pada

gambar untuk membiasakan model dengan berbagai kondisi pencahayaan. Transformasi warna, di sisi lain, melibatkan pengubahan kecerahan, kontras, atau bahkan warna secara keseluruhan untuk mengajarkan model mengenali objek dalam berbagai warna. Pemangkasan, atau *cropping*, melibatkan penghapusan bagian gambar tertentu untuk membantu model fokus pada informasi penting dan mengurangi *noise*. Terakhir, pembalikan, atau *flipping,* melibatkan pembalikan gambar secara horizontal atau vertikal untuk memberikan variasi tambahan pada data latih dan membuat model lebih fleksibel dalam mengenali objek dengan orientasi yang berbeda. Dengan menerapkan berbagai jenis augmentasi ini, *dataset* pelatihan dapat diperluas dan model dapat dilatih untuk mengenali objek dengan lebih baik dalam berbagai situasi dan kondisi. *Output* dari proses *training* ini adalah model dari YOLO v8 berformat .pt. Berikut adalahh *output* dari proses *training* dapat dilihat pada Gambar 4.18.



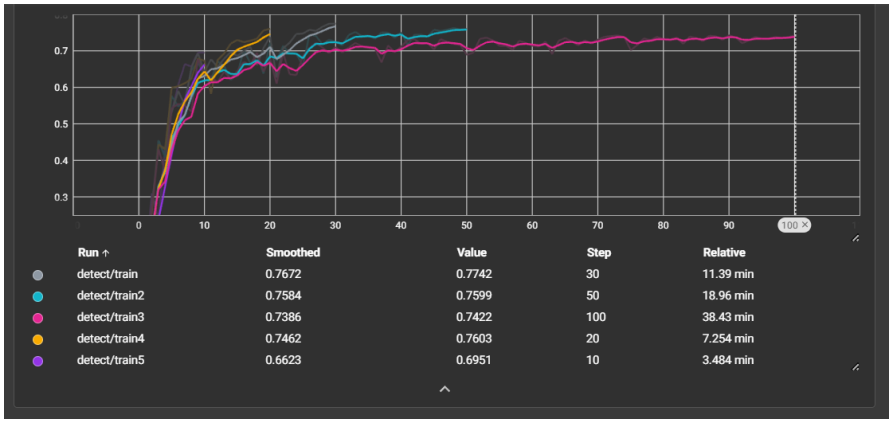
Gambar 4. 18 *Output* Proses *Training*

Pada proses pelatihan model, penulis melakukan beberapa percobaan dengan berbagai jumlah parameter *epoch* (10, 20, 30, 50, dan 100), penulis memutuskan untuk menggunakan *epoch* sebanyak 30 dalam pengembangan model deteksi penyakit pada daun singkong. Berdasarkan hasil percobaan, model dengan 30 *epoch* menunjukkan akurasi terbaik tanpa mengalami *overfitting*. Hal ini terlihat dari performa model yang konsisten baik pada data pelatihan maupun data validasi. Oleh karena itu, *epoch* 30 dipilih sebagai parameter pelatihan untuk mencapai keseimbangan antara akurasi

dan generalisasi model. Berikut adalah tabel dan gambar hasil perbandingan pelatihan dari beberapa parameter dan ditentukan berdasarkan nilai akurasi yang dihitung dari akurasi *confusion matrix* dan grafik *TensorBoard* dapat dilihat pada Tabel 4.6 dan Gambar 4.19.

Tabel 4. 6 Perbandingan Percobaan Parameter Pelatihan

Epoch	Akurasi
10	69%
20	76%
30	77%
50	75%
100	74%



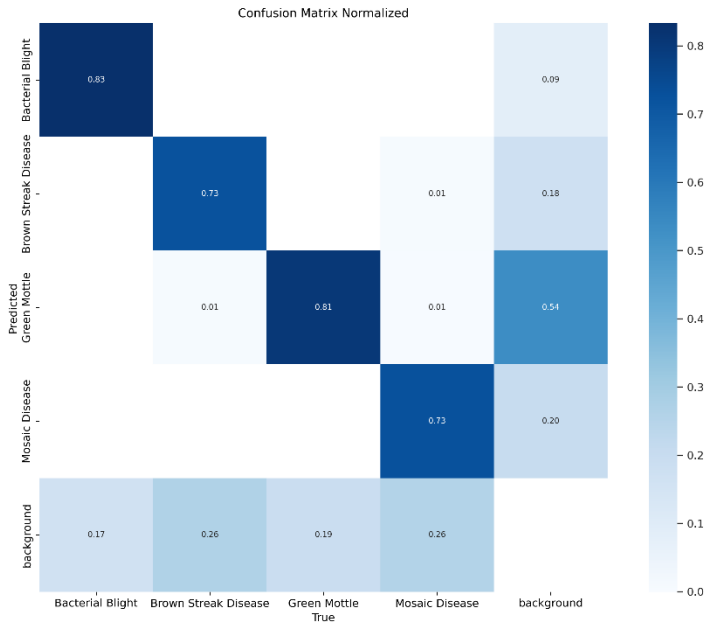
Gambar 4. 19 Grafik Perbandingan TensorBoard

f) Evaluasi Model

1) *Confusion Matrix*

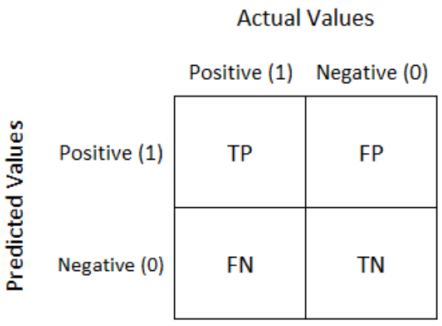
Confusion Matrix digunakan untuk mengevaluasi kinerja deteksi objek dengan membandingkan hasil prediksi model terhadap label yang sebenarnya pada *dataset* validasi. Matriks ini menyajikan jumlah prediksi yang benar (*true positives*), jumlah prediksi yang salah (*false positives*), serta objek yang gagal terdeteksi (*false negatives*) untuk setiap kelas objek yang

diprediksi. Dengan memeriksa matriks kebingungan, pengembang dapat memperoleh wawasan yang lebih baik tentang kelemahan dan kekuatan model dalam mengenali berbagai kelas objek. Berikut adalah hasil *Confusion Matrix* dapat dilihat pada Gambar 4.20.



Gambar 4. 20 *Confusion Matrix*

Untuk menghitung akurasi *Confusion Matrix* dilakukan dengan menerapkan perhitungan persamaan 1 pada rumus berikut ini[34]:



Gambar 4. 21 Perhitungan *Confusion Matrix*

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

$$Accuracy = \frac{83 + 73 + 81 + 73}{17 + 26 + 19 + 26 + 83 + 73 + 81 + 73}$$

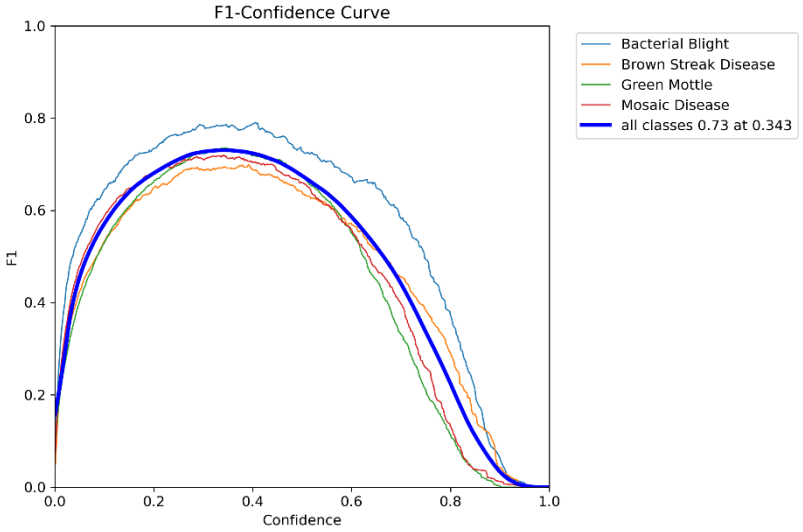
$$= \frac{310}{398} = 0{,}778$$

$$= 0{,}77 \text{ x } 100\% = 77\%$$

Hasil akurasi yang diperoleh untuk seluruh data dalam proses pelatihan model dengan 30 *epoch* adalah 77%. 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 cukup baik dan akurat dalam mendeteksi penyakit pada daun tanaman singkong, namun akurasinya masih belum terlalu tinggi.

2) *F1-Confidence Curve*

F1-Confidence Curve adalah kurva yang menggambarkan hubungan antara *F1-score* (sebuah metrik yang mengukur keseimbangan antara presisi dan *recall*) dan tingkat kepercayaan (*confidence level*) dari model klasifikasi. Kurva ini memberikan pemahaman tentang seberapa baik model dapat mempertahankan keseimbangan antara presisi dan *recall* pada berbagai tingkat kepercayaan. Dengan menganalisis *F1-Confidence Curve*, pengguna dapat memilih *threshold* yang sesuai untuk menyesuaikan *trade-off* antara presisi dan *recall* berdasarkan kebutuhan spesifik dari aplikasi atau kasus penggunaan tertentu. Berikut adalah hasil *F1-Confidence Curve* dapat dilihat pada Gambar 4.22.



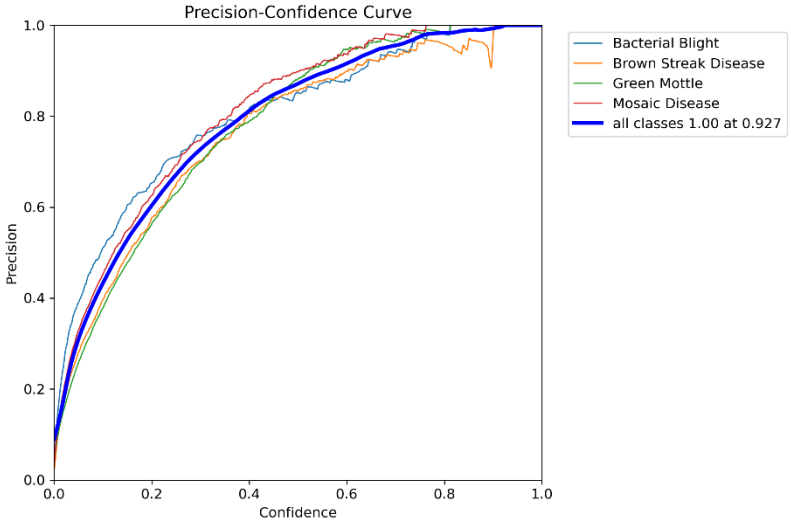
Gambar 4. 22 *F1-Confidence Curve*

Grafik F1 yang dihasilkan dari proses pelatihan model YOLO v8 dengan 30 *epoch*. Dari hasil pelatihan yang diperoleh, skor F1 mencapai nilai rata-rata tertinggi sebesar 0,73 pada nilai *confidence* 0,379. Grafik tersebut menggambarkan hubungan antara nilai *confidence* dan skor F1, yang menunjukkan seberapa baik model dalam menyeimbangkan *precision* dan *recall* pada berbagai tingkat keyakinan dalam prediksi. Nilai *confidence* 0,379 adalah titik di mana model mencapai keseimbangan optimal antara *precision* dan *recall*, menghasilkan skor F1 tertinggi.

3) *Precision-Confidence Curve*

Precision-Confidence Curve adalah alat visualisasi yang mengilustrasikan hubungan antara tingkat presisi dan tingkat kepercayaan (*confidence level*) dari model klasifikasi. Kurva ini memberikan gambaran 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 *threshold* yang sesuai

untuk memenuhi kebutuhan spesifik dari aplikasi atau kasus penggunaan tertentu. Berikut adalah hasil *Precision-Confidence Curve* dapat dilihat pada Gambar 4.23.



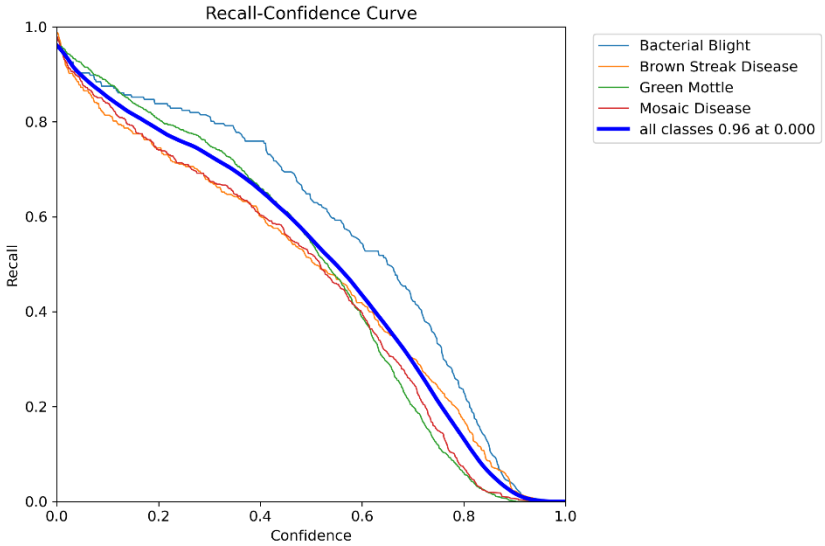
Gambar 4. 23 *Precision-Confidence Curve*

Gambar di atas menunjukkan grafik nilai *precision* terhadap nilai *confidence*. Dari hasil pelatihan yang diperoleh, nilai *precision* mencapai rata-rata maksimal sebesar 1,00 pada nilai *confidence* 0,927. Ini berarti 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 grafik 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 *threshold* yang sesuai untuk mencapai tingkat *recall* yang diinginkan berdasarkan kebutuhan spesifik dari aplikasi atau kasus penggunaan tertentu. Berikut adalah hasil *Recall-Confidence Curve* dapat dilihat pada Gambar 4.24.



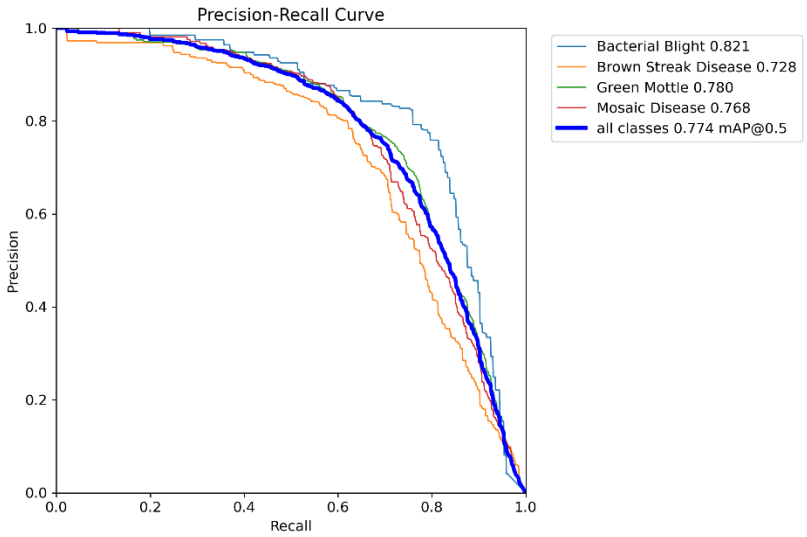
Gambar 4. 24 *Recall-Confidence Curve*

Gambar di atas menunjukkan grafik nilai *recall* terhadap nilai *confidence*. Dari hasil pelatihan yang diperoleh, nilai *recall* mencapai rata-rata maksimal sebesar 0,96 pada nilai *confidence* 0,00. Ini menunjukkan bahwa pada tingkat keyakinan yang sangat rendah (0,00), model mampu mengingat kembali sebagian besar *instance* positif yang sebenarnya.

5) *Precision-Recall Curve*

Precision-Recall Curve adalah sebuah grafik yang menampilkan 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,* yang bergantung pada prioritas dan kebutuhan spesifik dari aplikasi atau kasus penggunaan tertentu. Berikut adalah hasil *Precision-Recall Curve* dapat dilihat pada Gambar 4.25.



Gambar 4. 25 *Precision-Recall Curve*

Gambar di atas menampilkan grafik nilai *Precision-Recall* terhadap nilai *confidence*. Dari hasil pelatihan yang diperoleh, nilai *Precision-Recall* mencapai rata-rata maksimal sebesar 0,774 pada nilai *mAP* 0,5. Semakin tinggi skor *mAP*, semakin baik performa yang dihasilkan. Ini menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi dan *recall* pada tingkat kepercayaan yang ditunjukkan.

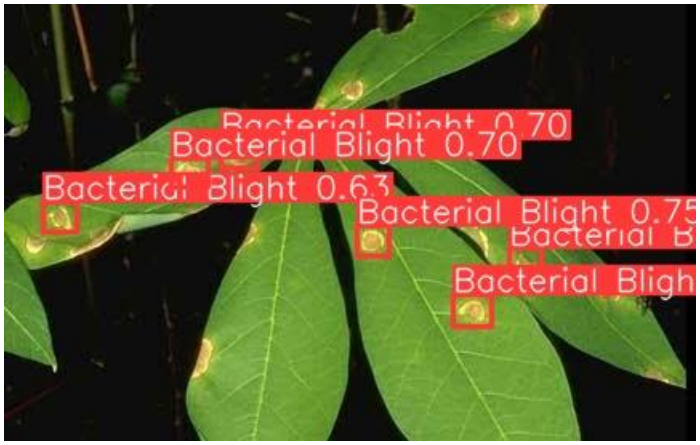
Skor *mAP* atau *"mean Average Precision"* adalah metrik evaluasi umum dalam deteksi objek dalam bidang *computer vision* yang mengukur kemampuan suatu model dalam mengidentifikasi dan mengklasifikasikan objek-objek dalam gambar dengan akurasi tinggi. Proses penghitungan *mAP* melibatkan perhitungan 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-objek yang relevan dalam gambar serta seberapa akurat model dalam mengklasifikasikan objek-objek tersebut, dengan nilai yang lebih tinggi menunjukkan kinerja yang lebih baik dari model deteksi objek. Semakin tinggi skor *mAP*, semakin baik performa yang dihasilkan oleh model dalam menghasilkan prediksi yang akurat dan meminimalkan kesalahan.

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* 0.45 model hanya akan menampilkan *output* dengan skor >= 0.45 Berikut adalah hasil-hasilnya:

1) *Bacterial Blight*



Gambar 4. 26 Hasil Deteksi - *Bacterial Blight*

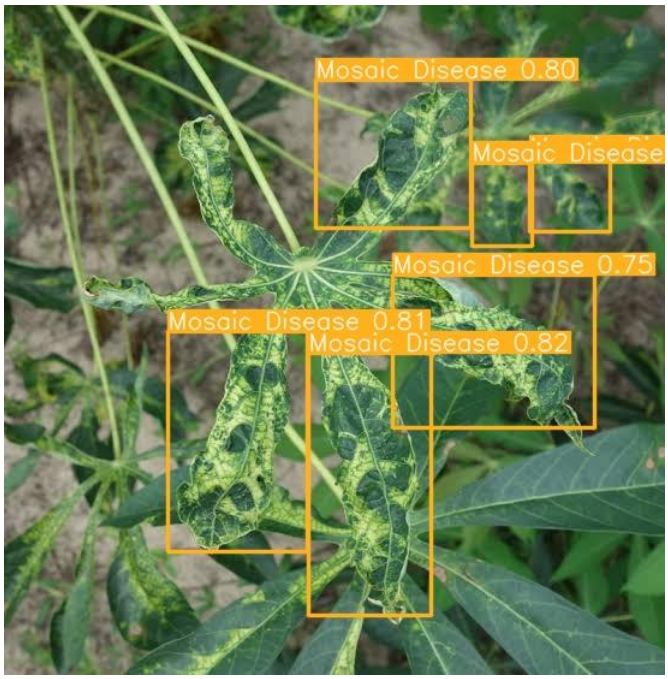
2) *Brown Streak Disease*

Gambar 4. 27 Hasil Deteksi - *Brown Streak*

3) *Green Mottle*

Gambar 4. 28 Hasil Deteksi - *Green Mottle*

4) *Mosaic Disease*

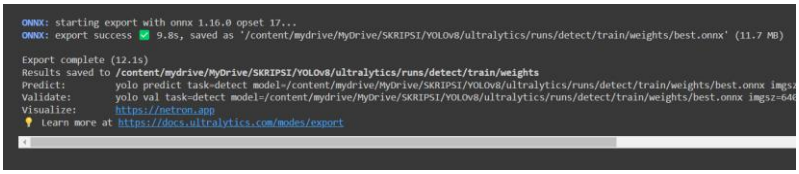


Gambar 4. 29 Hasil Deteksi - *Mosaic Disease*

4. Implementasi Sistem

a) *Convert* Model

Setelah model selesai dikembangkan, selanjutnya adalah mengkonversi format awal model yaitu .pt menjadi .onnx untuk selanjutnya dilakukan *deployment* ke aplikasi Android.

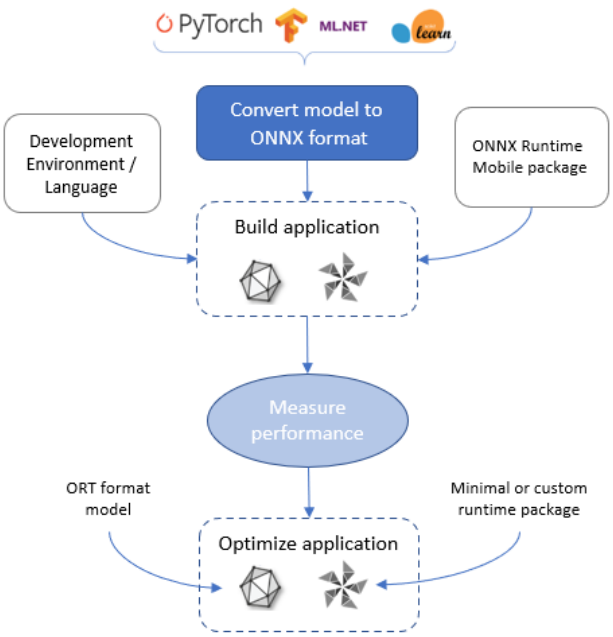


Gambar 4. 30 *Convert Model*

b) *Deployment*

Proses *deployment* ONNX pada proyek *Android Kotlin* melibatkan beberapa langkah. Pertama, penulis mengonversi model dari YOLO v8 ke format ONNX. Setelah memiliki model dalam format ONNX, langkah berikutnya adalah menyiapkan proyek *Android Kotlin* di lingkungan pengembangan *Android Studio*.

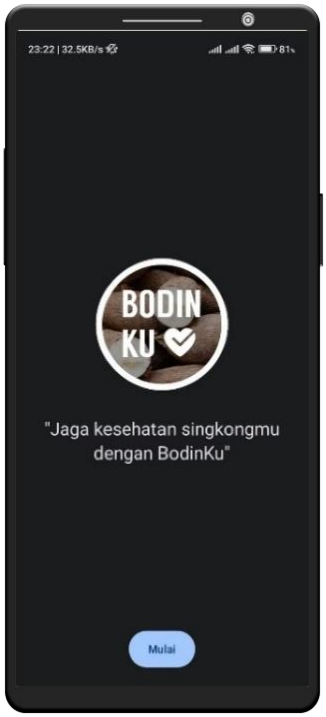
Selanjutnya, tambahkan dependensi atau *library* yang diperlukan seperti *ONNX Runtime for Android*, ke dalam *file* build.gradle. Setelah itu, penulis mengintegrasikan model ONNX ke dalam proyek menggunakan kode fungsi sesuai panduan *deployment* menggunakan Kotlin. Setelah itu penulis melakukan uji coba aplikasi untuk memastikan model berfungsi dengan baik dan lakukan optimisasi jika diperlukan. Terakhir, setelah berhasil dikembangkan projek Android ini siap untuk di *build* menjadi aplikasi Android siap *install*[35].



Gambar 4. 31 Alur *Deployment* ONNX

c) Hasil Aplikasi

1) Halaman *Splash Screen*



Gambar 4. 32 Hasil Aplikasi Halaman *Splash Screen*

Splash screen adalah halaman pertama yang muncul saat membuka aplikasi. Pada halaman ini, terdapat informasi mengenai aplikasi yang berupa tagline. Ini menunjukkan bahwa aplikasi telah berhasil diakses. Untuk memulai aplikasi, pengguna perlu menekan tombol ‘Mulai’.

2) Halaman Artikel

Gambar 4. 33 Hasil Aplikasi Halaman Artikel

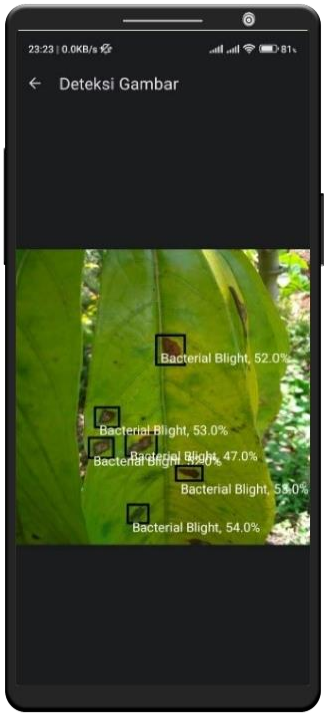
Halaman ini akan terbuka ketika pengguna selesai menekan tombol ‘Mulai’ pada *splash screen*. Halaman ini juga bisa diakses melalui navigasi aplikasi yang terletak di bawah. Pada halaman ini pengguna bisa memilih artikel untuk dibaca, untuk memilihnya pengguna bisa melakukan *scrolling* atau gulir layar ke bawah untuk melihat artikel lainnya. Untuk membuka artikelnya pengguna perlu menekan pada *thumbnail* atau judul pada artikel yang ingin dibaca.

3) Halaman Deteksi *Realtime*

Gambar 4. 34 Hasil Aplikasi Halaman Deteksi *Realtime*

Halaman ini adalah fitur utama pada aplikasi deteksi penyakit pada daun tanaman singkong yaitu fitur *realtime detection* atau deteksi secara langsung. Untuk mengakses halaman ini pengguna bisa membukanya melalui navigasi yang ada pada aplikasi. Cara kerja metode ini adalah pengguna hanya perlu mengarahkan ke objek yang ingin dideteksi lalu aplikasi akan memunculkan hasilnya.

4) Halaman Deteksi Via *Upload*



Gambar 4. 35 Hasil Aplikasi Halaman Deteksi Via *Upload*

Halaman ini juga termasuk fitur utama dalam aplikasi yang dikembangkan penulis yaitu pilihan kedua setelah deteksi secara langsung. Untuk mengakses halaman ini pengguna cukup menekan tombol ‘Galeri’ pada halaman deteksi secara langsung atau *realtime* maka aplikasi akan mengarahkan pengguna untuk membuka galeri. Untuk melakukan deteksi melalui metode ini pengguna hanya memilih gambar yang ingin dideteksi yang ada di galeri, setelah pengguna memilih gambarnya maka otomatis aplikasi akan mendeteksi.

5) Halaman *About*

Gambar 4. 36 Hasil Aplikasi Halaman *About*

Halaman ini adalah halaman terakhir pada aplikasi yang dikembangkan oleh penulis. Untuk mengakses halaman ini pengguna bisa membukanya melalui navigasi yang ada dalam aplikasi. Halaman ini berfungsi untuk menampilkan keterangan tentang aplikasi.

5. Pengujian

Pengujian dilakukan untuk memastikan kinerja aplikasi yang baik. Dalam penelitian ini jenis pengujian yang digunakan yaitu *Black Box* dan *User Acceptance Testing* (UAT).

a) 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.

1) Hasil Perhitungan Pengujian *Black Box*

Hasil pengujian *Black Box* dapat dilihat pada tabel 4.7.

Tabel 4. 7 Hasil Pengujian *Black Box*

Nama Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan					
				Diterima			Ditolak		
				1	2	3	1	2	3
Splash Screen	*User* menekan tombol mulai.	*User* dapat menekan tombol mulai dan dapat masuk ke halaman artikel.	Aplikasi akan menampilkan halaman artikel.	✓	✓	✓			
Navigasi	*User* menekan beberapa tombol pada navigasi.	*User* dapat masuk ke halaman yang dituju dari navigasi aplikasi.	Aplikasi akan menampilkan halaman dituju.	✓	✓	✓			
Halaman Artikel	*User* melakukan gulir atau *scrolling* kebawah.	*User* dapat melihat daftar artikel.	Aplikasi akan menampilkan daftar artikel.	✓	✓	✓			

	User melakukan menekan pada *thumbnail* gambar atau judul artikel.	*User* dapat melihat isi artikel.	Aplikasi akan menampilkan keseluruhan isi artikel.	✓	✓	✓			
Halaman Deteksi	*User* mengarahkan kamera ke objek yang akan dideteksi.	*User* dapat melihat hasil deteksi *realtime*.	Aplikasi akan menampilkan hasil deteksi *realtime*.	✓	✓	✓			
	User menekan tombol ‘Galeri’ pada halaman deteksi.	*User* dapat memilih gambar dari galeri untuk dideteksi.	Aplikasi akan menampilkan galeri dan akan menampilkan hasil deteksi ketika *user* telah memilih gambar.	✓	✓	✓			
Halaman Tentang	*User* menekan tombol ‘Tentang’ pada navigasi aplikasi.	*User* dapat melihat isi halaman tentang aplikasi.	Aplikasi akan menampilkan halaman tentang aplikasi.	✓	✓	✓			

2) Kesimpulan Hasil Pengujian *Black Box*

Berdasarkan pengujian *Black Box,* dari 7 pengujian pada aplikasi yang didapat dari 3 responden, berikut ini hasil pengujian *Black Box*:

- Pengujian Pertama

  Tercapai: $\frac{7}{7} \times 100\% = 100\%$

  Gagal: $\frac{0}{7} \times 100\% = 0\%$

- Pengujian Kedua

  Tercapai: $\frac{7}{7} \times 100\% = 100\%$

  Gagal: $\frac{0}{7} \times 100\% = 0\%$

- Pengujian Ketiga

  Tercapai: $\frac{7}{7} \times 100\% = 100\%$

  Gagal: $\frac{0}{7} \times 100\% = 0\%$

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

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 presentasenya adalah 0%. Kesimpulannya, aplikasi berjalan sesuai dengan fungsinya yang diharapkan.

b) Pengujian *User Acceptance Testing* (UAT)

Dalam pengujian ini, UAT dilakukan oleh lima responden yang terdiri dari masyarakat umum dan para petani singkong. Pengujian ini melibatkan pengguna yang akan menggunakan aplikasi dalam kehidupan sehari-hari mereka, memberikan umpan balik yang berharga mengenai kinerja dan kegunaannya. Melalui

proses ini, pengembang dapat memastikan bahwa aplikasi tidak hanya berfungsi secara teknis tetapi juga memuaskan kebutuhan dan ekspektasi pengguna, khususnya dalam membantu petani mendeteksi penyakit pada daun tanaman singkong secara efektif.

1) Hasil Perhitungan Pengujian *User Acceptance Testing* (UAT)
   Berikut ini merupakan hasil kuesioner pengujian *User Acceptance Testing* (UAT) yang telah disebarkan kepada 5 responden. Hasil pengujian UAT dapat dilihat pada Tabel 4.8.

Tabel 4. 8 Hasil Pengujian *User Acceptance Testing* (UAT)

Pertanyaan	Hasil Pengujian				
	Responden 1	Responden 2	Responden 3	Responden 4	Responden 5
1	5	4	4	5	5
2	5	4	4	5	5
3	5	5	4	5	5
4	5	3	4	4	3
5	5	4	3	4	5
6	3	4	4	5	5
7	4	4	5	5	4
8	4	4	4	5	4
9	3	3	4	4	4
10	5	5	5	5	5
Jumlah Skor	44	40	41	47	45
Presentase	88%	80%	82%	94%	90%
Total	434%				

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:

Presentase rata-rata = $\frac{Jumlah\ total\ presentasi}{Jumlah\ responden}$

Presentase rata-rata = $\frac{434\%}{5} = 86{,}8\ \%$

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 86,8%. Dengan demikian, dapat disimpulkan bahwa pengujian UAT pada aplikasi ini memperoleh kategori yang sangat baik.

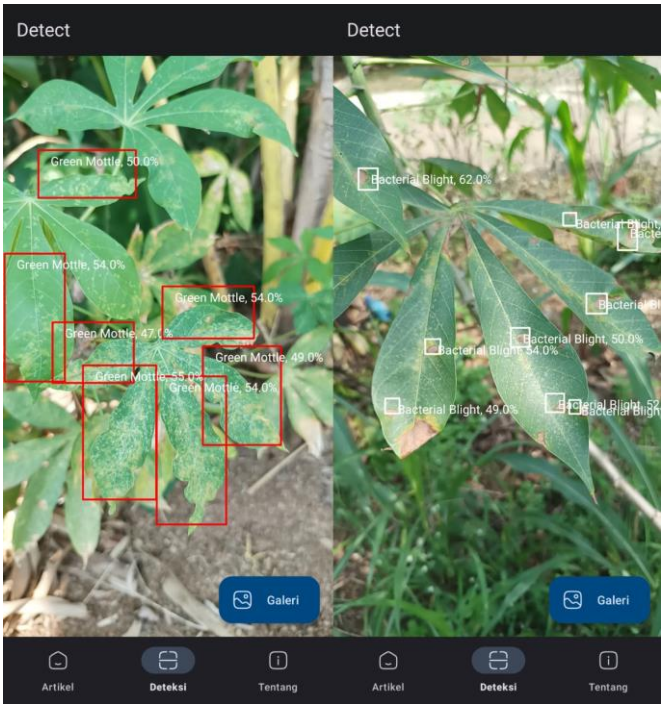
c) Pengujian Lapangan

Pengujian lapangan aplikasi deteksi penyakit pada daun tanaman singkong dilakukan di Perkebunan Singkong Dukuh Beber, Kec. Randudongkal, Kab. Pemalang, Jawa Tengah. Pengujian ini bertujuan untuk mengevaluasi kemampuan aplikasi dalam mendeteksi penyakit secara langsung pada daun tanaman singkong di kondisi lapangan.

Gambar 4. 37 Pengujian Lapangan - Bersama Para Petani

Selama survei yang dilakukan di berbagai titik dalam perkebunan singkong, teridentifikasi bahwa dua jenis penyakit utama yang menyerang tanaman singkong adalah *Green Mottle* dan *Bacterial Blight*. *Green Mottle* ditandai dengan bercak titik putih pada daun dan mempunyai nama lain yaitu *Tunggau*, sementara *Bacterial Blight* menyebabkan layu dan bercak cokelat pada daun dan dikenal juga dengan nama *Rapak*.

Para petani telah memverifikasi bahwa kedua penyakit ini diakibatkan oleh ulat. Kedua penyakit ini cukup umum dan dapat menyebabkan kerugian signifikan jika tidak segera ditangani. Umumnya, para petani di sini mengatasi penyakit *Rapak* dan *Tunggau* menggunakan *Endrin* (Obat tanaman). Berikut adalah hasil deteksi *realtime* pada daun tanaman singkong yang ada di perkebunan Dukuh Beber:



Gambar 4. 38 Pengujian Lapangan - Deteksi *Realtime*

Pada pengujian lapangan, tidak ditemukan adanya penyakit *Brown Streak* dan *Mosaic Disease* di perkebunan ini, 2 penyakit ini adalah 2 *class* penyakit yang lain selain *Bacterial Blight* dan *Green Mottle*. Brown Streak biasanya ditandai dengan bercak kuning pada daun. Sementara Mosaic Disease ditandai dengan pola mosaik pada daun yang disebabkan oleh infeksi virus. Ketiadaan kedua penyakit ini menunjukkan bahwa perkebunan ini mungkin memiliki kondisi yang tidak mendukung penyebaran kedua penyakit tersebut atau mungkin adanya praktik pengelolaan tanaman yang efektif.

Hasil dari pengujian lapangan ini mengindikasikan bahwa aplikasi deteksi penyakit pada daun tanaman singkong bisa menjadi alat yang sangat berguna bagi para petani. Dengan kemampuan untuk melakukan deteksi dini dan akurat, aplikasi ini dapat membantu mengurangi kerugian akibat penyakit tanaman, meningkatkan hasil panen, dan mendukung praktik pertanian yang lebih berkelanjutan. Perkebunan Singkong Dukuh Beber di

Kecamatan Randudongkal menjadi contoh nyata bagaimana teknologi modern dapat diterapkan untuk meningkatkan efisiensi dan produktivitas dalam sektor pertanian.

Gambar 4. 39 Pengujian Lapangan - Lingkungan Perkebunan

B. Pembahasan

1. Analisis Kebutuhan

Pada tahap awal, yaitu analisis kebutuhan, diperlukan proses analisis yang bertujuan untuk merinci kebutuhan yang akan digunakan dalam pengembangan aplikasi. Ini meliputi analisis kebutuhan data, sistem, serta fungsional. Pada tahap ini, kebutuhan untuk aplikasi deteksi penyakit pada tanaman singkong akan dikembangkan dikumpulkan, dipahami, dan diuraikan secara detail. Analisis kebutuhan data mencakup identifikasi jenis data yang diperlukan, seperti citra penyakit daun singkong. Pengumpulan data yang akurat sangat penting untuk melatih model deteksi penyakit dengan tepat.

Analisis kebutuhan sistem melibatkan identifikasi infrastruktur teknis yang dibutuhkan untuk mendukung aplikasi. Ini termasuk perangkat keras yang diperlukan untuk pengambilan gambar dan pemrosesan data, serta perangkat lunak yang digunakan untuk pengembangan aplikasi. Secara keseluruhan, tahap analisis kebutuhan ini sangat krusial karena menjadi dasar bagi desain dan pengembangan

aplikasi selanjutnya. Dengan memahami dan merinci kebutuhan secara detail, pengembang dapat memastikan bahwa aplikasi yang dibuat akan sesuai dengan kebutuhan pengguna dan mampu memberikan solusi yang efektif untuk deteksi penyakit pada daun tanaman singkong.

2. Desain

Setelah selesai mengumpulkan kebutuhan aplikasi, langkah berikutnya adalah tahap desain. Dalam tahap ini, penulis menggunakan model perancangan *Unified Modelling Language* (UML), yang mencakup empat jenis diagram UML yaitu: *Use Case, Activity, Sequence,* dan *Class Diagram*. Selain menggunakan model UML, penulis juga menggunakan desain antarmuka atau *User Interface* (UI). Dalam desain antarmuka ini, penulis merancang *Wireframe* dengan desain sederhana untuk mempermudah dalam 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. Diagram 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. Diagram ini menunjukkan langkah-langkah yang harus diambil dari awal hingga akhir dalam berbagai proses, seperti proses deteksi penyakit. Dengan menggunakan *Activity Diagram*, penulis dapat memastikan bahwa setiap proses dijelaskan secara rinci dan efisien.

Sequence Diagram digunakan untuk menggambarkan interaksi antara objek dalam urutan waktu tertentu. Diagram ini membantu dalam memahami urutan kejadian dalam sistem, seperti bagaimana data dikirim, bagaimana hasil analisis diproses, dan bagaimana informasi disajikan kepada pengguna. Dengan *Sequence Diagram*, penulis dapat

memastikan bahwa interaksi antar komponen sistem berjalan dengan lancar dan sesuai dengan yang diharapkan.

Class Diagram digunakan untuk menggambarkan struktur sistem dalam hal kelas-kelas dan hubungan antar kelas tersebut. Diagram ini membantu dalam menentukan atribut dan metode yang dimiliki oleh setiap kelas, serta bagaimana kelas-kelas tersebut saling berinteraksi. Dengan *Class Diagram*, penulis dapat memastikan bahwa struktur data dan logika aplikasi terorganisir dengan baik dan mendukung kebutuhan fungsional yang telah diidentifikasi.

Selain model UML, desain antarmuka atau *User Interface* (UI) juga menjadi fokus utama dalam tahap desain ini. Penulis merancang *Wireframe* dengan desain sederhana untuk mempermudah pengembangan aplikasi. *Wireframe* ini berfungsi sebagai *blueprint* visual yang menunjukkan tata letak dasar dari elemen-elemen antarmuka pengguna, seperti tombol, menu, dan formulir *input*. Dengan *Wireframe*, penulis dapat memastikan bahwa antarmuka pengguna intuitif dan mudah digunakan, sehingga pengguna dapat berinteraksi dengan aplikasi dengan efisien dan efektif.

3. Implementasi

Implementasi yang pertama adalah implementasi model. Model dikembangkan dari mulai pengolahan *dataset* sampai dengan *training* model. Setelah model selesai dikembangkan, selanjutnya adalah implementasi kedua yaitu implementasi sistem. Pada implementasi sistem, model yang sudah selesai di *training* dan menghasilkan *output* .pt kemudian di konversi menjadi format .onnx kemudian di *deploy* ke *projek* Android untuk pengembangan lebih lanjut. Setelah model .onnx berhasil diintegrasikan ke dalam aplikasi Android, langkah berikutnya adalah melakukan pengujian untuk memastikan bahwa model berfungsi dengan baik di lingkungan Android. Pengujian ini meliputi verifikasi akurasi prediksi, performa, dan *responsivitas* model ketika diakses

melalui aplikasi Android. Setelah pengujian selesai dan hasilnya memuaskan, aplikasi Android yang mengandung model ini siap untuk didistribusikan atau dipublikasikan untuk digunakan oleh pengguna.

4. Pengujian

Langkah terakhir dalam proses pengembangan adalah pengujian untuk memverifikasi kualitas dari aplikasi yang dikembangkan. Penulis melakukan 3 jenis pengujian: *Black Box Testing*, *User Acceptance Testing* (UAT), dan Pengujian Lapangan. Dalam *Black Box Testing*, hasilnya adalah 100% keberhasilan, sedangkan pengujian yang gagal mendapat 0% dari 3 responden dan 7 pengujian dilengkapi dengan masukan dari responden yang tercantum di Lampiran. Selanjutnya, pengujian *User Acceptance Testing* (UAT) menghasilkan tingkat keberhasilan yang sangat baik, dengan presentase sebesar 83,3% dari 5 responden yang menjawab 10 pertanyaan. Yang terakhir adalah pengujian lapangan menghasilkan hasil bahwa aplikasi dapat mendeteksi penyakit pada daun tanaman singkong dengan baik dan akurat.

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* penyakit daun pada tanaman singkong menggunakan algoritma YOLO v8 dengan 30 *epoch* dan jumlah *dataset* sebanyak 718, diperoleh akurasi sebesar 77%. *Dataset* tersebut dibagi menjadi 518 data (72%) untuk pelatihan dan 200 data (28%) untuk validasi. Hasil ini menunjukkan bahwa model yang dikembangkan memiliki kemampuan yang cukup baik dalam mendeteksi penyakit pada daun singkong. 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 penyakit tanaman berbasis Android.
2. Pengujian *Black Box* dilakukan oleh 3 responden dengan total 7 pengujian. Beberapa halaman yang diuji meliputi *splash screen*, navigasi, halaman artikel, halaman deteksi, halaman tentang aplikasi. Hasil dari pengujian ini menunjukkan tingkat keberhasilan sebesar 100%, dengan tingkat kegagalan 0%.
3. Pengujian *User Acceptance Testing* dilakukan oleh 5 responden menggunakan 10 pertanyaan. Pertanyaan mencakup aspek kegunaan, kemudahan penggunaan, dan *User Interface* (UI). Hasil pengujian ini menunjukkan persentase sebesar 86,8% dengan kategori sangat baik.
4. Pengujian lapangan aplikasi deteksi penyakit pada daun tanaman singkong di Perkebunan Singkong Dukuh Beber, Kec. Randudongkal, Kab. Pemalang, Jawa Tengah, menunjukkan bahwa aplikasi yang dikembangkan oleh penulis mampu secara akurat mendeteksi penyakit

Rapak (*Bacterial Blight*) dan Tunggau (*Green Mottle*) yang umum menyerang tanaman singkong di perkebunan singkong Dukuh Beber. Para petani di sini umumnya mengatasi kedua penyakit tersebut menggunakan Endrin. Selain itu, tidak ditemukan adanya penyakit *Brown Streak* dan *Mosaic Disease* di perkebunan ini. Hasil ini menunjukkan potensi besar aplikasi teknologi dalam meningkatkan efisiensi dan produktivitas dalam pertanian.

5. Potensi Aplikasi berbasis Android ini memiliki potensi besar untuk digunakan oleh petani dan agronomis dalam mendeteksi penyakit pada daun singkong secara cepat dan akurat, yang dapat membantu dalam pengambilan keputusan terkait penanganan tanaman.
6. 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 singkong.

B. Saran

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

1. Mengeksplorasi kemungkinan untuk mengembangkan aplikasi deteksi penyakit pada daun tanaman singkong untuk platform lain seperti IOS atau *website*, sehingga dapat diakses oleh lebih banyak pengguna.
2. Melakukan optimasi lebih lanjut terhadap model deteksi penyakit menggunakan algoritma YOLO v8 untuk meningkatkan akurasi dan keandalan dalam mendeteksi berbagai jenis penyakit pada daun tanaman singkong.
3. Fokus pada pengembangan teknologi untuk meningkatkan kemampuan deteksi secara *real-time*, sehingga aplikasi dapat memberikan respons yang lebih cepat dan lebih presisi saat digunakan oleh pengguna di lapangan. Ini dapat melibatkan penyempurnaan kode pemrograman dalam projek aplikasi.

DAFTAR PUSTAKA

[1] Mirza Faturrachman, Indra Yustiana, and Somantri, “SISTEM PENDETEKSI PENYAKIT PADA DAUN TANAMAN SINGKONG MENGGUNAKAN DEEP LEARNING DAN TENSORFLOW BERBASIS ANDROID,” *IJIS - Indonesian Journal on Information System*, vol. 7, no. 2, p. 177, Sep. 2022.

[2] Lusiana Rahma, Hadi Syaputra, A.Haidar Mirza, and Susan Dian Purnamasari, “Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once),” *Jurnal Nasional Ilmu Komputer*, vol. 2, no. 3, p. 213, Aug. 2021.

[3] Mohammad Syarief, Amirul Mukminin, Novi Prastiti, and Wahyudi Setiawan, “PENERAPAN METODE NAÏVE BAYES CLASSIFIER UNTUK DETEKSI PENYAKIT PADA TANAMAN JAGUNG,” *Jurnal Ilmiah NERO*, vol. 3, no. 1, pp. 61–68, 2017.

[4] Lutfi Hakim, Sepyan Purnama Kristanto, Dianni Yusuf, Aditya Roman Asyari, and Khoirul Umam, “SISTEM DETEKSI PENYAKIT DAN CRAWLING INFORMASI PADA TANAMAN BUAH NAGA BERBASIS WEB DAN ANDROID,” *JURNAL TEKNOINFO*, vol. 17, no. 1, pp. 27–35, Jan. 2023.

[5] Yohani Setiya Rafika Nur, Auliya Burhanuddin, Dasril Aldo, and Widya Lelisa Army, “Sistem Pakar Deteksi Penyakit Bawang Merah dengan Metode Case Based Reasoning,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 3, pp. 1356–1366, Jul. 2022.

[6] Vita Indri Safitri, “SISTEM PAKAR DETEKSI PENYAKIT TANAMAN KENTANG MENGGUNAKAN METODE CERTAINTY FACTOR,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 1, no. 1, pp. 798–805, Mar. 2017.

[7] Inuk Wahyuni Istiqomah and Angga Martha Mahendra, “Pemberdayaan Masyarakat Melalui Penyuluhan Inovasi Pengolahan Singkong Dan Opak Sebagai Upaya Pengembangan Produk Unggulan Di Desa Bleberan Kecamatan Jatirejo Kabupaten Mojokerto,” *Jurnal Pengabdian kepada MasyarakatInstitut Teknologi dan Bisnis Asia Malang*, vol. 3, no. 1, p. 26, May 22AD.

[8] Oyewola DO, Dada EG, Misra S, and Damaševičius R, “Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing.,” *PeerJ Comput Sci*, vol. 7, Mar. 2021.

[9] SUWARTIJAH, Tjuk, and Prof.Dr.Ir. Haryono Semangun, “Kajian penyakit hawar bakteri ubi kayu yang disebabkan oleh Xanthomonas campestris pv. Manihotis dan mekanisme ketahanan tanaman,” https://etd.repository.ugm.ac.id/home/detail_pencarian/12889.

[10] F. Robson, D. L. Hird, and E. Boa, “Cassava brown streak: A deadly virus on the move,” *Plant Pathol*, vol. 73, no. 2, pp. 221–241, Feb. 2024, doi: 10.1111/ppa.13807.

[11] Amelia Devi Putri A, Salsabiil Hasanah, M. Bahrul Subkhi, and Nanik Suciati, “Analisis Penggunaan Pra-proses pada Metode Transfer Learning untuk Mendeteksi Penyakit Daun Singkong,” *Techno.COM*, vol. 22, no. 2, p. 338, May 2023.

[12] Nasir Saleh, Mudji Rahayu, Sri Wahyuni Indiati, Budhi Santoso Radjit, and Sri Wahyuningsih, *HAMA, PENYAKIT, DAN GULMA PADA TANAMAN UBI KAYU*. Bogor, 2013.

[13] Intan Mayla Faiza, Gunawan, and WrestiAndriani, “Tinjauan Pustaka Sistematis: Penerapan Metode Machine Learninguntuk Deteksi Bencana Banjir,” *Jurnal Minfo Polgan*, vol. 11, no. 2, p. 60, Sep. 2022.

[14] K. Kristiawan, D. D. Somali, T. A. Linggan jaya, and A. Widjaja, “Deteksi Buah Menggunakan Supervised Learning dan Ekstraksi Fitur untuk Pemeriksa Harga,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 3, Dec. 2020, doi: 10.28932/jutisi.v6i3.3029.

[15] Desta Yolanda, Mohammad Hafiz Hersyah, and Eno Marozi, “Implementasi Metode Unsupervised Learning Pada Sistem Keamanan Dengan Optimalisasi Penyimpanan Kamera IP,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 6, pp. 1099–1105, Dec. 2021, doi: 10.29207/resti.v5i6.3552.

[16] J. Andreanus and A. Kurniawan, “Sejarah, Teori Dasar dan Penerapan Reinforcement Learning: Sebuah Tinjauan Pustaka,” *Jurnal Telematika*, vol. 12, no. 2, pp. 113–118, Jul. 2018, doi: 10.61769/telematika.v12i2.193.

[17] sis.binus.ac.id, “Mengenal Deep Learning Beserta Contoh Penerapannya,” https://sis.binus.ac.id/2023/07/18/mengenal-deep-learning-beserta-contoh-penerapannya/.

[18] Y. Devianto and S. Dwiasnati, “Kerangka Kerja Sistem Kecerdasan Buatan dalam Meningkatkan Kompetensi Sumber Daya Manusia Indonesia,” *Jurnal Telekomunikasi dan Komputer*, vol. 10, no. 1, p. 19, Apr. 2020, doi: 10.22441/incomtech.v10i1.7460.

[19] Y. A. Suwitono and F. J. Kaunang, “Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Daun Dengan Metode Data Mining SEMMA Menggunakan Keras,” *Jurnal Komtika (Komputasi dan Informatika)*, vol. 6, no. 2, pp. 109–121, Nov. 2022, doi: 10.31603/komtika.v6i2.8054.

[20] Ii Munadhif, Debri Hasbi Fathoni, and Mohammad Abu Jamiin, “PENGENDALIAN CCTV MENGGUNAKAN YOU ONLY LOOK ONCE (YOLO) ,” *Seminar Nasional Terapan Riset Inovatif (SENTRINOV) Ke-6*, vol. 6, no. 1, p. 959, 2020.

[21] Muhamad Rizky Fauzan and Ari Purno Wahyu W, “PENDETEKSIAN PLAT NOMOR KENDARAAN MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE V3 DAN TESSERACT,” *Jurnal Ilmiah Teknologi Informasi Terapan*, vol. 8, no. 1, p. 58, Dec. 2021.

[22] M Abdy Mulya, Zaenul Arif, and Syefudin, “Tinjauan Pustaka Sistematis : Penerapan Metode Gabor WaveletPada Computer Vision,” *J Comput Sci Technol*, vol. 1, no. 2, p. 83, May 2023.

[23] Agung Rilo Pambudi, Garno, and Purwantoro, “DETEKSI KEASLIAN UANG KERTAS BERDASARKAN WATERMARK DENGAN PENGOLAHAN CITRA DIGITAL,” *JIP (Jurnal Informatika Polinema)*, vol. 6, no. 4, pp. 69–74, Aug. 2020.

[24] mie.binus.ac.id, “Teknik pre-processing dan classification dalam data science,” https://mie.binus.ac.id/2022/08/26/teknik-pre-processing-dan-classification-dalam-data-science/.

[25] Nur Azis, Gali Pribadi, and Manda Savitrie Nurcahya, “Analisa dan Perancangan Aplikasi Pembelajaran Bahasa Inggris Dasar Berbasis Android,” *Jurnal IKRA-ITH Informatika*, vol. 4, no. 3, p. 2, Nov. 2020.

[26] onnx.ai, “Open Neural Network Exchange,” https://onnx.ai/.

[27] Putu Pasek Okta Mahawardana, Gusti Arya Sasmita, and I Putu Agus Eka Pratama, “Analisis Sentimen Berdasarkan Opini dari Media Sosial Twitter terhadap ‘Figure Pemimpin’ Menggunakan Python,” *JITTER- Jurnal Ilmiah Teknologi dan Komputer*, vol. 3, no. 1, 2022.

[28] Asep Toyib Hidayat, Rio, and I Gede Olka Santoso, “MEMBERSHIPPLICATION BERBASIS ANDROID DENGAN PENERAPAN KOTLIN PROGRAMMING LANGUAGE DI WIJAYA FITNESS CENTER (WFC),” *Jurnal Sistem Informasi Musi Rawas*, vol. 8, no. 1, p. 10, Jun. 2023.

[29] Dede Wira Trise Putra and Rahmi Andriani, “Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD,” *Jurnal TEKNOIF*, vol. 7, no. 1, pp. 32–39, Apr. 2019.

[30] Yahya Dwi Wijaya and Muna Wardah Astuti, “Sistem Informasi Penjualan Tiket Wisata Berbasis Web Menggunakan Metode Waterfall,” *Seminar Nasional Teknologi Informasi dan Komunikasi 2019*, 2019.

[31] H. Kurniawan, W. Apriliah, I. Kurniawan, and D. Firmansyah, “Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Penggajian Pada SMK Bina Karya Karawang,” *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 14, no. 4, pp. 13–23, Jan. 2020, doi: 10.35969/interkom.v14i4.58.

[32] Anisya Caty Praniffa, Alfi Syahri, Fitriani Sandes, Umi Fariha, Qhoiril Aldi Giansyah, and Muhammad Luthfi Hamzah, “PENGUJIAN BLACK BOX DAN WHITE BOX SISTEM INFORMASI PARKIR BERBASIS WEB,” *Jurnal Testing dan Implementasi Sistem Informasi*, vol. 4, no. 1, p. 4, Jan. 2023.

[33] Bayu Priyatna, April Lia Hananto, and Muhammad Nova, “Application of UAT (User Acceptance Test) Evaluation Model in Minggon E-Meeting Software Development,” *SYSTEMATICS*, vol. 2, no. 2, p. 112, Dec. 2020.

[34] Nadini Mardiah Yasen, Silfia Rifka, Rikki Vitria, and Yulindon, “Pemanfaatan Yolo Untuk Deteksi Hama Dan Penyakit Pada Daun Cabai Menggunakan Metode Deep Learning,” *Jurnal Ilmiah Politeknik Negeri Padang*, vol. 15, no. 2, p. 68, Dec. 2023.

[35] onnxruntime.ai, “How to develop a mobile application with ONNX Runtime,” https://onnxruntime.ai/docs/tutorials/mobile/.

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 : upgrismg@gmail.com, Homepage : www.upgrismg.ac.id

LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : TEGAR RAMADHANI
NPM : 20670038
Program Studi : Informatika
Judul Skripsi : Analisis Deteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Algoritma You Only Look Once (YOLO) v8 Berbasis Android

Dosen Pembimbing I : Febrian Murti Dewanto, SE., M.Kom
Dosen Pembimbing II : Ade Tirtawangsa Kurniadi, S.Kom, M.Kom

No.	Hari Tanggal	Uraian Bimbingan	Paraf
1	07/03/24	Bimbingan Judul	
2	14/03/24	Bimbingan Bab 1	
3	21/03/24	Bimbingan Bab 2	
4	02/05/24	Bimbingan Bab 3	
5	13/05/24	Bimbingan Bab 4	
6	30/5/24	Revisi Bab 4	
7	3/6/24	Bimbingan Bab 5	
8	3/6/24	ACC Skripsi	

Dosen Pembimbing I,
Febrian Murti Dewanto, SE., M.Kom.
NIDN. 0606027801

Mahasiswa,
Tegar Ramadhani
NPM. 20670038

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 : TEGAR RAMADHANI

NPM : 20670038

Program Studi : Informatika

Judul Skripsi : Aplikasi Deteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Algoritma You only Look Once (YOLO) v8 Berbasis Android

Dosen Pembimbing I : Febrian Murti Dewanto, SE., M.Kom

Dosen Pembimbing II : Aris Trijaka Harjanta, S.Kom., M.Kom

No.	Hari Tanggal	Uraian Bimbingan	Paraf
1	07/02/24	Bimbingan Judul	
2	14/03/24	Bimbingan Bab 1	
3	21/03/24	Bimbingan Bab 2	
4	02/05/24	Bimbingan Bab 3	
5	15/05/24	Bimbingan Bab 4	
6	30/05/24	Revisi Bab 4	
7	03/06/24	Bimbingan Bab 5	
8	3/6/24	ACC	

Dosen Pembimbing II,

Aris Trijaka Harjanta, S.Kom., M.Kom.
NIDN. 0619048202

Mahasiswa,

Tegar Ramadhani
NPM. 20670038

Lampiran 2 Kuesioner Pengujian *Black Box*

Kuesioner Pengujian *Black Box* pada Aplikasi Deteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Algoritma *You Only Look Once* (YOLO) v8 Berbasis Android

Nama Penguji : Bambang Agus H, S.kom., M.kom
Tanggal Pengujian : 27 Mei 2024

Nama Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Diterima	Ditolak
Splash Screen	*User* menekan tombol mulai.	*User* dapat menekan tombol mulai dan dapat masuk ke halaman artikel.	Aplikasi akan menampilkan halaman artikel.	✓	
Navigasi	*User* menekan beberapa tombol pada navigasi.	*User* dapat masuk ke halaman yang dituju dari navigasi aplikasi.	Aplikasi akan menampilkan halaman dituju.	✓	
Halaman Artikel	*User* melakukan gulir atau *scrolling* kebawah.	*User* dapat melihat daftar artikel.	Aplikasi akan menampilkan daftar artikel.	✓	
	User melakukan menekan pada *thumbnail* gambar atau judul artikel.	*User* dapat melihat isi artikel.	Aplikasi akan menampilkan keseluruhan isi artikel.	✓	
Halaman Deteksi	*User* mengarahkan kamera ke objek	*User* dapat melihat hasil deteksi *realtime*.	Aplikasi akan menampilkan hasil deteksi *realtinie*.	✓	

	yang akan dideteksi.				
	User menekan tombol 'Galeri' pada halaman deteksi.	*User* dapat memilih gambar dari galeri untuk dideteksi.	Aplikasi akan menampilkan galeri dan akan menampilkan hasil deteksi ketika *user* telah memilih gambar.	✓	
Halaman Tentang	*User* menekan tombol 'Tentang' pada navigasi aplikasi.	*User* dapat melihat isi halaman tentang aplikasi.	Aplikasi akan menampilkan halaman tentang aplikasi.	✓	

Saran dari penguji:

.....

.....

.....

.....

Semarang,

.....

NIDN 0601088201

Kuesioner Pengujian *Black Box* pada Aplikasi Deteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Algoritma *You Only Look Once* (YOLO) v8 Berbasis Android

Nama Penguji : Nur Latifah Dwi MS, M.kom.
Tanggal Pengujian : 27 Mei 2024

Nama Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Diterima	Ditolak
Splash Screen	*User* menekan tombol mulai.	*User* dapat menekan tombol mulai dan dapat masuk ke halaman artikel.	Aplikasi akan menampilkan halaman artikel.	✓	
Navigasi	*User* menekan beberapa tombol pada navigasi.	*User* dapat masuk ke halaman yang dituju dari navigasi aplikasi.	Aplikasi akan menampilkan halaman dituju.	✓	
Halaman Artikel	*User* melakukan gulir atau *scrolling* kebawah.	*User* dapat melihat daftar artikel.	Aplikasi akan menampilkan daftar artikel.	✓	
	User melakukan menekan pada *thumbnail* gambar atau judul artikel	*User* dapat melihat isi artikel.	Aplikasi akan menampilkan keseluruhan isi artikel.	✓	
Halaman Deteksi	*User* mengarahkan kamera ke objek	*User* dapat melihat hasil deteksi *realtime*.	Aplikasi akan menampilkan hasil deteksi *realtime*.	✓	

	yang akan dideteksi.				
	User menekan tombol 'Galeri' pada halaman deteksi.	*User* dapat memilih gambar dari galeri untuk dideteksi.	Aplikasi akan menampilkan galeri dan akan menampilkan hasil deteksi ketika *user* telah memilih gambar.	✓	
Halaman Tentang	*User* menekan tombol 'Tentang' pada navigasi aplikasi.	*User* dapat melihat isi halaman tentang aplikasi.	Aplikasi akan menampilkan halaman tentang aplikasi.	✓	

Saran dari penguji:

Coba cari contoh dalam satu daun mempunyai lebih dr 2 penyakit?

Semarang, 27 Mei 2024.

Nurdiafifah Dwi, MT, M.Kom

NIDN 0623039001

Kuesioner Pengujian *Black Box* pada Aplikasi Deteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Algoritma *You Only Look Once* (YOLO) v8 Berbasis Android

Nama Penguji : Ramadhan Renaldy, S.Kom, M.Kom
Tanggal Pengujian : 29 mei 2024

Nama Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil yang Didapatkan	Keterangan	
				Diterima	Ditolak
Splash Screen	*User* menekan tombol mulai.	*User* dapat menekan tombol mulai dan dapat masuk ke halaman artikel.	Aplikasi akan menampilkan halaman artikel.	✓	
Navigasi	*User* menekan beberapa tombol pada navigasi.	*User* dapat masuk ke halaman yang dituju dari navigasi aplikasi.	Aplikasi akan menampilkan halaman dituju.	✓	
Halaman Artikel	*User* melakukan gulir atau *scrolling* kebawah.	*User* dapat melihat daftar artikel.	Aplikasi akan menampilkan daftar artikel.	✓	
	User melakukan menekan pada *thumbnail* gambar atau judul artikel.	*User* dapat melihat isi artikel.	Aplikasi akan menampilkan keseluruhan isi artikel.	✓	
Halaman Deteksi	*User* mengarahkan kamera ke objek	*User* dapat melihat hasil deteksi *realtime*.	Aplikasi akan menampilkan hasil deteksi *realtime*.	✓	

	yang akan dideteksi.				
	User menekan tombol 'Galeri' pada halaman deteksi.	*User* dapat memilih gambar dari galeri untuk dideteksi.	Aplikasi akan menampilkan galeri dan akan menampilkan hasil deteksi ketika *user* telah memilih gambar.	✓	
Halaman Tentang	*User* menekan tombol 'Tentang' pada navigasi aplikasi.	*User* dapat melihat isi halaman tentang aplikasi.	Aplikasi akan menampilkan halaman tentang aplikasi.	✓	

Saran dari penguji:

Tambahkan menu untuk cara penakaian aplikasi untuk memudahkan pengguna baru

Semarang, 27 mei 2024

Ramadhan Renaldy, S.kom. M.kom

~~NIDN~~ NPP 29990167

Lampiran 3 Pengujian UAT

Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi Deteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Algoritma *You Only Look Once* (YOLO) v8 Berbasis Android

Nama Penguji : Ananta Agil Nusantara
Tanggal Pengujian : 27 mei 2024

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Aspek Kegunaan						
1.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong dapat bermanfaat bagi pengguna, khususnya petani?					✓
2.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memberikan informasi tentang penyakit pada daun tanaman singkong dan cara mengatasinya?					✓
Aspek Kemudahan Pengguna						
3.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong mudah dioperasikan?					✓
4.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai yang diharapkan?					✓
5.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong berisi informasi yang dibutuhkan?					✓

6.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai keperluan Anda?			✓		
Aspek *User Interface* (UI)						
7.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tampilan yang mudah dipahami?				✓	
8.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tampilan yang menarik?				✓	
9.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tema warna yang enak dilihat?			✓		
10.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong perlu dikembangkan lagi?					✓

Keterangan:

1 = Tidak Setuju

2 = Kurang Setuju

3 = Cukup Setuju

4 = Setuju

5 = Sangat Setuju

Semarang, 27 mei 2024

Aranta Agie Nusantara

Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi Deteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Algoritma *You Only Look Once* (YOLO) v8 Berbasis Android

Nama Penguji : Tri Setyobudi

Tanggal Pengujian : 27 mei 2024

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Aspek Kegunaan						
1.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong dapat bermanfaat bagi pengguna, khususnya petani?				✓	
2.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memberikan informasi tentang penyakit pada daun tanaman singkong dan cara mengatasinya?				✓	
Aspek Kemudahan Pengguna						
3.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong mudah dioperasikan?					✓
4.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai yang diharapkan?			✓		
5.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong berisi informasi yang dibutuhkan?				✓	

6.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai keperluan Anda?				✓	
Aspek *User Interface* (UI)						
7.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tampilan yang mudah dipahami?				✓	
8.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tampilan yang menarik?				✓	
9.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tema warna yang enak dilihat?			✓		
10.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong perlu dikembangkan lagi?					✓

Keterangan:

1 = Tidak Setuju

2 = Kurang Setuju

3 = Cukup Setuju

4 = Setuju

5 = Sangat Setuju

Semarang, 27 Mei 2024

Tri [illegible]

Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi Deteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Algoritma *You Only Look Once* (YOLO) v8 Berbasis Android

Nama Penguji : Ahmad Nur Khotis
Tanggal Pengujian : 27 Mei 2024

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Aspek Kegunaan						
1.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong dapat bermanfaat bagi pengguna, khususnya petani?				✓	
2.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memberikan informasi tentang penyakit pada daun tanaman singkong dan cara mengatasinya?				✓	
Aspek Kemudahan Pengguna						
3.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong mudah dioperasikan?				✓	
4.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai yang diharapkan?				✓	
5.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong berisi informasi yang dibutuhkan?			✓		

6.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai keperluan Anda?				✓	
Aspek *User Interface* (UI)						
7.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tampilan yang mudah dipahami?					✓
8.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tampilan yang menarik?				✓	
9.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tema warna yang enak dilihat?				✓	
10.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong perlu dikembangkan lagi?					✓

Keterangan:

1 = Tidak Setuju

2 = Kurang Setuju

3 = Cukup Setuju

4 = Setuju

5 = Sangat Setuju

Semarang, 27 Mei 2024

Ahmad Mukhlas

Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi Deteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Algoritma *You Only Look Once* (YOLO) v8 Berbasis Android

Nama Penguji : Tariono

Tanggal Pengujian : 2 Juni 2024

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Aspek Kegunaan						
1.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong dapat bermanfaat bagi pengguna, khususnya petani?					✓
2.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memberikan informasi tentang penyakit pada daun tanaman singkong dan cara mengatasinya?					✓
Aspek Kemudahan Pengguna						
3.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong mudah dioperasikan?					✓
4.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai yang diharapkan?			✓		
5.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong berisi informasi yang dibutuhkan?					✓

6.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai keperluan Anda?					✓
Aspek *User Interface* (UI)						
7.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tampilan yang mudah dipahami?				✓	
8.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tampilan yang menarik?				✓	
9.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tema warna yang enak dilihat?				✓	
10.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong perlu dikembangkan lagi?					✓

Keterangan:

1 = Tidak Setuju

2 = Kurang Setuju

3 = Cukup Setuju

4 = Setuju

5 = Sangat Setuju

Pemalang, 2 Juni 2024

Tarono

Kuesioner Pengujian *User Acceptance Testing* (UAT) Pada Aplikasi Deteksi Penyakit Pada Daun Tanaman Singkong Menggunakan Algoritma *You Only Look Once* (YOLO) v8 Berbasis Android

Nama Penguji : Ali Sodikin

Tanggal Pengujian : 2 Juni 2024

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Aspek Kegunaan						
1.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong dapat bermanfaat bagi pengguna, khususnya petani?					✓
2.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memberikan informasi tentang penyakit pada daun tanaman singkong dan cara mengatasinya?					✓
Aspek Kemudahan Pengguna						
3.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong mudah dioperasikan?					✓
4.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai yang diharapkan?				✓	
5.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong berisi informasi yang dibutuhkan?				✓	

6.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong sesuai keperluan Anda?					✓
Aspek *User Interface* (UI)						
7.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tampilan yang mudah dipahami?					✓
8.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tampilan yang menarik?					✓
9.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong memiliki tema warna yang enak dilihat?				✓	
10.	Apakah aplikasi deteksi penyakit pada daun tanaman singkong perlu dikembangkan lagi?					✓

Keterangan:

1 = Tidak Setuju

2 = Kurang Setuju

3 = Cukup Setuju

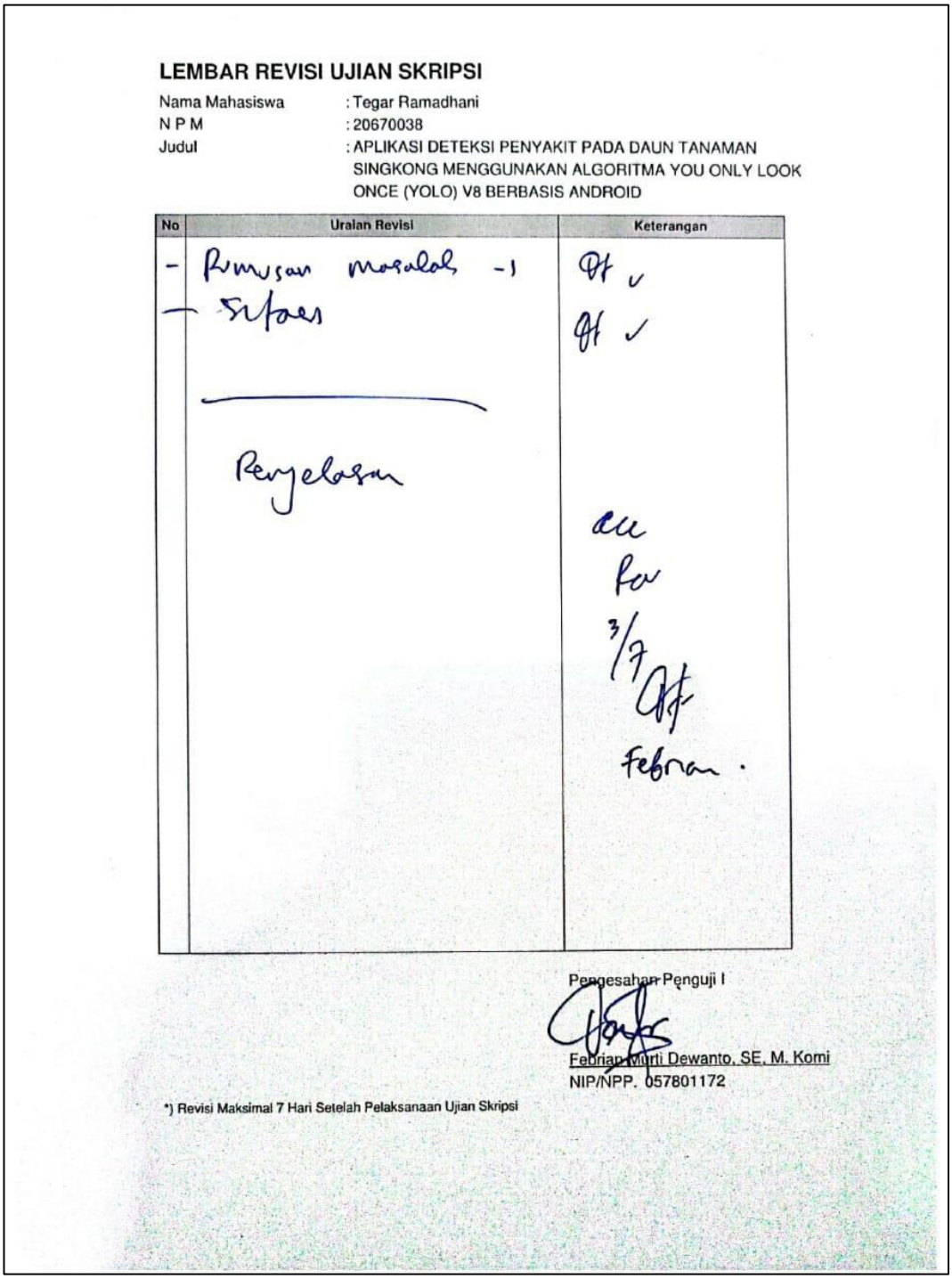
4 = Setuju

5 = Sangat Setuju

Pemalang, 2 Juni 2024

Ali Sobirin.

Lampiran 4 Lembar Revisi Ujian Skripsi



LEMBAR REVISI UJIAN SKRIPSI

Nama Mahasiswa : Tegar Ramadhani
N P M : 20670038
Judul : APLIKASI DETEKSI PENYAKIT PADA DAUN TANAMAN SINGKONG MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE (YOLO) V8 BERBASIS ANDROID

No	Uraian Revisi	Keterangan
-	Rumusan masalah -1	✓
-	Sistem	✓
	Penjelasan	acc Fav 3/7 Febrian.

Pengesahan Penguji I

Febrian Murti Dewanto, SE. M. Komi
NIP/NPP. 057801172

*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi

LEMBAR REVISI UJIAN SKRIPSI/TUGAS AKHIR

Nama Mahasiswa : Tegar Ramadhani
N P M : 20670038
Judul : APLIKASI DETEKSI PENYAKIT PADA DAUN TANAMAN SINGKONG MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE (YOLO) V8 BERBASIS ANDROID

No	Uraian Revisi	Keterangan
1.	Penomoran halaman.	
2.	Kasat Eksperimen untuk setiap parameter epoch 150?	

Pengesahan Penguji II

Aris Tri Joko Harjanto S.Kom., M.Kom
NIP/NPP. 148201443

*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi

LEMBAR REVISI UJIAN SKRIPSI/TUGAS AKHIR

Nama Mahasiswa : Tegar Ramadhani
N P M : 20670038
Judul : APLIKASI DETEKSI PENYAKIT PADA DAUN TANAMAN SINGKONG MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE (YOLO) V8 BERBASIS ANDROID

No	Uraian Revisi	Keterangan
1	Penambahan tambahkan dataset uji coba / analisi	ace 3/7/2024
2	Cek Landasan teori	
3	Pada kerangka berfikir tahap tranyatan dll belum tepat.	
4	Use Case Diagram	
5	Form pengujian perlakan dalam BAB III	
6	Gambar : berikan border template HP / Tab.	

Pengesahan Penguji III

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

*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi