

Rancang Bangun Aplikasi *Mobile* Berbasis YOLOv11 untuk Deteksi Penyakit Daun Tanaman Anggur

Au Izaldi Fachril Rahmadani¹, Salamun Rohman Nudin²

D4 Manajemen Informatika, Universitas Negeri Surabaya
Jl. Ketintang, Ketintang, Kec. Gayungan, Surabaya, Jawa Timur 60231

¹auizaldi.21026@mhs.unesa.ac.id

²salamunrohman@unesa.ac.id

Abstrak— Tanaman anggur rentan terhadap berbagai penyakit daun yang dapat menurunkan produktivitas, sehingga diperlukan metode deteksi yang cepat dan akurat untuk menggantikan identifikasi manual yang kurang konsisten. Penelitian ini bertujuan untuk mengembangkan sistem deteksi penyakit daun anggur dalam bentuk aplikasi seluler menggunakan model YOLOv11 sebagai object detector. Metode penelitian mencakup pengumpulan data, pelatihan model, evaluasi model, dan ekspor model. Dataset yang digunakan terdiri dari empat kelas penyakit dan satu kelas untuk daun sehat, dengan pembagian ke dalam tiga skema proporsi data, yaitu 70:20:10, 80:10:10, dan 60:20:20. Setiap skema, data dipisahkan kedalam folder train set, validation set, dan test set. Model kemudian dilatih menggunakan empat optimizer yang berbeda yakni Adam, AdamW, NAdam, dan RAdam, untuk mengetahui konfigurasi terbaik dalam meningkatkan performa deteksi. Hasil pelatihan menunjukkan bahwa skema 80:10:10 dengan optimizer NAdam memberikan performa paling unggul, dengan F1-Score 98,93%, precision 99,81%, recall 98,13%, dan mAP50-95 sebesar 91,63%. Penelitian ini tidak hanya berhasil membangun model tetapi juga mengimplementasikannya ke dalam aplikasi mobile yang praktis dan kokoh dikondisi nyata.

Kata Kunci— YOLOv11, Deteksi Penyakit Daun, Tanaman Anggur, Optimizer, Aplikasi Mobile.

Abstract— Grapewine plants are vulnerable to various leaf diseases that can reduce productivity, making fast and accurate detection methods necessary to replace inconsistent manual identification. This study aims to develop a mobile-based grape leaf disease detection system using the YOLOv11 model as an object detector. The research method includes data collection, model training, model evaluation, and model export. The dataset consists of four disease classes and one healthy leaf class, divided into three data split schemes: 70:20:10, 80:10:10, and 60:20:20. For each scheme, the data were separated into training, validation, and test sets. The model was trained using four different optimizers, namely Adam, AdamW, NAdam, and RAdam, to determine the best configuration for improving detection performance. The experimental results show that the 80:10:10 data split with the NAdam optimizer achieved the best performance, with an F1-Score of 98.93%, precision of 99.81%, recall of 98.13%, and mAP50-95 of 91.63%. This study successfully developed and implemented the model into a practical and robust mobile application for real-world conditions.

Keywords — YOLOv11, Leaf Disease Detection, Grapewine, Optimizer, Mobile Application.

I. PENDAHULUAN

Tanaman anggur merupakan salah satu komoditas hortikultura yang populer baik di negara maju maupun berkembang. Permintaan buah anggur di Indonesia cenderung fluktuatif. Dikutip dari [1], menurut data kementerian perdagangan tahun 2020 menunjukkan 63.221 ton anggur diimpor sepanjang tahun 2016, meningkat hingga 115.717 ton pada 2019, kemudian menurun menjadi 62.226 ton pada 2020, salah satu akibat pandemi. Anggur memiliki nilai ekonomi dan kesehatan yang tinggi, karena kandungan bioaktif seperti asam fenolik, flavonoid, antosianin, stilbena, dan lipid yang dapat menurunkan risiko penyakit kardiovaskular termasuk hipertensi [2]. Selain dikonsumsi sebagai buah segar, anggur juga dimanfaatkan sebagai ahan baku kosmetik dengan manfaat antioksidan, antimikroba, antiinflamasi, proteksi terhadap sinar UVA, hingga efek pencerahan kulit [3].

Seperi tanaman lain, anggur rentan terhadap berbagai penyakit yang dapat mengganggu kesehatan dan produktivitas tanaman. Pengaruh penyakit tersebut seperti gugurnya daun sehingga menyebabkan terhambatnya proses pematangan buah, hal ini diakibatkan oleh bakteri plasmopara viticola atau yang sering disebut penyakit bulu berbulu halus (downy mildew) [4]. Upaya pengendalian dapat dilakukan dengan produk seperti biostimulan dan lime sulfur, yang terbukti efektif dalam menekan downy mildew maupun leaf spot pada tanaman anggur *vitis labrusca* [5]. Tindakan deteksi dini penyakit diperlukan untuk mencegah penyebaran lebih lanjut dan menjaga keberlanjutan rantai pasok.

Perkembangan teknologi kecerdasan buatan membuka peluang besar dalam mendukung kemajuan dunia pertanian, khususnya melalui pemanfaatan computer vision untuk diagnosis penyakit tanaman. Computer vision memungkinkan komputer menafsirkan gambar layaknya indera penglihatan manusia dan telah digunakan dalam berbagai aplikasi seperti analisis kualitas benih, manajemen irigasi, analisis kesehatan tanaman, hingga estimasi hasil panen. Hasil tinjauan literatur menunjukkan

bahwa *convolutional neural network* (CNN) merupakan fondasi utama dalam pengembangan metode modern *computer vision* dibidang pertanian [6].

Peninjauan terhadap studi terdahulu yang telah menggunakan model YOLO dalam deteksi penyakit daun tanaman anggur diperlukan untuk mengidentifikasi pendekatan yang sudah ada serta melihat celah penelitian yang masih belum terjawab.

Beberapa studi telah memanfaatkan arsitektur YOLO untuk deteksi penyakit daun anggur. Diantaranya, penelitian [7] mengusulkan model YOLOv8-ACCW untuk mendeteksi penyakit black measles, black rot, blight fungus, dan satu kelas daun sehat (healthy). Model YOLOv8-ACCW meraih F1-Score 92,40%, mAP%50 92,8%, dan mAP50-95 73,8%. Penelitian [8] mengusulkan GFCD-YOLOXS yaitu model YOLOXS yang ditingkatkan untuk deteksi penyakit anggur dengan menambahkan modul FOCUS pada *backbone* untuk meningkatkan ekstraksi fitur, kemudian pada prediction head dilengkapi dengan CBAM (Convolutional Block Attention Module) agar model fokus pada fitur penting penyakit juga penambahan *double residual edge* untuk menjaga performa deep network dan memanfaatkan fitur non-kunci. Model ini memperoleh metrik mAP sebesar 99,10%.

Penelitian [9] memperkenalkan YOLOv5-CA sebuah model berbasis YOLOv5 yang dimodifikasi dengan integrasi mekanisme coordinate attention (CA) untuk menyoroti fitur visual terkait penyakit downy mildew untuk meningkatkan kinerja deteksi. Model ini meraih metrik performa F1-Score: 84,64%, precision: 85,59%, recall: 83,70%, mAP50:89,50%, dan FPS: 58,82. Selain penelitian yang berfokus pada peningkatan arsitektur model, terdapat pula penelitian terdahulu yang menekankan pada penerapan model pada platform tertentu.

Penelitian [10] menggunakan model YOLOv8 untuk mendeteksi penyakit black rot, black measles, blight fungus dan satu kelas daun sehat (healthy). Model ini diimplementasikan menjadi aplikasi mobile akan tetapi hasil deteksi pada aplikasi ini tidak menyertakan confidence score dan penanganan penyakit. Model ini meraih metrik performa precision: 99.9%, recall:100%, mAP: 99.5%, dan mAP50-95:88%.

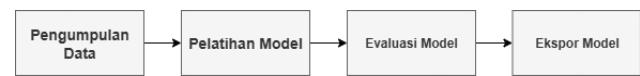
Penelitian [11] memperkenalkan model GA-YOLO yang merupakan hasil peningkatan dari YOLOv4 meliputi desain backbone baru yakni SE-CSPGhostnet yang ringan dan efisien, penerapan mekanisme *adaptive spatial feature fusion* untuk meningkatkan kemampuan deteksi pada objek padat dan tertutup, serta penggunaan *loss function* baru untuk mengoptimalkan efisiensi dan akurasi deteksi secara keseluruhan. Model ini meraih metrik performa F1-Score: 94,78%, precision: 95,33%, recall: 94,22%, mAP:96,87%, dan FPS: 20,245. Model ini digunakan untuk mendeteksi tanda buah anggur dan diimplementasikan menjadi aplikasi desktop.

Penelitian [12] menggunakan model YOLOv8m untuk mendeteksi daun sehat dan daun berpenyakit yang diimplementasikan menjadi aplikasi website. Model ini

menghasilkan metrik mAP:86,8%, precision:73,6%, recall: 80,1%, dan F1-Score: 76,7%.

Penelitian ini berfokus pada pengembangan aplikasi mobile bernama VineCare. Aplikasi ini mampu mendeteksi empat macam penyakit yaitu *black rot*, *esca*, *leaf blight*, dan *downy mildew* serta dapat mendeteksi daun yang sehat. Selain itu, aplikasi ini juga menampilkan pencegahan dan penanganan pada tiap-tiap penyakit. Dalam tahap pengembangan model, dilakukan pelatihan menggunakan beberapa skema split dataset serta penerapan beberapa algoritma optimizer untuk memperoleh konfigurasi model dengan performa terbaik. Pendekatan ini diharapkan dapat meningkatkan akurasi deteksi dan memberikan solusi praktis bagi petani dalam melakukan pencegahan dini terhadap penyebaran penyakit daun anggur.

II. METODE PENELITIAN



Gambar 1 Alur penelitian

Metode yang digunakan dalam penelitian ini terdiri atas beberapa tahapan yang dirancang secara sistematis untuk menghasilkan model deteksi penyakit daun anggur yang optimal. Tahapan utama dalam penelitian ini meliputi pengumpulan data, pelatihan model, evaluasi model, dan eksport.

A. Pengumpulan Data

Dataset yang digunakan dalam penelitian ini diperoleh dari platform Roboflow oleh akun GIORGIO (<https://universe.roboflow.com/giorgio-m6t5s/vineyards-leaf-disease-detection>), berisi citra daun anggur yang telah dianotasi dalam lima kelas: *healthy* (daun sehat), *black rot*, *esca* (*black measles*), *downy mildew*, dan *leaf blight*. Sebelum proses pelatihan, data melalui beberapa tahap pre-processing yang mencakup anotasi, pembagian dataset (*split*), *preprocessing*, dan augmentasi.

Tahap anotasi dilakukan untuk memberi label pada citra berupa bounding box dan kelas kategori. Selanjutnya, dataset dibagi menjadi tiga direktori terpisah yakni, train, validation, dan testing dengan beberapa rasio, yaitu 70:20:10, 80:10:10, dan 60:20:20, guna mengevaluasi pengaruh pembagian data terhadap akurasi dan generalisasi model. Tahap *preprocessing* meliputi penyesuaian format dan *resize* citra agar sesuai dengan kebutuhan arsitektur YOLOv11 yang memerlukan dimensi *input* seragam.

Terakhir, augmentasi data diterapkan melalui rotasi, perubahan kecerahan, dan skala acak untuk meningkatkan variasi data serta mengurangi risiko overfitting. Seluruh proses ini bertujuan untuk memastikan dataset berada dalam kondisi optimal sehingga mendukung pelatihan model deteksi penyakit daun tanaman anggur secara efektif.

B. Pelatihan Model

Pelatihan model YOLOv11 pada penelitian ini dilakukan menggunakan Google Colab dengan dukungan GPU NVIDIA Tesla T4 berkapasitas 16 GB untuk mempercepat proses komputasi. Pelatihan dijalankan selama 40 epochs dengan ukuran batch 16, guna menyeimbangkan efisiensi memori dan kecepatan pelatihan. Seluruh citra pada dataset diubah ukurannya menjadi 640 x 640 piksel.

Beberapa optimizer yang digunakan meliputi Adam, AdamW, NAdam, dan RAdam untuk membandingkan performa konvergensi model. Nilai learning rate awal (lr_0) ditetapkan sebesar 0,001, sedangkan learning rate akhir (lr_f) sebesar 0,01. Parameter momentum bernilai 0,937 untuk menjadi stabilitas pembaruan bobot, dan weight decay sebesar 0,0005 diterapkan sebagai regularisasi guna mencegah *overfitting*. Fase *warm-up* berlangsung selama tiga *epochs* pertama untuk menstabilkan proses pelatihan. Bobot pada fungsi *loss* diatur seimbang, dengan $box_loss = 0,05$, $class\ loss = 0,5$, dan distribution focal loss = 1.

TABEL 1 HYPERPARAMETER

Hyperparameter	Optimizer			
	Adam	AdamW	NAdam	RAdam
epochs	40	40	40	40
imgsz	640x640	640x640	640x640	640x640
lr0	0,001	0,0025	0,001	0,001
lrf	0,01	0,01	0,01	0,01
momentum	0,937	0,937	0,937	0,937
weight_decay	0,0005	0,0005	0,0005	0,0005
warmup_epochs	3	3	3	3
warmup_momentum	0,8	0,8	0,8	0,8
warmup_bias_lr	0,1	0,1	0,1	0,1
batch	16	16	16	16
box	0,05	0,05	0,05	0,05
cls	0,5	0,5	0,5	0,5
dfl	1,0	1,0	1,0	1,0

Tahapan pelatihan model pada penelitian ini dilakukan secara bertahap untuk memastikan proses berjalan optimal dan hasil model memiliki performa yang baik. Secara umum, proses pelatihan terdiri atas lima tahap utama, yaitu instalasi *package*, pengunduhan *dataset*, pelatihan (*training*), validasi (*validation*), dan ekspor model (*export*).

1) Instalasi Package

Tahap awal dilakukan dengan menginstal seluruh dependensi dan *package* yang dibutuhkan. *Package* roboflow digunakan untuk mengunduh *dataset* menggunakan API, ultralytics digunakan untuk keperluan *training*, *validation*, dan *export*, setelahnya seperti onnxruntime-gpu, onnx-graphsurgeon, onnx, sng4onnx, onnx2tf, onnxslim, ai-edge-litert, dan tensorflow digunakan untuk proses ekspor model ke tensorflow lite.

2) Pengunduhan Dataset

Dataset diperoleh dari platform roboflow dengan memanfaatkan API *key* yang disediakan. Proses ini memungkinkan pengunduhan dataset secara otomatis ke dalam environment Google Colab. Setelah kode dijalankan maka akan muncul direktori sesuai dengan nama dataset dan di dalamnya terdapat direktori *train*, *val*, dan *test*. Selain itu, terdapat file konfigurasi (*data.yaml*) yang berisi path dari gambar masing-masing tiga folder (*train*, *val*, dan *test*) ubah path tersebut sesuai dengan *path* masing-masing folder *images* di setiap tiga folder tersebut.

3) Pelatihan (training)

Tahap pelatihan dilakukan menggunakan framework Ultralytics YOLO dengan memanfaatkan model dasar *yolov11n.pt*. Proses pelatihan dimulai dengan memanggil file konfigurasi dataset (*data.yaml*) yang berisi informasi kelas dan direktori data pelatihan, validasi, serta pengujian. Pelatihan dilakukan selama 40 epochs dengan ukuran gambar yang diresize menjadi 640x640 piksel, menggunakan optimizer yang berbeda-beda (Adam, AdamW, NAdam, dan RAdam). Kemudian parameter selanjutnya seperti learning rate diisi sesuai dengan tabel 1.

C. Evaluasi Model

Tahap validasi dilakukan untuk mengevaluasi kinerja model hasil pelatihan menggunakan Ultralytics YOLO. Model terbaik yang diperoleh dari proses pelatihan dimuat dan diuji menggunakan dataset validasi yang sama seperti saat pelatihan. Pengujian dilakukan dengan parameter ukuran gambar 640x640 piksel, *batch size* 16, serta *confidence threshold* 0,25 dan *IoU threshold* 0,6, guna memastikan keseimbangan antara *confidence score* dan akurasi *bounding box*. Selama proses validasi, metrik evaluasi seperti F1-Score, precision, recall, dan mAP (*mean average precision*) dihitung untuk menilai performa model dalam mengenali setiap kelas. Selain itu, nilai mAP50, mAP75, dan mAP50-59 juga digunakan untuk mengukur akurasi model pada berbagai tingkat ambang IoU. Hasil evaluasi ini memberikan gambaran menyeluruh terhadap kemampuan model YOLOv11 dalam mendekripsi dan mengklasifikasikan penyakit daun anggur secara akurat.

D. Ekspor Model

Tahap ekspor dilakukan untuk mengonversi model hasil pelatihan ke dalam format tensorflow lite. Model yang dipilih untuk diekspor adalah model dari hasil pelatihan yang memiliki metrik terbaik pada tahap validasi. Model ini nantinya akan diekspor dengan menggunakan fungsi *export()* dari library Ultralytics YOLO. Pada proses ini, model dikonversi ke format tensorflow lite (*tflite*) dengan aktivasi *quantization int8* agar ukuran model menjadi lebih kecil dan efisien saat dijalankan di perangkat dengan sumber daya terbatas seperti *smartphone*. Selain itu, parameter *device=0* menunjukkan bahwa proses konversi dilakukan menggunakan GPU untuk mempercepat eksekusi. Hasil

dari tahap ini adalah model YOLOv11 dalam format tflite yang siap diintegrasikan ke dalam aplikasi *mobile*.

III. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari proses pelatihan dan validasi model. Pembahasan meliputi analisis nilai loss, evaluasi metrik kinerja, serta perbandingan performa antar skema pelatihan yang digunakan. Selain itu, juga disajikan hasil pengujian aplikasi dengan skema tertentu.

A. Analisis Hasil Performance Metrics

TABEL 2 HASIL PELATIHAN SKEMA 1

Optimizer	Skema	Skema 3 - 60:20:20					
		mAP-50-95 (%)	mAP-75 (%)	mAP-50 (%)	mAP-50 (%)	Precision (%)	Recall (%)
Adam	98,36	99,86	97,05	98,59	98,56	90,20	
AdamW	98,39	99,82	97,14	98,41	98,41	90,42	
Nadam	98,62	99,93	97,47	98,60	98,60	90,95	
RAdam	98,59	99,95	97,41	98,62	98,57	90,48	

Pada skema 1 (70:20:10), seluruh optimizer menunjukkan performa tinggi dan konsistennya di berbagai metrik evaluasi. Nadam memperoleh **F1-Score** tertinggi sebesar **98,62%** dengan keseimbangan **precision 99,93%** dan **recall 97,47%**. RAdam unggul dalam precision dari seluruh *optimizer* dengan **99,95%** sementara Adam dan AdamW tetap kompetitif meski sedikit di bawah dalam hal recall dan mAP50-95. Secara keseluruhan, Nadam memberikan hasil paling seimbang dengan **mAP50-95** mencapai **90,95%**.

TABEL 3 HASIL PELATIHAN SKEMA 2

Optimizer	Skema	Skema 1 70:20:10					
		mAP-50-95 (%)	mAP-75 (%)	mAP-50 (%)	mAP-50 (%)	Precision (%)	Recall (%)
Adam	98,84	99,86	97,92	98,92	98,92	91,32	
AdamW	98,59	99,65	97,64	98,70	98,61	89,38	
Nadam	98,93	99,81	98,13	98,88	98,80	91,63	
RAdam	98,70	99,62	97,87	98,88	98,80	91,05	

Pada skema 2 (80:10:10), performa model meningkat terutama pada metrik F1-Score. Nadam menunjukkan hasil terbaik dengan **F1-Score 98,93%**, **recall tertinggi 98,13%**, dan **mAP50-95** sebesar **91,63%**, menandakan

generalisasi yang sangat baik. Adam juga kompetitif dengan **F1-Score 98,84%** dan **mAP50-95 91,32%**, sedangkan RAdam tetap stabil dengan nilai **F1-Score 98,70%**. AdamW menunjukkan performa terendah terutama pada **mAP50-95 (89,38%)**. Secara keseluruhan, Nadam menjadi *optimizer paling optimal* dengan konsistensi tinggi di seluruh metrik.

TABEL 4 HASIL PELATIHAN SKEMA 3

Optimizer	Skema	Skema 3 - 60:20:20					
		mAP-50-95 (%)	mAP-75 (%)	mAP-50 (%)	F1-Score (%)	Precision (%)	Recall (%)
Adam	98,84	99,93	97,86	98,90	98,80	90,96	
AdamW	98,72	99,82	97,74	98,78	98,78	90,16	
Nadam	98,92	99,85	98,07	98,72	98,72	90,59	
RAdam	98,79	99,30	98,30	98,96	98,86	90,66	

Pada skema 3 (60:20:20), performa model tetap tinggi meskipun data latih lebih sedikit. Nadam menghasilkan **F1-Score tertinggi 98,82%** dengan **precision 99,85%** dan **recall 98,07%**, menunjukkan keseimbangan yang baik antara akurasi dan sensitivitas. Adam menunjukkan performa unggul pada metrik **mAP40-95 sebesar 90,96%**. RAdam menampilkan kemampuan generalisasi yang baik dengan **recall 98,30%** dan **mAP50-95 sebesar 90,66%**, sedangkan AdamW memiliki performa sedikit lebih rendah namun tetap kompetitif. Secara keseluruhan Nadam menjadi *optimizer paling seimbang dan andal*.

B. Perbandingan dengan Penelitian Terdahulu

Bagian ini membahas perbandingan hasil penelitian dengan penelitian terdahulu yang berfokus pada deteksi penyakit tanaman menggunakan model berbasis YOLO. Tujuannya adalah untuk mengevaluasi sejauh mana peningkatan performa dan efisiensi yang dicapai dalam penelitian ini dibandingkan penelitian terdahulu.

TABEL 5 PERBANDINGAN PENELITIAN TERDAHULU

Model	Kelas deteksi	Hasil	Implementasi	Kelebihan/Keterbatasan
YOLO v8 [10]	(black measles, black rot, blight fungus, and healthy)	Precision: 99.9% Recall: 100 % mAP: 99.5% mAP50-95: 95.88%	Aplikasi mobile	Metrik evaluasi tinggi, tetapi hanya 4 kelas, aplikasi hanya menampilkan hasil deteksi penyakit

Model	Kelas deteksi	Hasil	Implementasi	Kelebihan/Keterbatasan
GA-YOLO [11]	Mendeteksi tandan buah anggur	F1-Score: 94,78% Precision: 95,33% Recall: 94,22% mAP: 96,87% FPS: 20,245	Aplikasi desktop	Implementasi desktop kurang fleksibel
YOLO v8-ACC W [7]	(black measles, black rot, blight fungus, dan healthy)	F1-Score: 92,40% mAP50: 92,8% mAP50-95: 73,8%	Hanya model	Fokus pada peningkatan performa model, tidak ada implementasi aplikasi
GFCD	15 penyakit tanaman	mAP: 99,10%	Hanya model	Kelas deteksi beragam, fokus pada peningkatan performa model, tidak ada implementasi aplikasi
YOLO XS [8]	Mendeteksi tanaman anggur tidak hanya daun	mAP: 99,10%	Hanya model	Implementasi berbasis website namun kurang praktis dilapangan dan jumlah kelas terbatas
YOLO v8m [12]	Mendeteksi daun berpenyakit dan daun sehat	mAP: 86,8% Precision: 73,6% Recall: 80,1% F1-Score: 76,7%	Website	Kelas deteksi khusus penyakit downy mildew, fokus pada peningkatan model, tidak ada implementasi aplikasi
YOLO v5-CA [9]	Downy mildew	F1-Score: 84,64% Precision: 85,59% Recall: 83,70% mAP50: 89,50% FPS: 58,82	Hanya model	Mendeteksi 5 kelas (Black measles, Black rot, Blight fungus, Esca, Health y)
YOLO v11n (skema terbaik)		F1-Score: 98,93% Precision: 99,81% Recall: 98,13% mAP50-95: 91,63%	Aplikasi mobile	Menampilkan hasil deteksi dan confidence score, menampilkan penanganan, implementasi ke aplikasi

Model	Kelas deteksi	Hasil	Implementasi	Kelebihan/Keterbatasan
				mobile lebih praktis dilapangan

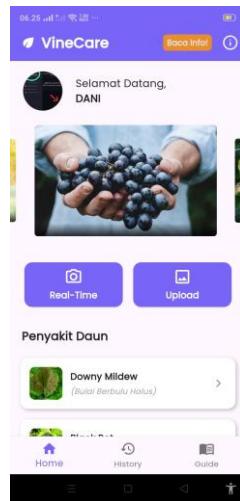
Tabel 5 menunjukkan hasil penelitian terdahulu dan penelitian ini. Skema 2 dengan optimizer NAdam menghasilkan F1-Score tertinggi sebesar 98,93%, berada sedikit di bawah penelitian [10] dengan 99,95%, namun lebih tinggi dibandingkan [12]. Nilai mAP50 terbaik dicapai oleh skema 3 dengan optimizer RAdam sebesar 98,96%, mendekati hasil [10] sebesar 99,5% dan [8] sebesar 99,1%, serta jauh melampaui penelitian [12] dan [9]. Sementara itu, nilai mAP50-95 tertinggi sebesar 91,63% juga menunjukkan keunggulan YOLOv11 dibandingkan [10] dan [7].

Dari sisi implementasi, penelitian ini unggul karena berhasil mengintegrasikan model ke dalam aplikasi mobile yang menampilkan akurasi tiap deteksi dan solusi penanganan penyakit secara langsung. Keunggulan lainnya adalah cakupan lima kelas deteksi (black rot, esca, leaf blight, downy mildew, dan daun sehat), lebih banyak dibandingkan sebagian besar penelitian [10], [7], [12], dan [9]. Namun, penelitian ini masih memiliki keterbatasan pada jumlah kelas penyakit yang dideteksi jika dibandingkan dengan penelitian [8].

C. Tampilan Antarmuka Aplikasi

Aplikasi ini menampilkan hasil deteksi secara real-time melalui kamera, serta menyediakan informasi detail mengenai kelas penyakit yang teridentifikasi. Gambar berikut menunjukkan tampilan antarmuka aplikasi pada proses pendekripsi, mulai dari halaman utama hingga hasil prediksi yang ditampilkan kepada pengguna.

1) Tampilan Menu Home



Gambar 2 Menu Home

Menu Home menampilkan foto profil pengguna beserta nama pengguna sebagai identitas akun yang sedang digunakan. Pada menu ini juga tersedia beberapa tombol utama, yaitu pilihan menu deteksi untuk melakukan proses pendekripsi penyakit, menu daftar penyakit yang berisi informasi jenis-jenis penyakit, serta menu informasi aplikasi yang menyediakan penjelasan terkait penggunaan dan fitur aplikasi.

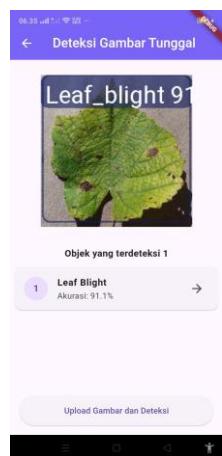
2) Tampilan Deteksi Real-Time



Gambar 3 Deteksi Real-Time

Menu deteksi real-time berfungsi untuk mendekripsi penyakit secara real-time. Terdapat tampilan objek yang terdeteksi, FPS, waktu deteksi, dan tombol potret. Hasil dari deteksi ini berupa bounding box, nama kelas, dan akurasi.

3) Tampilan Deteksi Unggah Gambar



Gambar 4 Deteksi Unggah Gambar

Menu Deteksi Unggah Gambar digunakan untuk melakukan proses deteksi penyakit tanaman melalui gambar yang sudah tersedia di galeri gawai pengguna, sehingga pengguna tidak perlu mengambil gambar secara langsung menggunakan kamera.

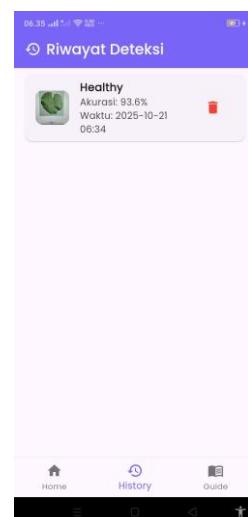
4) Tampilan Hasil Deteksi



Gambar 5 Hasil Deteksi

Tampilan ini menampilkan hasil deteksi seperti gambar deteksi, nama kelas, akurasi, penyebab, gejala, pencegahan, dan tombol simpan untuk menyimpan hasil deteksi.

5) Tampilan Menu History



Gambar 6 Menu History

Menu ini menampilkan seluruh hasil deteksi yang pernah disimpan oleh pengguna sebagai riwayat, serta dilengkapi dengan tombol hapus yang berfungsi untuk menghapus hasil deteksi tertentu sehingga data dapat dikelola dengan lebih rapi dan sesuai kebutuhan pengguna.

6) Tampilan Menu Guide



Gambar 7 Menu Guide

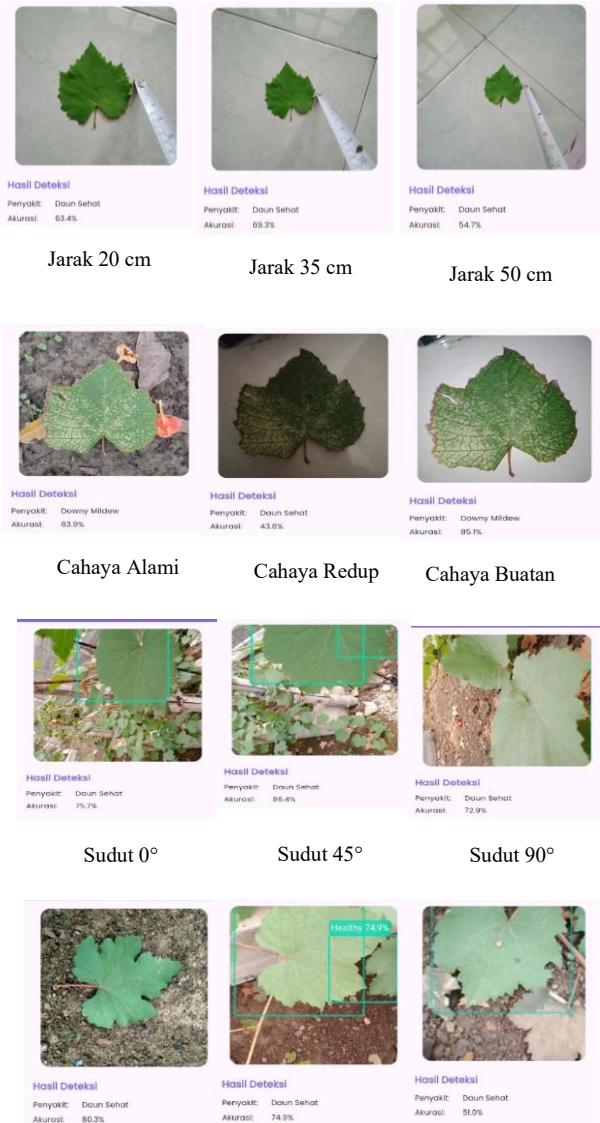
Menu ini juga menyediakan petunjuk penggunaan aplikasi beserta tips dan saran yang membantu pengguna dalam memaksimalkan fitur aplikasi agar hasil deteksi yang diperoleh lebih optimal.

D. Skema Ujicoba Deteksi

Ujicoba ini melibatkan empat skenario utama yakni, jarak pengambilan gambar, kondisi pencahayaan, sudut kamera, dan latar belakang. Setiap kondisi diuji untuk memastikan bahwa aplikasi mampu mendeteksi objek daun anggur secara optimal pada berbagai situasi yang mungkin terjadi di lapangan.

TABEL 6 SKEMA PENGUJIAN DETEKSI

No.	Skema Pengujian	Kondisi Uji	Akurasi (%)
1.	Jarak	20 cm	63,4
		35 cm	69,3
		50 cm	54,7
2.	Pencahayaan	Cahaya terang alami (outdoor)	83,9
		Cahaya redup	43,8
		Cahaya buatan	85,1
3.	Sudut Kamera	0° (tegak lurus)	86,4
		45° (miring)	75,7
		90° (samping)	72,9
4.	Latar Belakang	Polos	80,3
		Alami	74,9
		Kompleks	51,0



Gambar 8 Hasil Uji Coba Skema Deteksi

Hasil pengujian menunjukkan bahwa faktor jarak, pencahayaan, sudut kamera, dan latar belakang memiliki pengaruh signifikan terhadap akurasi deteksi. Pada skema jarak, akurasi terbaik diperoleh pada jarak 35 cm dengan nilai 69,3% karena objek berada pada posisi ideal sehingga fitur terlihat jelas, sedangkan jarak terlalu dekat (20 cm, 63,4%) dan terlalu jauh (50 cm, 54,7%) menurunkan ketajaman fitur yang tertangkap kamera. Pada pengujian pencahayaan, cahaya buatan (flash kamera) memberikan akurasi tertinggi sebesar 85,1% diikuti dengan cahaya terang alami sebesar 83,9%, karena kedua kondisi menyediakan intensitas cahaya yang cukup untuk menampilkan detail objek. Sebaliknya, cahaya redup menghasilkan akurasi terendah sebesar 43,8% akibat rendahnya kualitas pencitraan. Pada skema sudut kamera, sudut tegak lurus (0°) memberikan akurasi tertinggi yaitu

86,4% karena seluruh fitur objek tampak jelas, sementara 45° dan 90° menurunkan akurasi masing-masing menjadi 75,7% dan 72,9% akibat berkurangnya area fitur yang terdeteksi. Pada pengujian latar belakang, latar polos menghasilkan akurasi tertinggi (80,3%) karena minim gangguan visual, sedangkan latar alami (74,9%) dan latar kompleks (51,0%) menurunkan akurasu akibat meningkatnya gangguan visual yang mengganggu proses identifikasi objek.

IV. KESIMPULAN

Penelitian ini berhasil mengembangkan model deteksi penyakit daun tanaman anggur berbasis YOLOv11 melalui tahapan yang sistematis, meliputi pengunduhan dataset menggunakan API, penyesuaian konfigurasi data, pelatihan dengan pengaturan hyperparameter dan augmentasi, validasi performa, serta ekspor model ke format TensorFlow Lite (.tflite) untuk implementasi pada perangkat mobile. Hasil evaluasi menunjukkan bahwa model memiliki performa yang baik, dengan konfigurasi terbaik pada skema pembagian data 80:10:10 dan optimizer NAdam yang menghasilkan F1-Score sebesar 98,93% dan mAP50-95 sebesar 91,63%. Model berhasil diintegrasikan ke dalam aplikasi *mobile* dengan fitur deteksi *real-time*, unggah gambar, dan penyimpanan hasil, serta mampu beroperasi secara stabil dan memberikan informasi yang akurat kepada pengguna.

Penelitian selanjutnya disarankan untuk menggunakan dataset yang lebih besar dan beragam guna meningkatkan ketahanan model terhadap variasi kondisi lapangan serta mengurangi potensi ketidakseimbangan data antar kelas. Pengembangan lanjutan juga dapat mencakup optimalisasi arsitektur model dan pemanfaatan algoritma optimasi lain untuk meningkatkan efisiensi. Dari sisi aplikasi, disarankan penambahan fitur notifikasi, dashboard monitoring, dan konsultasi dengan pakar. Selain itu, perlu dikembangkan mekanisme pelaporan otomatis untuk kasus daun yang tidak terdeteksi, sehingga data tersebut dapat dikumpulkan dan dimanfaatkan dalam pembaruan model dan pengembangan aplikasi secara berkelanjutan.

REFERENSI

- [1] [1] R. A. R. Dian Nurul Hikmah, Effendy, “Faktor-Faktor Yang Memengaruhi Permintaan Buah,” *e-J. Agrotekbis* 5 579–585, Oktober 2017, vol. 5, no. 5, hal. 579–585, 2024, [Daring]. Tersedia pada: file:///C:/Users/Galih/Downloads/1382-Article Text-3223-1-10-20220729.pdf
- [2] [2] A. Sabra, T. Netticadan, dan C. Wijekoon, “Grape bioactive molecules, and the potential health benefits in reducing the risk of heart diseases,” *Food Chem. X*, vol. 12, hal. 100149, 2021, doi: 10.1016/j.fochx.2021.100149.
- [3] [3] M. L. Castro, J. P. Ferreira, M. Pintado, O. L. Ramos, S. Borges, dan S. Baptista-Silva, “Grape By-Products in Sustainable Cosmetics: Nanoencapsulation and Market Trends,” *Appl. Sci.*, vol. 13, no. 16, 2023, doi: 10.3390/app13169168.
- [4] [4] O. Viret, *Science of Fungi in Grapevine*.
- [5] [5] G. M. Ferreira, R. R. Moreira, T. M. Jarek, C. N. Nesi, L. A. Biasi, dan L. L. May De Mio, “Alternative control of downy mildew and grapevine leaf spot on *Vitis labrusca*,” *Australas. Plant Pathol.*, vol. 51, no. 2, hal. 193–201, 2022, doi: 10.1007/s13313-021-00836-7.
- [6] [6] V. G. Dhanya dkk., “Deep learning based computer vision approaches for smart agricultural applications,” *Artif. Intell. Agric.*, vol. 6, hal. 211–229, 2022, doi: 10.1016/j.aiia.2022.09.007.
- [7] [7] Z. Chen, J. Feng, K. Zhu, Z. Yang, Y. Wang, dan M. Ren, “YOLOv8-ACCW: Lightweight Grape Leaf Disease Detection Method Based on Improved YOLOv8,” *IEEE Access*, vol. 12, no. September, hal. 123595–123608, 2024, doi: 10.1109/ACCESS.2024.3453379.
- [8] [8] C. Wang, Y. Wang, G. Ma, G. Bian, dan C. Ma, “Identification of Grape Diseases Based on Improved YOLOX,” *Appl. Sci.*, vol. 13, no. 10, 2023, doi: 10.3390/app13105978.
- [9] [9] Z. Zhang, Y. Qiao, Y. Guo, dan D. He, “Deep Learning Based Automatic Grape Downy Mildew Detection,” *Front. Plant Sci.*, vol. 13, no. June, hal. 1–12, 2022, doi: 10.3389/fpls.2022.872107.
- [10] [10] S. Bin Mamun, I. J. Payel, M. T. Ahad, A. S. Atkins, B. Song, dan Y. Li, “Grape Guard: A YOLO-based mobile application for detecting grape leaf diseases,” *J. Electron. Sci. Technol.*, vol. 23, no. 1, hal. 15, 2025, doi: 10.1016/j.jnlst.2025.100300.
- [11] [11] J. Chen dkk., “GA-YOLO: A Lightweight YOLO Model for Dense and Occluded Grape Target Detection,” *Horticulturae*, vol. 9, no. 4, hal. 1–22, 2023, doi: 10.3390/horticultrae9040443.
- [12] [12] E. Alfonsius, “Pengembangan Sistem Identifikasi Penyakit Tanaman Anggur Berdasarkan Citra Daun Menggunakan Algoritma Yolov8 (You Only Look Once),” vol. 6, no. 2, hal. 791–802, 2025, doi: 10.47065/josh.v6i2.6235.