

Optimalisasi Deteksi Tingkat Kematangan Tanda Buah Segar Kelapa Sawit Menggunakan YOLOv8 Dengan Platform Web

¹Iffatul Mardhiyah, ²Dyan Prawita Sari, ³Zahwa Genoveva, ^{4*}Rifki Kosasih, ⁵Dyah Cita Irawati

^{1,2,3,4,5} Informatika, Fakultas Teknologi Industri, Universitas Gunadarma, Depok, Indonesia

^{1,2,3,4,5} Jl. Margonda Raya 100, Depok, Jawa Barat, Indonesia

¹iffatul@staff.gunadarma.ac.id, ²wprawita@staff.gunadarma.ac.id, ³zahwagenoveva09@gmail.com,

⁴rifki_kosasih@staff.gunadarma.ac.id, ⁵dyahcita@staff.gunadarma.ac.id.

Abstract

Oil palm represents one of Indonesia's principal commodities. Traditionally, farmers manually monitor the ripeness level of palm oil, but this method is neither effective nor efficient for large-scale harvests. Therefore, a system that can automatically detect the ripeness level of fresh fruit bunches (FFB) is needed. In this study, the YOLOv8 algorithm was used which was integrated into a web-based application. The system is designed to improve accuracy and efficiency in the grading process of oil palm fruits, which directly impacts the quality of processed products and palm oil production. The dataset used consists of 6.592 images obtained through the Roboflow platform, covering various ripeness categories. The system development follows the CRISP-DM approach, consisting of business understanding, data understanding, data preparation, modeling, evaluation and deployment. The model training process approximately 3,1 hours, with evaluation results showing a precision of 94,5%, recall of 94,7%, and a mean Average Precision (mAP) of 98%. The model's performance is further supported by an F1-confidence curve of 95% and a precision-recall curve of 98%, indicating stable and accurate classification capabilities. The model is deployed through a Streamlit-based web interface, allowing users to perform real-time detection from images or videos without requiring additional installations.

Keywords: CRISP-DM, fresh fruit bunches (FFB), machine learning, object detection, YOLOv8.

Abstrak

Kelapa sawit termasuk komoditas utama di Indonesia. Untuk memanen kelapa sawit biasa dengan melihat tingkat kematangannya secara langsung oleh petani sawit, akan tetapi cara ini tidak efektif dan efisien jika diterapkan dalam panen dengan jumlah besar. Oleh karena itu dibutuhkan suatu sistem yang dapat mendeteksi secara otomatis tingkat kematangan Tandan Buah Segar (TBS) kelapa sawit. Dalam penelitian ini digunakan algoritma YOLOv8 yang diintegrasikan ke dalam aplikasi berbasis *web*. Sistem dirancang untuk meningkatkan ketepatan dan efisiensi dalam proses *grading* buah sawit, yang berpengaruh langsung terhadap kualitas hasil olah dan proses produksi minyak sawit. *Dataset* yang digunakan terdiri dari 6.592 citra yang diperoleh melalui platform Roboflow, mencakup berbagai kategori tingkat kematangan buah. Pengembangan sistem mengikuti pendekatan CRISP-DM, meliputi tahapan pemahaman bisnis, pemahaman data, persiapan *dataset*, pemodelan, evaluasi, dan *deployment*. Proses pelatihan model berlangsung selama ±3,1 jam, dengan hasil evaluasi menunjukkan nilai *precision* sebesar 94,5%, *recall* 94,7%, dan mAP mencapai 98%. Kinerja model juga diperkuat oleh nilai *F1-confidence curve* sebesar 95% dan *precision-recall curve* sebesar 98%, yang mengindikasikan kemampuan klasifikasi yang stabil dan akurat. Model diterapkan melalui antarmuka Streamlit berbasis *web*, memungkinkan pengguna melakukan deteksi langsung dari citra atau video tanpa memerlukan instalasi tambahan.

Kata Kunci: CRISP-DM, deteksi objek, *machine learning*, TBS kelapa sawit, YOLOv8.

1. Pendahuluan

Kemajuan teknologi pada era digital telah mendorong berbagai perubahan besar di banyak bidang, termasuk sektor pertanian [1], [2]. Sebagai salah satu produsen kelapa sawit terbesar di dunia, Indonesia tercatat memiliki areal perkebunan sekitar 15,4 juta hektar dengan total produksi mencapai 47 juta ton pada tahun 2023 [3]. Kelapa sawit memegang peran penting dalam perekonomian Indonesia, terutama sebagai sumber lapangan pekerjaan dan pendapatan bagi masyarakat.

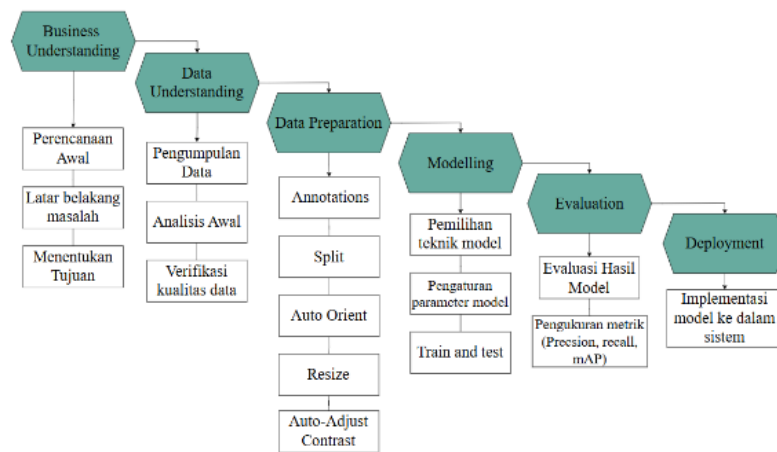
Menurut Sari, laju ekstraksi minyak dari TBS (Tandan Buah Segar) kelapa sawit sangat dipengaruhi oleh fase kematangannya [4]. TBS yang dipanen pada tingkat kematangan optimal umumnya menghasilkan minyak dengan kualitas tinggi serta kandungan asam lemak bebas yang rendah, sehingga perlu dilakukan klasifikasi yang akurat dalam proses *grading*. *Grading* adalah aktivitas pengelompokan TBS berdasarkan tingkat kematangan dan kualitas, seperti buah matang, kurang matang, terlalu matang dan mentah. Penilaian biasanya dilakukan secara visual, salah satunya melalui perubahan warna kulit buah [5]–[8].

Grading secara manual memiliki kekurangan. Penilaian yang bergantung pada pengamatan visual dan subjektivitas petugas sering kali menyebabkan ketidakkonsistenan. Akibatnya, buah yang tidak layak olah bisa tercampur dengan buah berkualitas yang dapat menurunkan mutu minyak. Untuk mengatasi hal tersebut, dibutuhkan solusi yang lebih akurat dan konsisten. Salah satunya adalah sistem *grading* otomatis berbasis teknologi pengolahan citra (*image processing*) dan kecerdasan buatan (*machine learning*). Perkembangan teknologi *Artificial Intelligence* (AI) menawarkan solusi yang mampu mengatasi tantangan dalam mendeteksi tingkat kematangan TBS kelapa sawit, khususnya melalui pendekatan *computer vision*. Salah satu metode yang efektif dalam menentukan kematangan buah kelapa sawit adalah dengan memanfaatkan model *deep learning* [5], [9]–[12]. Pendeteksi objek adalah aplikasi komputer yang bertujuan untuk mendeteksi dan mengidentifikasi objek tertentu [13], [14]. Kemampuan mendeteksi objek ini merupakan bagian dari ilmu *computer vision*, yang mempelajari cara komputer dapat bekerja seperti mata manusia untuk mengenali objek [15]–[18].

Beberapa penelitian terdahulu telah dilakukan untuk menangani klasifikasi tingkat kematangan TBS. Sensor optik untuk mendeteksi tingkat kematangan berdasarkan kekerasan buah dan tegangan sensor [19]. Meskipun akurat, pendekatan tersebut masih berbasis perangkat keras dan belum memanfaatkan pengolahan citra digital. Perkembangan model CNN dengan arsitektur MobileNet dan mencapai akurasi tinggi dalam klasifikasi tiga kelas kematangan. Meski hasilnya menjanjikan, model tersebut hanya diuji pada *dataset* terbatas dan belum diterapkan dalam sistem aplikasi yang dapat digunakan lebih luas [20].

Berdasarkan uraian tersebut, penelitian ini bertujuan mengembangkan model *deep learning* untuk mendeteksi tingkat kematangan kelapa sawit dengan memanfaatkan *dataset* dari platform Roboflow [21] yang memuat berbagai variasi citra buah pada fase kematangan berbeda. Data tersebut digunakan untuk melatih model YOLOv8, sementara kinerja algoritma dievaluasi menggunakan metrik standar seperti akurasi, F1-score, dan mAP. Setelah proses pelatihan selesai, model diimplementasikan ke dalam aplikasi berbasis web melalui platform Streamlit. Sistem yang dikembangkan tidak diarahkan pada pengumpulan umpan balik pengguna, melainkan difokuskan untuk menguji fungsionalitas model dalam mendeteksi tingkat kematangan secara otomatis.

2. Metode Penelitian



Gambar 1. Metode CRISP-DM

Pada penelitian ini digunakan sejumlah perangkat lunak, yaitu Google Colab dan Roboflow, serta data citra TBS kelapa sawit dengan penerapan metode YOLOv8. Metode penelitian yang digunakan adalah CRISP-DM, yakni pendekatan standar yang banyak digunakan dalam proses pengembangan sistem berbasis *data mining* karena memiliki alur kerja yang terstruktur, fleksibel, dan telah terbukti efektif di berbagai sektor [22]. Rangkaian tahapan CRISP-DM ditunjukkan pada Gambar 1.

Penelitian ini berfokus pada pendeteksian tingkat kematangan buah kelapa sawit melalui identifikasi kategori TBS. Tahapan penelitian mencakup tahapan *business understanding*, *data understanding*, *data preparation*, *modelling*, *evaluation*, serta *deployment* dengan menerapkan metode YOLOv8. *Dataset* yang digunakan dalam penelitian ini berasal dari platform Roboflow, dengan setiap citra akan diberi label “masak”, “kurang masak”, “terlalu masak” dan “mentah”.

2.1. Business Understanding





Pada tahap *business understanding*, telah dilakukan studi literatur yang menunjukkan bahwa perbedaan cara pandang dalam menilai kematangan TBS kelapa sawit sering menjadi penyebab utama ketidak konsistenan dalam pemilihan buah. Untuk mengatasi masalah tersebut, dibutuhkan sistem otomatis yang bisa menentukan tingkat kematangan buah dengan lebih tepat dan konsisten. Salah satu solusi yang ditawarkan adalah penggunaan teknologi kecerdasan buatan, khususnya metode pendeteksian objek seperti YOLOv8.

2.2. Data Understanding

Sama seperti manusia, komputer juga perlu diperkenalkan pada objek yang dipelajarinya. Salah satu cara mengenalkan objek tersebut adalah dengan memberikan kumpulan citra objek. Dalam kasus ini, objek yang dimaksud adalah citra TBS kelapa sawit dengan berbagai tingkat kematangan. Data yang digunakan dalam proses pembelajaran disebut sebagai *data understanding*. *Dataset* yang diperlukan untuk pembuatan program terdiri dari 6.592 citra yang diperoleh dari platform Roboflow yang mencakup berbagai tingkat kematangan TBS kelapa sawit [21]. Distribusi jumlah citra pada masing-masing kelas disajikan pada Tabel 1.

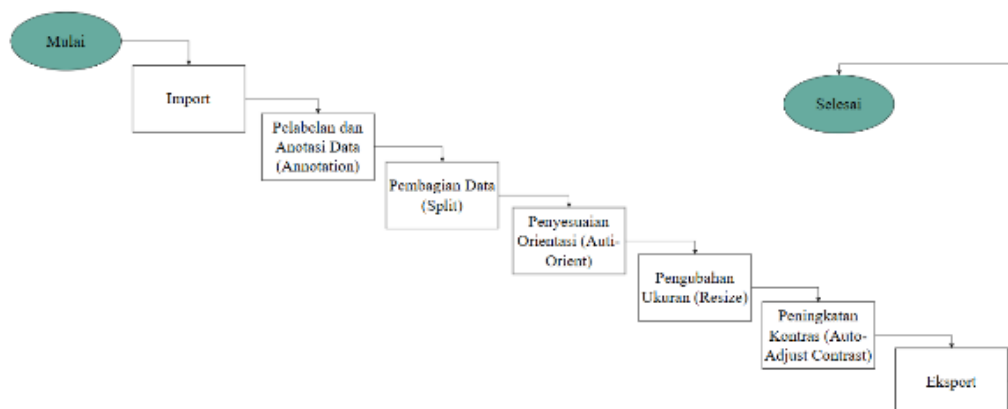
Berdasarkan pada Tabel 1, *dataset* yang digunakan memiliki 4 kelas dan masing-masing kelas memiliki total citra yang berbeda. TBS Matang sebanyak 2.118 citra, TBS Kurang Matang sebanyak 1.722 citra, TBS Terlalu Matang sebanyak 1.948 citra, dan TBS Mentah sebanyak 1.492 citra.

Tabel 1. Jumlah Citra Tiap Kategori Tingkat Kematangan TBS

Citra	Jenis	Jumlah
	TBS Matang	2.118
	TBS Kurang Matang	1.722
	TBS Terlalu Matang	1.948
	TBS Mentah	1.492

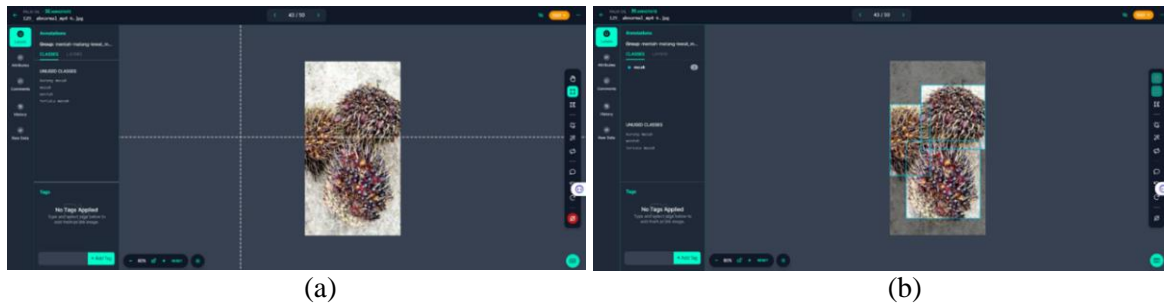
2.3.Data Preparation

Preprocessing pada penelitian ini dilakukan menggunakan Roboflow. Tahapan *preprocessing* yang dilakukan adalah anotasi, *split data*, *auto orient*, *resize* dan *auto-adjust-contrast*. Langkah-langkah yang dilakukan digambarkan secara visual pada Gambar 2.



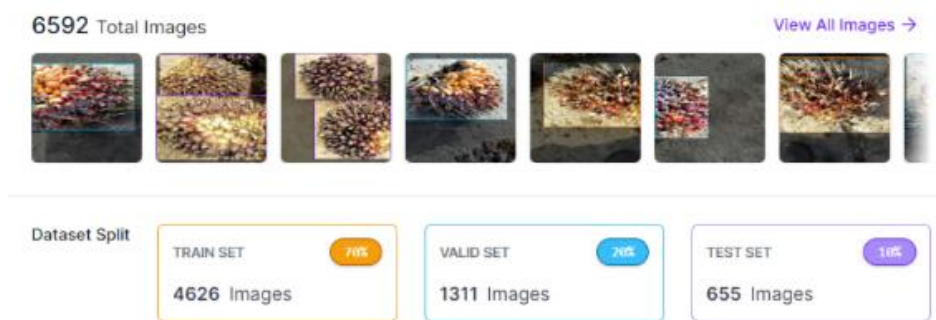
Gambar 2. Flowchart Tahap Data Preparation

Tahap pertama dalam proses persiapan data adalah melakukan kloning *dataset* dari platform Roboflow ke dalam *workspace* atau lingkungan kerja yang digunakan untuk pelatihan model. Proses kloning bertujuan agar seluruh data dapat diakses secara lokal, sehingga memudahkan pengguna dalam melakukan proses selanjutnya seperti pelabelan, augmentasi, serta pembagian data. Tahapan anotasi melibatkan penerapan label pada citra yang menunjukkan posisi dan jenis citra [22]. Penggunaan *labeling* pada Roboflow dilakukan untuk menggambar *bounding box* anotasi di sekitar objek dan memberikan informasi kepada model mengenai posisi dan jenis objek yang terdapat pada citra. Proses anotasi memiliki peran krusial dalam menghasilkan *dataset* latih berkualitas tinggi, serta mendukung kelancaran proses pengenalan objek pada tahap selanjutnya. Gambar 3 menunjukkan tahapan anotasi.



Gambar 3. Tahapan Anotasi pada Roboflow (a) Sebelum Anotasi (b) Sesudah Anotasi

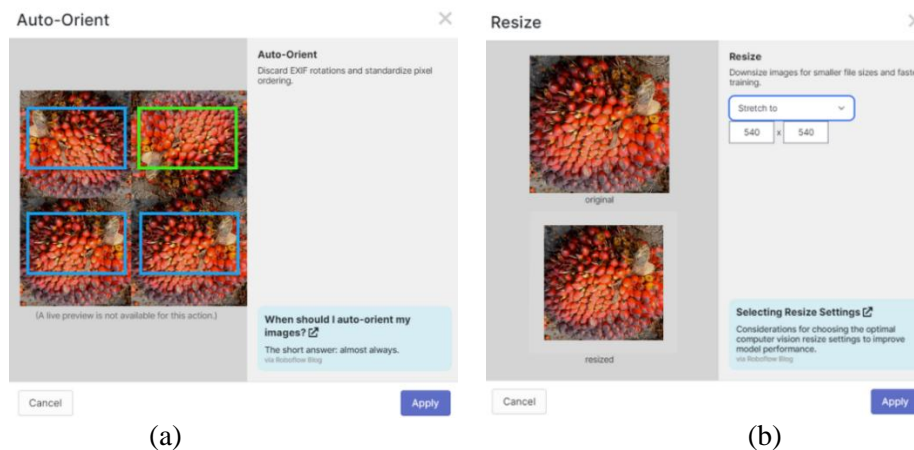
Tahapan *split dataset* dilakukan dengan membagi data menjadi tiga *subset*, yaitu *train*, *validation*, dan *test*, menggunakan platform Roboflow dengan rasio 7:2:1. Pembagian ini bertujuan untuk mengatur data yang digunakan dalam penelitian agar proses pelatihan, validasi, dan pengujian model dapat berjalan secara terstruktur sesuai prosedur. Ilustrasi pembagian *dataset* ditunjukkan pada Gambar 4.



Gambar 4. Tahapan *Split Dataset* pada Roboflow

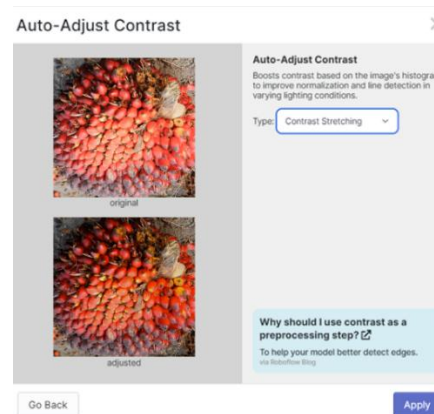
Tahapan *auto-orient* diterapkan untuk mengatasi perbedaan orientasi objek pada citra yang ada di *dataset*. Fitur ini secara otomatis menyesuaikan posisi objek sehingga memiliki orientasi yang konsisten, yang berkontribusi terhadap ketepatan dan keseragaman data. Proses *auto-orient* dapat dilihat pada Gambar 5(a).

Tahapan *resize* bertujuan mengubah ukuran citra menjadi 540×540 piksel melalui platform Roboflow. Ukuran ini dipilih karena cukup besar untuk mempertahankan detail penting, namun tetap ringan secara komputasi sehingga mempercepat proses pelatihan model. Penyamaan dimensi citra juga membuat model dapat memproses data dengan lebih konsisten tanpa harus menyesuaikan setiap ukuran citra yang berbeda. Tahapan ini ditampilkan pada Gambar 5(b).



Gambar 5. (a) Tahapan *Auto-Orient* pada Roboflow, (b) Tahapan *Resize* pada Roboflow

Tahap *auto-adjust contrast* menerapkan teknik *contrast stretching* untuk meningkatkan tingkat kontras citra berdasarkan distribusi histogramnya. Metode ini membantu menormalkan pencahayaan dan memperjelas garis atau detail pada berbagai kondisi pencahayaan. Dengan kontras yang lebih tinggi, perbedaan antar piksel menjadi lebih jelas sehingga objek pada citra dapat terdeteksi dengan lebih baik. Proses ini diperlihatkan pada Gambar 6.

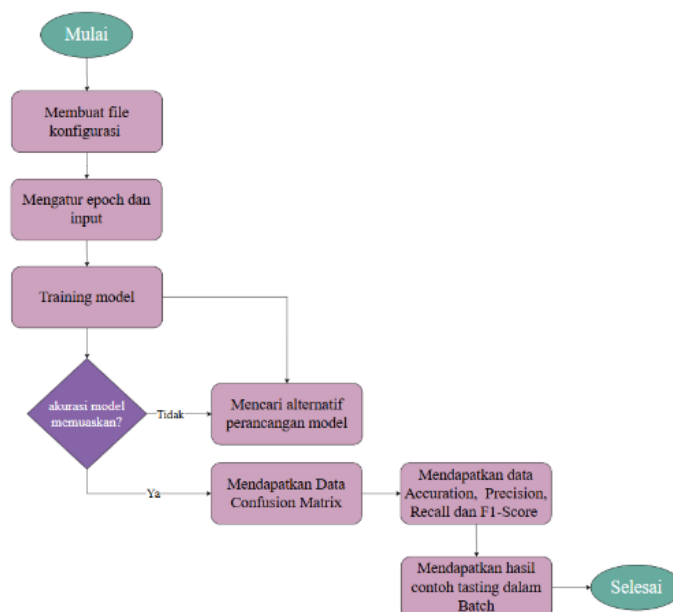


Gambar 6. Tahapan *Auto-Adjust Contrast* pada Roboflow

2.4. Modelling

Tahap *modelling* difokuskan pada proses pengembangan serta pelatihan model dengan memanfaatkan *dataset* yang telah dipersiapkan sebelumnya. Model yang digunakan dalam penelitian ini adalah YOLOv8 dengan bobot terbaik hasil pelatihan (*best.pt*). Model ini dibangun dengan tujuan untuk mengenali dan membedakan objek pada citra. Proses ini melibatkan tahap pelatihan model dengan memanfaatkan *dataset* yang telah mengalami sejumlah langkah *preprocessing*, termasuk *split*, anotasi, dan penyesuaian ukuran. Proses pelatihan model dilakukan di *Google Colaboratory*. Pelatihan model dilakukan sebanyak 100 *epoch*.

Setiap iterasi (*epoch*) membantu model mempelajari lebih dalam tentang variasi dalam data, sehingga model dapat mendeteksi tingkat kematangan buah dengan lebih akurat dan konsisten. *Flowchart* untuk pelatihan model terdapat pada Gambar 7.



Gambar 7. *Flowchart* Pelatihan Model

Berdasarkan *flowchart* pada Gambar 7, pelatihan model bertujuan untuk mengajarkan model bagaimana mengenali dan membedakan tingkat kematangan TBS kelapa sawit dari berbagai citra yang telah disiapkan, sehingga model dapat bekerja secara akurat dan efisien. Setelah model selesai dilatih, hasil akhir dari pelatihan akan ditampilkan. Hasil terbaik dari pelatihan disimpan dalam *file* bernama *best.pt*.

2.5. Evaluation

Setelah model selesai dilatih, tahap selanjutnya adalah melakukan evaluasi untuk mengetahui seberapa baik model mampu mendeteksi objek pada data pengujian. Pada tahap evaluasi, kinerja model dianalisis menggunakan sejumlah metrik, antara lain *confusion matrix*, *accuracy*, *precision*, *recall*, *F1-score*, dan mAP.

2.6. Deployment

Proses *deployment* sistem deteksi tingkat kematangan TBS kelapa sawit menggunakan algoritma YOLOv8 dilakukan melalui platform Streamlit Cloud, yang memungkinkan aplikasi diakses secara online tanpa memerlukan instalasi khusus. *Deployment* dimulai dengan menyiapkan *repository* GitHub yang berisi *source code* dan dependensi, kemudian menghubungkannya dengan akun Streamlit Cloud. Setelah konfigurasi awal diselesaikan, aplikasi kemudian di-*deploy* dan diuji untuk memastikan seluruh fitur berjalan dengan optimal.

3. Hasil dan Pembahasan

Model dibuat menggunakan algoritma YOLOv8 dengan pendekatan CRISP-DM, lalu diuji menggunakan metrik evaluasi yaitu akurasi, *precision*, dan *recall* untuk menilai kinerjanya. Setelah pelatihan selesai, model diintegrasikan ke dalam aplikasi berbasis web dengan memanfaatkan Streamlit. Melalui *platform* ini, pengguna dapat mengunggah citra maupun video dan memperoleh hasil deteksi tingkat kematangan TBS kelapa sawit secara otomatis serta mudah digunakan.

3.1. Analisis Kinerja Model

Proses evaluasi dilakukan untuk menilai performa model dalam mengidentifikasi tingkat kematangan TBS kelapa sawit pada citra. Penilaian ini menggunakan beberapa pendekatan pengukuran, antara lain mAP_{matrix}, kurva hubungan antara *precision* dengan *recall*, kurva hubungan antara F1 dengan *confidence*, dan *confusion matrix*.

a. mAP Matrix

mAP digunakan untuk mengukur performa keseluruhan model dalam mendeteksi dan mengenali berbagai kelas objek secara akurat [23]. Perhitungan mAP dilakukan dengan membandingkan *bounding box* hasil prediksi model dengan *bounding box* sebenarnya (*ground truth*). Dalam penelitian ini, mAP dihitung pada dua ambang nilai berbeda, yaitu mAP₅₀ dan mAP_{50–95}. Perbedaan ambang ini mencerminkan tingkat kesulitan deteksi yang beragam, sekaligus mengukur presisi model dalam berbagai kondisi. Matriks mAP terdapat pada Gambar 8.

Pada Gambar 8 ditunjukkan matriks mAP, di mana hasil pelatihan selama $\pm 3,1$ jam menghasilkan mAP sebesar 0,98 atau 98%. Nilai ini mengindikasikan bahwa model memiliki akurasi deteksi yang sangat tinggi, karena semakin besar nilai mAP, semakin tepat model dalam mengenali objek sesuai data latih.

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
97/100	3.02G	0.6022	0.2837	1.085	5	544: 100% 290/290 [01:33<00:00, 3.10it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% 41/41 [00:16<00:00, 2.53it/s]
	all	1311	2654	0.933	0.957	0.975 0.785
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
98/100	2.99G	0.5973	0.2786	1.085	4	544: 100% 290/290 [01:28<00:00, 3.28it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% 41/41 [00:14<00:00, 2.84it/s]
	all	1311	2654	0.936	0.952	0.976 0.787
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
99/100	3.02G	0.5968	0.274	1.082	6	544: 100% 290/290 [01:32<00:00, 3.13it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% 41/41 [00:16<00:00, 2.50it/s]
	all	1311	2654	0.934	0.955	0.976 0.786
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
100/100	2.99G	0.5919	0.2719	1.078	3	544: 100% 290/290 [01:29<00:00, 3.26it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% 41/41 [00:14<00:00, 2.80it/s]
	all	1311	2654	0.938	0.953	0.976 0.785

100 epochs completed in 3.107 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.5MB

Validating runs/detect/train/weights/best.pt...

Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11127132 parameters, 0 gradients, 28.4 GFLOPs

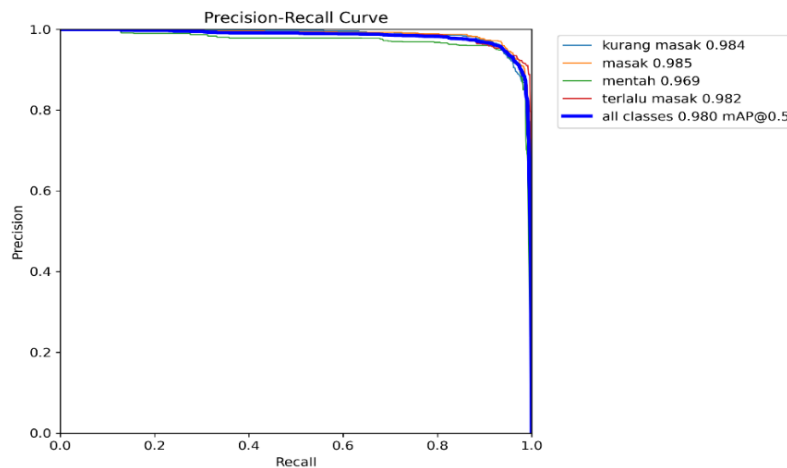
Class	Images	Instances	Box(P	R	mAP50	mAP50-95)
all	1311	2654	0.945	0.947	0.98	0.791
kurang masak	1311	757	0.958	0.934	0.984	0.78
masak	1311	556	0.939	0.96	0.985	0.843
mentah	1311	706	0.936	0.952	0.969	0.749
terlalu masak	1311	635	0.945	0.944	0.982	0.791

Speed: 0.2ms preprocess, 3.2ms inference, 0.0ms loss, 2.0ms postprocess per image
Results saved to runs/detect/train

Gambar 8. mAP Matrix

b. Precision dan Recall

Precision didefinisikan sebagai perbandingan antara jumlah prediksi positif yang tepat dengan seluruh prediksi positif yang dihasilkan oleh model [24], [25]. Sementara itu, *recall* merupakan perbandingan antara prediksi positif yang benar dengan keseluruhan objek positif yang terdapat dalam data sebenarnya. Gambar 9 menunjukkan kurva hubungan antara nilai *precision* dan *recall*.



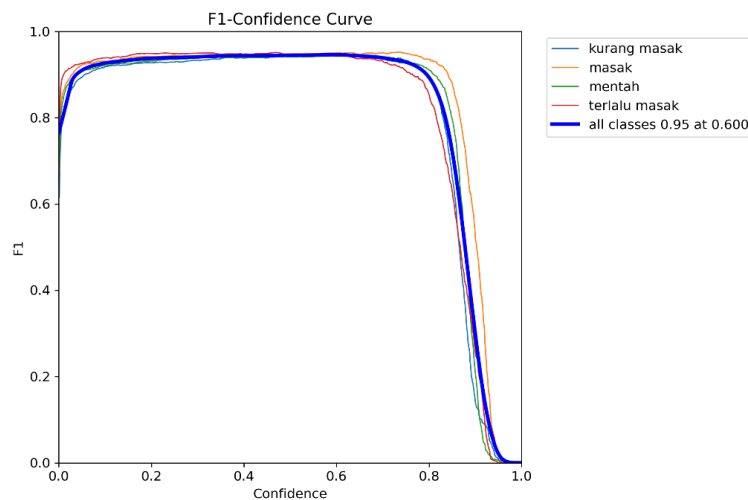
Gambar 9. Kurva Hubungan antara Precision dan Recall

Kurva pada Gambar 9 memperlihatkan bahwa sistem deteksi yang dikembangkan menghasilkan nilai *precision* dan *recall* yang tinggi pada seluruh kelas, dengan capaian sebesar 0,98 atau 98%. Hasil ini menandakan bahwa model mampu meminimalkan kesalahan prediksi sekaligus berhasil mengenali sebagian besar objek yang relevan dalam citra.

c. F1-Score dan Confidence

F1-score dihitung sebagai rata-rata harmonis dari *precision* dan *recall* sehingga mampu merepresentasikan keseimbangan antara kedua metrik tersebut [26], [27]. Sementara itu,

confidence menggambarkan tingkat keyakinan model terhadap hasil deteksi yang dihasilkan. Kurva *F1-Confidence* pada Gambar 10.

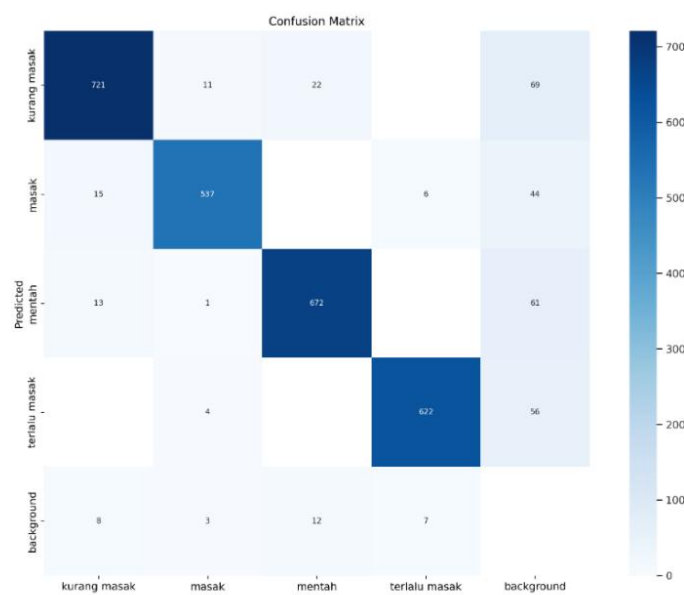


Gambar 10. Kurva Hubungan antara F1 dan *Confidence*

Gambar 10 memperlihatkan kurva hubungan antara *F1-score* dan *confidence*. Hasil terbaik menunjukkan nilai *F1-score* dan *confidence* tertinggi sebesar 95%. Temuan ini menunjukkan bahwa model mampu menjaga keseimbangan yang baik antara akurasi deteksi dan tingkat keyakinan prediksi, serta memiliki kemampuan tinggi dalam mengidentifikasi objek pada citra.

d. *Confusion Matrix*

Confusion matrix merupakan instrumen evaluasi yang digunakan untuk menilai kemampuan model dalam melakukan klasifikasi pada data pengujian. Matriks ini mengelompokkan hasil prediksi ke dalam empat kategori, yaitu *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). Melalui *confusion matrix*, dapat diketahui jumlah sampel yang berhasil diklasifikasikan secara benar sebagai data positif maupun negatif, serta berapa banyak data yang salah diklasifikasikan [28]. Ini memberi gambaran tentang seberapa baik model dapat mengenali objek atau fenomena yang dimaksud.



Gambar 11. *Confusion Matrix*

Pada Gambar 11, terlihat bahwa persentase prediksi benar untuk setiap kelas (*True Positive*) cukup tinggi, mencapai sekitar 97%. Namun, masih terdapat *False Negative*, yaitu objek yang seharusnya terdeteksi tetapi tidak terdeteksi, dengan angka tertinggi sebesar 3% pada kelas “mentah”. Hasil ini mengindikasikan bahwa model menunjukkan kinerja yang sangat baik, walaupun masih terdapat keterbatasan pada beberapa kelas tertentu.

e. Batch

Pendekatan *batch processing* digunakan untuk menangani ukuran *dataset* yang besar, dengan cara membagi data menjadi beberapa bagian kecil. Setiap *batch* memuat sejumlah sampel yang diproses dalam satu iterasi pelatihan. Metode ini juga diterapkan pada tahap validasi untuk memantau performa model di setiap langkah pelatihan. Dengan menggunakan pendekatan ini, pelatihan model dapat dilakukan secara bertahap dan efisien, memungkinkan penggunaan sumber daya komputasi yang optimal. *Batch* sampel data terdapat pada Gambar 12.



Gambar 12. *Batch Processing*

Gambar 12 memperlihatkan hasil prediksi model pada *batch* sampel data. Setiap *bounding box* berwarna dilengkapi label teks yang menunjukkan kelas tingkat kematangan buah kelapa sawit, disertai nilai *confidence score*. Skor ini merepresentasikan tingkat

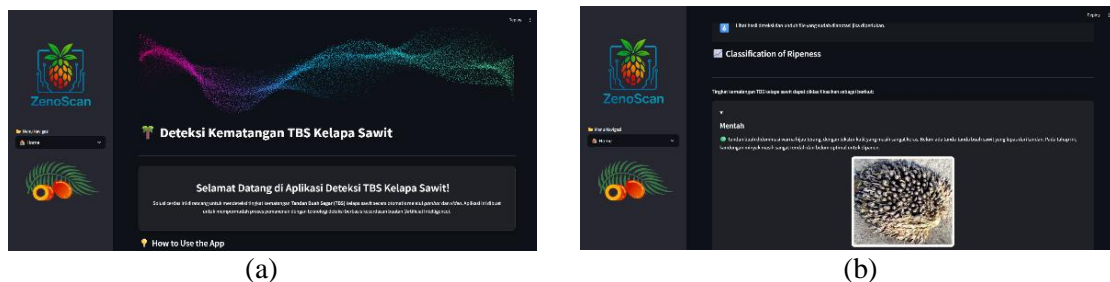
keyakinan model terhadap prediksi yang dihasilkan. Hasil yang ditampilkan membuktikan bahwa model mampu mendeteksi dan mengklasifikasikan objek dengan cukup akurat berdasarkan tingkat kematangannya.

3.2. Pengembangan Sistem

Tahap selanjutnya adalah mengimplementasikan model ke dalam sistem aplikasi. Pengembangan aplikasi deteksi tingkat kematangan TBS kelapa sawit dilakukan dengan menggunakan *framework* Streamlit. *Framework* ini dipilih karena kemampuannya dalam menyajikan antarmuka pengguna berbasis *web* secara cepat, interaktif, dan mudah diintegrasikan dengan model *machine learning* seperti YOLOv8.

a. Halaman Beranda (*Home*)

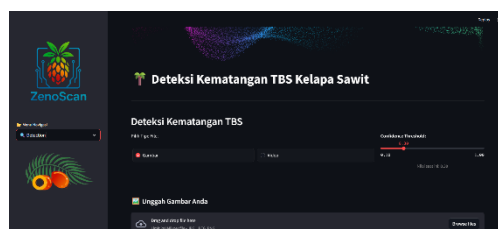
Halaman *Home* menampilkan penjelasan singkat mengenai fungsi aplikasi, tujuan penggunaan, panduan pengguna, serta deskripsi visual terhadap masing-masing tingkat kematangan TBS kelapa sawit yang dideteksi. Komponen visual ditampilkan menggunakan fungsi *st.image()* dan *st.caption()* untuk memudahkan pemahaman pengguna. Detail hasil tampilan halaman *Home* dapat dilihat pada Gambar 13(a) dan Gambar 13(b).



Gambar 13. (a) Tampilan Awal Beranda, (b) Visualisasi Tingkat Kematangan

b. Halaman Deteksi Objek Menggunakan Unggah Gambar dan Video

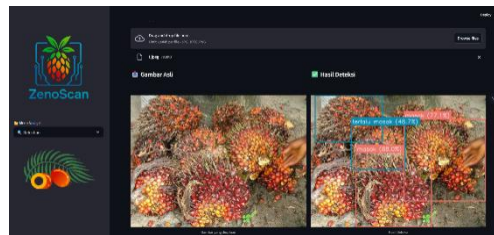
Halaman Deteksi merupakan inti dari sistem aplikasi, tempat pengguna dapat melakukan proses klasifikasi tingkat kematangan TBS kelapa sawit. Pengguna dapat mengunggah *input* berupa citra maupun video, yang kemudian diproses menggunakan model deteksi objek YOLOv8. Hasil deteksi ditampilkan secara visual dalam bentuk citra atau video yang telah dianotasi dengan label klasifikasi dan tingkat kepercayaannya (*confidence score*). Detail tampilan hasil dari berbagai kasus deteksi dapat dilihat pada Gambar 14 sampai Gambar 19. Tampilan awal halaman ditunjukkan pada Gambar 14, di mana pengguna diberikan opsi untuk mengunggah *file* citra atau video sesuai format yang didukung (seperti jpg, png, mp4, avi).



Gambar 14. Tampilan Awal Deteksi

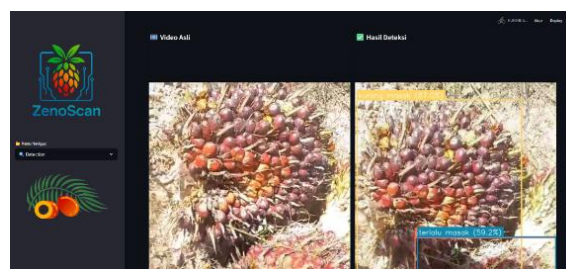
Setelah pengguna mengunggah citra, sistem memproses data tersebut menggunakan model YOLOv8 dan menampilkan hasil deteksi sebagaimana ditunjukkan pada Gambar 15. Hasil deteksi ditampilkan dalam dua bagian, yaitu citra asli dan citra hasil deteksi yang telah

diberi anotasi berupa *bounding box*, label kelas kematangan (contohnya “Matang”, “Mentah”), serta *confidence score* dari model.



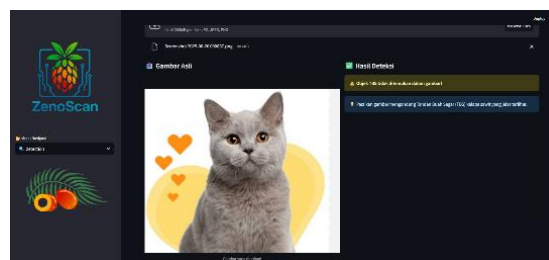
Gambar 15. Deteksi Citra

Jika pengguna memilih untuk mengunggah video, sistem mengekstrak *frame* per *frame* dari video tersebut dan melakukan deteksi terhadap setiap *frame* yang relevan. Contoh tampilan proses deteksi video dapat dilihat pada Gambar 16.



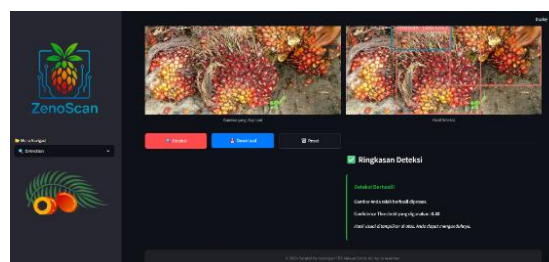
Gambar 16. Deteksi Video

Sistem juga dirancang untuk memberikan *feedback* ketika *file input* tidak sesuai dengan objek yang dapat dikenali oleh model. Sebagai contoh, jika pengguna mengunggah citra yang tidak relevan (misalnya citra kucing), sistem memberikan notifikasi peringatan bahwa tidak ada objek yang berhasil terdeteksi (Gambar 17).



Gambar 17. Deteksi Gagal (Citra Tidak Relevan)

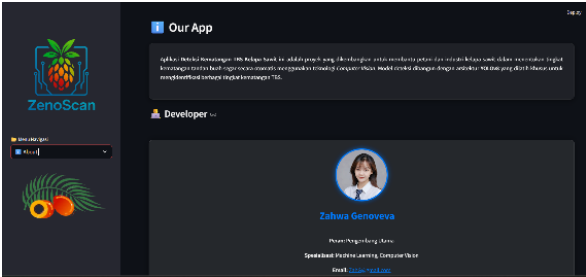
Setelah proses deteksi selesai, pengguna diarahkan ke bagian hasil, seperti yang ditampilkan pada Gambar 18. Halaman tersebut menyajikan status deteksi yang berhasil, serta menyediakan tombol untuk mengunduh hasil deteksi dalam format yang telah ditentukan.



Gambar 18. Tampilan Akhir Deteksi

c. Halaman Tentang Aplikasi (About)

Halaman *About* disediakan untuk memberikan informasi tambahan mengenai latar belakang aplikasi serta memperkenalkan identitas pengembang sistem. Desain halaman dibuat sederhana namun tetap informatif, agar pengguna dapat memahami siapa yang berada di balik pengembangan aplikasi serta tujuan utama dari sistem yang dibangun. Tampilan halaman *About* dapat dilihat pada Gambar 19.



Gambar 19. Tampilan Halaman *About*

3.3. Uji Model

Sistem yang dikembangkan untuk mendeteksi tingkat kematangan TBS kelapa sawit dapat dilakukan dari sumber data seperti *file* video dan citra. Pengujian dilakukan pada citra baru dan video. Pengujian dilakukan dengan *best weights* yang berarti iterasi *dataset* terbaik yang telah diproses.

a. Uji Coba Model

Tabel 2. Evaluasi Uji Model

No.	Citra Asli	Citra Hasil Deteksi	Keterangan
1.			Berhasil deteksi objek dengan benar
2.			Berhasil deteksi objek dengan benar
3.			Berhasil deteksi objek dengan benar
4.			Berhasil deteksi objek dengan benar
5.			Berhasil deteksi objek dengan benar

Pengujian model bertujuan untuk mengetahui sejauh mana tingkat ketepatan model dalam mendeteksi tingkat kematangan TBS kelapa sawit dari citra yang baru. Hasil prediksi dari model dievaluasi dengan membandingkan antara label sebenarnya dengan hasil klasifikasi model. Evaluasi dilakukan untuk mengukur efektivitas model dalam kondisi nyata serta untuk mengidentifikasi potensi kekurangan yang dapat diperbaiki pada pengembangan selanjutnya. Hasil pengujian model disajikan dalam Tabel 2 yang menampilkan performa prediksi pada masing-masing citra uji.


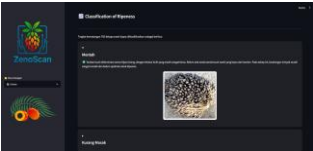

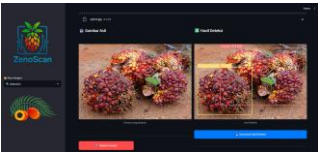
Berdasarkan hasil uji coba model yang disajikan pada Tabel 2, seluruh sampel citra uji berhasil dideteksi dengan benar oleh sistem. Model menunjukkan kemampuan mengklasifikasikan tingkat kematangan TBS kelapa sawit secara akurat pada kelima citra yang diuji, tanpa kesalahan klasifikasi.

b. Uji Coba

Pengujian sistem dilakukan untuk menilai kinerja keseluruhan aplikasi yang telah dibangun. Hasil dari pengujian dituangkan dalam Tabel 3, yang menunjukkan status keberhasilan dari setiap fitur yang diuji.

Tabel 3 merupakan hasil uji coba terhadap berbagai fitur utama pada aplikasi deteksi tingkat kematangan TBS kelapa sawit. Pengujian dilakukan menggunakan *blackbox testing*. Hasil pengujian menunjukkan bahwa sistem berhasil menjalankan seluruh komponen secara optimal, termasuk kemampuan dalam memuat model deteksi, menerima *input* citra dari pengguna, melakukan proses deteksi objek, menampilkan hasil dengan anotasi pada citra, serta menyimpan hasil prediksi.

Tabel 3. Uji Coba *Black Box* untuk App/ Website

No.	Skenario	Gambar	Hasil	Kesimpulan
1	Aplikasi/Website deteksi kematangan TBS kelapa sawit		Menampilkan halaman awal app/website	Berhasil
2	Beranda		Menampilkan halaman beranda	Berhasil
3	Tentang		Menampilkan halaman Tentang	Berhasil
4	Pengenalan deteksi		Menampilkan halaman pengenalan deteksi	Berhasil
5	Deteksi kematangan buah kelapa sawit		Menampilkan hasil deteksi	Berhasil

4. Kesimpulan

Pendeteksian tingkat kematangan TBS kelapa sawit secara otomatis sangat dibutuhkan untuk mempercepat proses pemilihan kelapa sawit yang siap diproduksi. Dalam penelitian ini digunakan metode *deep learning* dengan model YOLOv8. Model dilatih menggunakan dataset berisi 6592 citra dari Roboflow yang mewakili empat tingkat kematangan. Seluruh proses pengembangan mengikuti pendekatan CRISP-DM. Model kemudian diintegrasikan ke dalam aplikasi berbasis Streamlit sehingga memungkinkan deteksi kematangan secara langsung melalui citra atau video, tanpa memerlukan proses instalasi tambahan. Penerapan sistem deteksi mendukung proses *grading* yang lebih akurat dan efisien, sehingga dapat meningkatkan produktivitas serta memperkuat penerapan teknologi kecerdasan buatan dalam bidang perkebunan secara praktis dan berkelanjutan.

Berdasarkan hasil pengujian, tingkat akurasi deteksi mencapai 98%, didukung oleh *precision* 94,5% dan *recall* 94,7%, nilai kurva F1-*confidence* sebesar 95%, dan kurva *precision-recall* sebesar 98%. Faktor yang memengaruhi akurasi meliputi kualitas *dataset*, sudut pengambilan citra, pencahayaan, dan luas area objek dalam citra. Tumpang tindih atau terpotongnya objek dalam citra dapat menurunkan kualitas deteksi.

Meskipun sistem deteksi yang dikembangkan menunjukkan kinerja yang sangat baik, penelitian ini masih memiliki beberapa keterbatasan. Model dilatih menggunakan citra statis dengan kondisi pencahayaan dan sudut pengambilan tertentu, sehingga performa deteksi berpotensi menurun pada kondisi lapangan yang lebih dinamis. Selain itu, proses deteksi masih bergantung pada unggahan citra atau video melalui platform web, sehingga belum sepenuhnya mendukung pemrosesan *real-time* secara langsung di lokasi perkebunan. Penelitian selanjutnya disarankan untuk mengembangkan sistem berbasis *real-time* video, mengimplementasikan model pada perangkat *edge* (*edge device*) seperti Raspberry Pi atau NVIDIA Jetson untuk penggunaan langsung di lapangan, serta optimasi model agar lebih ringan dan efisien tanpa mengurangi akurasi, sehingga sistem dapat diterapkan secara lebih luas dalam lingkungan operasional perkebunan kelapa sawit.

Daftar Referensi

- [1] W. E. Sari, Muslimin, A. Franz, and P. Sugiartawan, "Deteksi tingkat kematangan tandan buah segar kelapa sawit dengan k-means," *Sintech J.*, vol. 5, no. 2, pp. 154–164, 2022, doi: 10.31598/sintechjournal.v5i2.1146.
- [2] J. Yu, G. Yusri, Y. Mohamed, and S. Mohamed, "Fresh fruit bunch ripeness classification methods: A review," *Food Bioprocess Technol.*, vol. 18, pp. 183–206, 2025, doi: 10.1007/s11947-024-03483-0.
- [3] Badan Pusat Statistik (BPS) Republik Indonesia, *Statistik Kelapa Sawit Indonesia 2023*. Jakarta: BPS, 2024.
- [4] N. Sari, M. Shiddiq, R. Hayu, and N. Zakyyah, "Klasifikasi tingkat kematangan tandan buah segar kelapa sawit menggunakan probe optik," *J. Aceh Phys. Soc.*, vol. 8, no. 3, pp. 72–77, 2019, doi: 10.24815/jacps.v8i3.14122.
- [5] M. Rifqi and Suharjito, "Deteksi kematangan Tanda Buah Segar (TBS) kelapa sawit berdasarkan komposisi warna menggunakan deep learning," *J. Tek. Inform.*, vol. 14, no. 2, pp. 125–134, 2021, doi: 10.15408/jti.v14i2.23295.
- [6] K. A. Bagaskara and E. Seniwati, "Identifikasi tingkat kematangan buah tomat dengan citra warna," *Inf. Technol. J.*, vol. 5, no. 1, pp. 1–10, 2023, doi: 10.24076/intechnojournal.2023v5i1.1575.
- [7] K. U. Putra, W. Yosfand, and A. Ramadhanu, "Klasifikasi kematangan buah pepaya berdasarkan warna menggunakan convolutional neural network," *J. Ilm. Teknol. Sist.*

- Inf.*, vol. 6, no. 1, pp. 1–6, 2025, doi: 10.62527/jitsi.
- [8] M. Aksa, A. Ranggareksa, A. B. Kaswar, D. Darma, and R. Nurul, “Deteksi tingkat kematangan buah mangga berdasarkan fitur warna menggunakan pengolahan citra digital,” *J. Tek. Inform. dan Sist. Inf.*, vol. 11, no. 2, pp. 240–250, 2025, doi: 10.28932/jutisi.v11i2.10578.
- [9] C. D. Pangati, G. Hoendarto, and Hendro, “Klasifikasi kematangan buah sawit berbasis website menggunakan deep learning,” *J. Widyadharma*, vol. 10, pp. 1–11, 2025.
- [10] A. Rahmawati, M. Akbar, and D. Nurdiansyah, “Klasifikasi tingkat kematangan pada buah kelapa sawit berbasis deep learning,” in *Proceedings Economic, Social Science, Computer, Agriculture and Fisheries (ESCAF) 4th 2025*, 2025, pp. 1064–1073.
- [11] R. Triyogi, R. Magdalena, and B. Hidayat, “Mendeteksi kematangan buah kelapa sawit menggunakan convolution neural network deep learning,” *J. Nas. Sains Tek.*, vol. 1, no. 1, pp. 22–27, 2023, doi: 10.25124/logic.v1i1.6732.
- [12] R. Kurniawan, A. T. Martadinata, and S. D. Cahyo, “Klasifikasi tingkat kematangan buah sawit berbasis deep learning dengan menggunakan arsitektur yolov5,” *J. Inf. Syst. Res.*, vol. 5, no. 1, pp. 302–309, 2023, doi: 10.47065/josh.v5i1.4408.
- [13] S. Aras, P. Tanra, and M. Bazhar, “Deteksi tingkat kematangan buah tomat menggunakan yolov5,” *Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, pp. 623–628, 2024, doi: 10.57152/malcom.v4i2.1270.
- [14] I. G. Khresna, N. P. Sutramiani, and N. K. Ayu Wirdiani, “Network reduction strategy on yolov8 model for mango leaf disease detection,” *J. Buana Inform.*, vol. 16, no. 2, pp. 124–133, 2025.
- [15] M. Shiddiq, L. B. Sitohang, I. R. Husein, and S. A. Ningsih, “Hidung elektronik berbasis sensor gas mos untuk karakterisasi kematangan buah kelapa sawit,” *J. Tek. Pertan. Lampung*, vol. 10, no. 2, pp. 170–182, 2021, doi: 10.23960/jtep-l.v10i2.170-182.
- [16] Syaddam, S. D. Soeksin, and R. Nizar, “Teknologi computer vision untuk melakukan deteksi dan penentuan kualitas bibit ayam day old chicks,” *J. Inform. J. Pengemb. IT*, vol. 9, no. 3, pp. 296–305, 2024, doi: 10.30591/jpit.v9i3.7923.
- [17] J. Subur, Suryadhi, M. Taufiqurrohman, and N. R. Al Hafizh, “Pemanfaatan teknologi computer vision untuk deteksi ukuran ikan bandeng dalam membantu proses sortir ikan,” *J. Tek. Elektro*, vol. 7, no. 01, pp. 52–60, 2024, doi: 10.30651/cl.v7i01.21239.
- [18] D. Pakiding, A. Selao, and Wahyuddin, “Implementasi computer vision dalam mendeteksi penyakit pada tanaman cabai dan tomat menggunakan algoritma convolutional neural networks,” *Indones. J. Mach. Learn. Comput. Sci.*, vol. 5, no. 3, pp. 841–850, 2025, doi: 10.57152/malcom.v5i3.1989.
- [19] O. W. Wijaya, D. R. Manday, A. Napitupulu, M. Turnip, and S. Manurung, “Identifikasi tingkat kematangan buah pada tanaman kelapa sawit menggunakan algoritma convolutional neural network dan pendekatan deep learning,” *J. Tugas Akhir Manaj. Inform. Komputerisasi Akunt.*, vol. 4, no. 2, pp. 232–240, 2024, doi: 10.46880/tamika.Vol4No2(SEMNASTIK).pp232-240.
- [20] I. Budiman, T. Prahasto, and Y. Christyono, “Data clustering menggunakan metodologi CRISP-DM untuk pengenalan pola proporsi pelaksanaan tridharma,” 2012, doi: 10.21456/vol1iss3pp129-134.
- [21] Roboflow, “Palm Oil Fresh Fruit Bunch Ripeness Dataset,” Roboflow Universe, 2024. [Online]. Available: <https://app.roboflow.com/zah/palm-oil-2-1gztp/models>. Accessed: Jun. 2025.
- [22] S. Yulianto, N. F. Amani, F. Akhyar, and K. Usman, “Sistem inspeksi permukaan baja berbasis deep learning menggunakan metode anchor-free,” *J. Ilm. Tek. Mesin, Elektro dan Komput.*, vol. 2, no. 3, pp. 184–190, 2022, doi: 10.51903/juritek.v2i3.364.

- [23] Suriansyah, A. I. Rachman, L. Fanani, A. Halid, and G. Pratiwi, “Peningkatan kinerja database melalui teknik batch loading dan parallel processing pada proses load data,” *J. Ilm. Sist. Inf. dan Tek. Inform.*, vol. 7, no. 1, pp. 146–153, 2024, doi: 10.57093/jisti.v7i1.199.
- [24] R. Kosasih, Sudaryanto, and A. Fahrurrozi, “Classification of six banana ripeness levels based on statistical features on machine learning approach,” *Int. J. Adv. Appl. Sci.*, vol. 12, no. 4, pp. 317–326, 2023, doi: 10.11591/ijaas.v12.i4.pp317-326.
- [25] R. Kosasih and A. Alberto, “Sentiment analysis of game product on shopee using the TF-IDF method and naive bayes classifier,” *Ilk. J. Ilm.*, vol. 13, no. 2, pp. 101–109, 2021, doi: 10.33096/ilkom.v13i2.721.101-109.
- [26] S. Clara, D. L. Prianto, R. Al Habsi, E. F. Lumbantobing, and N. Chamidah, “Implementasi seleksi fitur pada algoritma klasifikasi machine learning untuk prediksi penghasilan pada adult income dataset,” in *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, 2021, April, pp. 741–747.
- [27] P. L. Romadloni, B. A. Kusuma, and W. M. Baihaqi, “Komparasi metode pembelajaran mesin untuk implementasi pengambilan keputusan dalam menentukan promosi jabatan karyawan,” *Jurnal Mhs. Tek. Inform.*, vol. 6, no. 2, pp. 622–628, 2022, doi: 10.36040/jati.v6i2.5238.
- [28] R. Kosasih and M. Arfiansyah, “Pendeteksian kendaraan dengan menggunakan metode running average background subtraction dan morfologi citra,” *J. Media Inform. Budidarma*, vol. 4, no. 4, pp. 979–985, 2020, doi: 10.30865/mib.v4i4.2315.