INSECT: Jurnal Teknik Informatika
INFORMATICS AND SECURITY e-ISSN. 2614-431X

Implementasi Algoritma YOLOv12 Untuk Deteksi Tingkat Kesiapan Panen Kangkung Potong Secara Realtime Berbasis Android

Muhammad Rizki Setyawan[1], Fajar Rahardika Bahari Putra*[2], Suhardi Aras[3], Dimas Adi Suseno[4], Andini Setyaningsi[5]
[1,2,3,4,5]Program Studi Teknik Informatika, Universitas Muhammadiyah Sorong
E-mail: rizki@um-sorong.ac.id[1], fajar_rbp@um-sorong.ac.id[2*], suhardi.aras@um-sorong.ac.id[3]
dimasadisuseno02@gmail.com[4], andiniiiidini19@gmail.com[5]

Abstrak

Penentuan tingkat kesiapan panen kangkung potong umumnya masih dilakukan secara manual melalui pengamatan visual petani yang bersifat subjektif sehingga rentan menimbulkan ketidakkonsistenan dan memengaruhi kualitas serta nilai jual hasil panen. Penelitian ini bertujuan mengimplementasikan algoritma YOLOv12 untuk mendeteksi tingkat kesiapan panen kangkung potong ke dalam dua kelas, yaitu siap panen dan belum panen, secara real-time pada perangkat Android. Dataset terdiri atas 3.556 citra yang dikumpulkan dari repositori publik, kemudian dianotasi, dipra-pemrosesan, dan dibagi dengan proporsi 70:20:10, dengan augmentasi diterapkan hanya pada subset data latih. Model YOLOv12s dilatih selama 100 epoch pada Google Colab, lalu dikonversi ke format ONNX agar dapat dijalankan langsung di perangkat melalui ONNX Runtime. Hasil evaluasi menunjukkan model memperoleh precision 0,841, recall 0,888, mAP50 0,914, dan mAP50-95 0,539, dengan performa terbaik pada kelas siap panen (recall 0,990). Sistem berhasil diimplementasikan menjadi aplikasi Android Kangkung Detector dan diuji menggunakan black box testing yang seluruh fungsinya berjalan sesuai harapan, serta usability testing terhadap 25 responden dengan rata-rata penerimaan 92,08%. Dengan demikian, YOLOv12 berhasil diimplementasikan pada perangkat Android untuk mendeteksi tingkat kesiapan panen kangkung potong dengan akurasi yang baik pada data uji, sehingga berpotensi membantu petani menentukan waktu panen secara lebih objektif dan konsisten.

Kata kunci*— YOLOv12, Deteksi Objek, Kesiapan Panen, Kangkung Potong, Android*

1. PENDAHULUAN

Perkembangan teknologi *computer vision* dan kecerdasan buatan (*artificial intelligence*) telah memberikan kontribusi yang besar di berbagai bidang, termasuk pertanian [1]. Salah satu pendekatan yang banyak digunakan adalah *deep learning*, yang memungkinkan sistem mengenali, mendeteksi, serta mengklasifikasikan objek berdasarkan pola visual dari data citra secara otomatis dan akurat [2]. Dalam ranah deteksi objek, algoritma *You Only Look Once* (YOLO) menjadi salah satu yang paling menonjol karena menggabungkan proses lokalisasi dan klasifikasi ke dalam satu kali proses (*single-stage detection*), sehingga mampu menghasilkan informasi posisi objek (*bounding box*) sekaligus kelasnya secara cepat dan cocok untuk aplikasi *real-time* [3]. Pada bidang pertanian, teknologi berbasis *YOLO* tidak hanya digunakan untuk mendeteksi keberadaan objek, tetapi juga berpotensi mengklasifikasikan kondisi tanaman secara lebih spesifik, seperti tingkat kesegaran maupun kesiapan panen [4], [5], [6].

Kangkung (*Ipomoea reptans Poir*) merupakan salah satu sayuran daun yang banyak dibudidayakan dan dikonsumsi oleh masyarakat Indonesia [7]. Tanaman ini memiliki nilai ekonomis yang cukup tinggi dan mudah dibudidayakan karena mampu tumbuh pada berbagai

kondisi lingkungan [8]. Salah satu jenis yang umum dibudidayakan adalah kangkung potong, yang termasuk kangkung darat dan dipanen berdasarkan tingkat pertumbuhan tertentu dengan masa panen yang relatif singkat, yaitu sekitar 25–30 hari setelah tanam. Karena siklus panennya yang cepat, ketepatan dalam menentukan waktu panen menjadi faktor yang penting, sebab panen yang terlalu cepat maupun terlambat dapat menurunkan kualitas hasil sehingga berpengaruh pada nilai jual dan produktivitas petani [9].

Di Kota Sorong, kangkung termasuk komoditas sayuran dengan tingkat produksi yang cukup tinggi namun fluktuatif. Berdasarkan data Badan Pusat Statistik Kota Sorong, total produksi kangkung menurun dari 9.923 kuintal pada tahun 2022 menjadi 9.195,05 kuintal pada tahun 2023, atau turun sekitar 7,3% [10]. Fluktuasi ini mengindikasikan adanya kendala dalam proses budidaya, khususnya dalam menentukan waktu panen yang optimal. Salah satu penyebabnya adalah penentuan kesiapan panen yang masih dilakukan secara manual melalui pengamatan visual petani yang bersifat subjektif dan bergantung pada pengalaman masing-masing, sehingga standar penentuan waktu panen menjadi tidak seragam dan berpotensi menimbulkan inefisiensi dalam proses produksi.

Penilaian kesiapan panen secara manual tersebut sulit menghasilkan keputusan yang konsisten, terlebih ketika dihadapkan pada kondisi lapangan yang beragam seperti perbedaan pencahayaan dan kepadatan tanaman. Oleh karena itu, diperlukan suatu sistem yang mampu mengidentifikasi tingkat kesiapan panen secara otomatis berdasarkan citra. Pemanfaatan perangkat *Android* menjadi pilihan yang praktis karena hampir setiap petani telah memiliki *smartphone*, sehingga sistem deteksi dapat digunakan langsung di lahan tanpa memerlukan perangkat keras tambahan yang mahal maupun koneksi ke *server* eksternal.

Beberapa penelitian terdahulu telah menerapkan algoritma *YOLO* untuk deteksi objek pada berbagai bidang, namun masih dihadapkan pada tantangan yang mengakibatkan metrik evaluasi yang rendah. Chandrashekhar dkk. mengusulkan kerangka kerja berbasis *YOLOv12* untuk mendeteksi objek berskala sangat kecil dari citra *drone*, namun untuk objek berukuran kurang dari 3 piksel, model *baseline YOLOv12* kesulitan mempertahankan detail spasial sehingga hanya memperoleh *Precision* 63,8%, *Recall* 31,4%, dan *mAP@0.5* sebesar 50,2% [11]. Ramos dan Sappa membandingkan berbagai arsitektur *YOLO* untuk identifikasi penyakit daun tomat dan melaporkan bahwa varian *YOLOv9-T (Tiny)* memberikan hasil paling tidak optimal dengan *Precision* 67,5%, *Recall* 55,0%, dan *mAP@50* 60,4%, terutama karena model tersebut sering gagal mengenali kelas penyakit secara spesifik [12]. Yang dkk. berupaya mengoptimasi YOLOv12 untuk mendeteksi gulma di area pertanian, tetapi evaluasi yang menggunakan *optimizer SGD (Stochastic Gradient Descent)* mencatatkan hasil yang rendah dengan nilai *Precision* hanya 64,0% dan *mAP@0.5* sebesar 72,7% akibat lambatnya konvergensi pada lingkungan agrikultur yang kompleks [13]. Pradana dan Sanjaya ER menerapkan *YOLOv8* untuk mendeteksi buah durian dan manggis menggunakan citra dari internet yang diproses melalui *Roboflow* dan dilatih di *Google Colab*; namun dengan jumlah data yang sangat terbatas, model hanya memperoleh nilai *mAP* yang sangat rendah, yaitu hanya sekitar 0,27 [14]. Sementara itu, Nuha dan Alexandro memanfaatkan *YOLO* untuk pengenalan kesegaran buah mangga, tetapi pada saat menguji mangga segar dan busuk secara bersamaan, model hanya mendapatkan tingkat *accuracy* 73%, *precision* 66%, dan *recall* 81% karena sistem masih mengalami *underfitting* akibat kurangnya variabilitas pada dataset yang digunakan [15].

Berdasarkan kajian tersebut, terlihat bahwa rendahnya metrik evaluasi pada berbagai penelitian sebelumnya tidak semata-mata disebabkan oleh versi algoritma *YOLO* yang digunakan, melainkan lebih banyak dipengaruhi oleh karakteristik *dataset* dan kompleksitas objek, seperti keterbatasan jumlah dan variabilitas data yang memicu *underfitting*, ukuran objek yang sangat kecil, hingga pemilihan *optimizer* yang menyebabkan konvergensi lambat. Selain itu, sebagian besar penelitian masih berfokus pada deteksi atau penghitungan objek tanpa menekankan klasifikasi tingkat kesiapan panen, serta umumnya berhenti pada tahap pengujian model atau

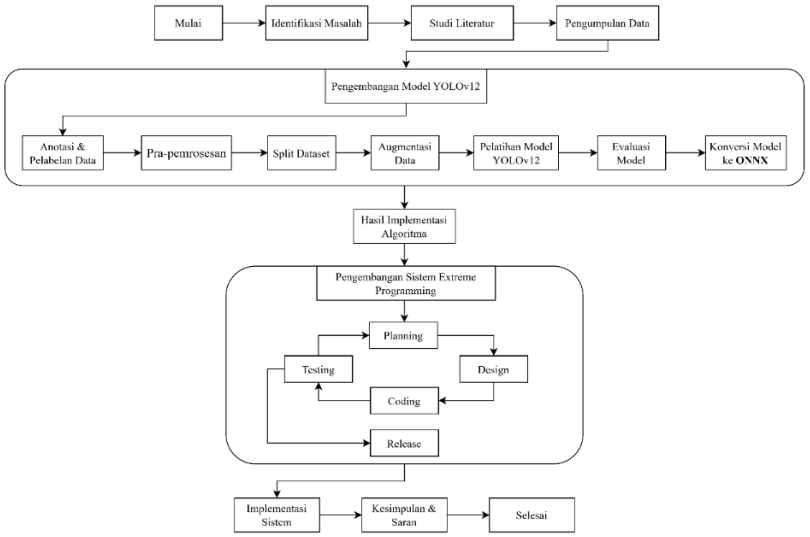
aplikasi berbasis desktop dan web sehingga belum diimplementasikan pada perangkat *Android* secara *real-time*. Penerapannya pada sayuran daun seperti kangkung pun masih sangat terbatas.

Berdasarkan permasalahan dan celah penelitian tersebut, penelitian ini mengangkat judul "Implementasi Algoritma *YOLOv12* untuk Deteksi Tingkat Kesiapan Panen Kangkung Potong Secara *Realtime* Berbasis *Android*". *YOLOv12* dipilih karena merupakan pengembangan terbaru dari keluarga *YOLO* dengan arsitektur *attention-centric* yang dirancang untuk mempertahankan kecepatan deteksi *real-time* sekaligus meningkatkan akurasi. Untuk mengatasi persoalan keterbatasan data yang menjadi penyebab utama rendahnya performa pada penelitian terdahulu, penelitian ini menggunakan dataset kangkung potong yang lebih representatif dan diperkaya melalui augmentasi guna meningkatkan variabilitas data. Penelitian ini bertujuan mengimplementasikan arsitektur *YOLOv12* ke dalam sistem visi komputer untuk mendeteksi tingkat kesiapan panen kangkung potong ke dalam dua kelas, yaitu "siap panen" dan "belum panen", serta mengevaluasi kinerjanya berdasarkan parameter akurasi (*mean Average Precision*) dan kecepatan deteksi. Dengan diimplementasikan pada perangkat *Android*, sistem ini diharapkan dapat membantu petani menentukan waktu panen secara lebih cepat, akurat, dan konsisten sehingga mendukung kestabilan produksi kangkung, khususnya di Kota Sorong.

2. METODE PENELITIAN

2.1 Alur Penelitian

Penelitian ini dilaksanakan melalui beberapa tahapan proses penelitian yang sistematis. Tahap awal dimulai dengan identifikasi masalah untuk memahami persoalan yang diangkat, yaitu penentuan tingkat kesiapan panen kangkung potong yang masih dilakukan secara manual dan subjektif oleh petani. Tahap tersebut dilanjutkan dengan studi literatur untuk memperoleh dasar teori serta penelitian terkait [16], [17], [18], [19], [20]. Selanjutnya dilakukan pengumpulan data sebagai bahan utama penelitian, yang kemudian digunakan pada tahap pengembangan model [21]. Model yang telah dikembangkan dan dievaluasi selanjutnya diimplementasikan ke dalam sistem berbasis Android menggunakan metode Extreme Programming (XP) [22]. XP dipilih karena bersifat iteratif dan adaptif sehingga sesuai untuk pengembangan aplikasi berskala kecil dengan kebutuhan yang dapat berubah selama proses integrasi model. Metode ini terdiri atas empat tahap, yaitu planning (penentuan kebutuhan dan fitur utama aplikasi), design (perancangan flowchart, use case diagram, dan antarmuka), coding (implementasi antarmuka serta integrasi model ONNX ke dalam aplikasi), dan testing (pengujian fungsionalitas dan kegunaan sistem), yang diakhiri dengan tahap release berupa pembuatan berkas APK. Tahap akhir penelitian mencakup penarikan kesimpulan dan pemberian saran untuk pengembangan penelitian selanjutnya. Keseluruhan alur penelitian ditunjukkan pada Gambar 1.



Gambar 1 Alur Penelitian

Tahap pertama adalah mengidentifikasi permasalahan yang menjadi fokus penelitian, yaitu belum adanya sistem yang mampu menentukan tingkat kesiapan panen kangkung potong secara objektif dan *real-time*. Berikutnya, peneliti menelusuri penelitian terdahulu berupa artikel dan jurnal yang berkaitan dengan deteksi objek, *deep learning*, serta penerapan algoritma *YOLO* di bidang pertanian. Setelah itu, peneliti mengumpulkan data berupa citra kangkung potong dari repositori publik seperti *Kaggle* dan *Roboflow* serta beberapa sumber daring lainnya, kemudian menyimpannya untuk diproses pada tahap pengembangan model. Data tersebut digunakan untuk melatih model *YOLOv12* dalam membedakan dua kelas, yaitu "siap panen" dan "belum panen". Model hasil pelatihan kemudian dievaluasi dan dikonversi ke format *ONNX* agar dapat dijalankan pada perangkat *mobile*, lalu diintegrasikan ke dalam aplikasi *Android* yang dikembangkan dengan metode *XP*. Tahap akhir berupa pengujian sistem, penarikan kesimpulan, serta penyusunan saran.

2.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk menentukan perangkat dan sumber daya yang diperlukan dalam proses pengembangan model, implementasi sistem, hingga pengujian aplikasi. Kebutuhan tersebut dikelompokkan menjadi kebutuhan perangkat keras, perangkat lunak, dan *dataset*.

1. Kebutuhan Perangkat Keras
    a. *Laptop MSI Thin 15 B12UCX*
    b. *12th Gen Intel(R) Core (TM) i5-12450H*
    c. *RAM 16GB DDR4*
    d. *NVIDIA GeForce RTX 2050*
    e. *Smartphone Android*
2. Kebutuhan Perangkat Lunak
    a. *Windows 11*
    b. *Android Studio*
    c. *Google Colab*
    d. *Draw.io*
    e. *Figma*
    f. *Roboflow*
    g. *Python (Ultralytics)*
    h. *ONNX Runtime*
    i. *Java*
3. Kebutuhan Dataset
    a. 3.556 citra kangkung potong dalam dua kelas, yaitu "siap panen" dan "belum panen" dalam format *.jpg*
    b. Label dan anotasi *bounding box*

2.3 Pengembangan Model

Pengembangan model difokuskan pada pembentukan model deteksi objek berbasis *YOLOv12* untuk mengenali tingkat kesiapan panen kangkung potong. *Dataset* sebanyak 3.556 citra dengan variasi pencahayaan, sudut pengambilan, dan latar belakang dianotasi menggunakan *bounding box* pada *platform Roboflow*, lalu diberi label dua kelas, yaitu "siap panen" dan "belum panen". Tahap pra-pemrosesan selanjutnya mencakup penyeragaman ukuran citra menjadi 640×640 piksel dan pembersihan gambar buram.

Dataset yang telah dipra-pemrosesan kemudian dibagi secara acak dengan proporsi 70:20:10 yang menghasilkan data latih 2.489 citra, validasi 711 citra, dan uji 356 citra. Setelah pembagian tersebut, augmentasi berupa rotasi, *flipping*, penyesuaian kecerahan, *zoom*, dan kontras diterapkan hanya pada subset data latih untuk menambah variasi data sekaligus mengurangi *overfitting*, sehingga tidak terjadi kebocoran data (*data leakage*) antara data latih, validasi, dan uji.

Setelah pelatihan, model dievaluasi menggunakan *confusion matrix* serta metrik

precision, *recall*, *mean Average Precision* (mAP), dan *Intersection over Union* (IoU), kemudian model terbaik dikonversi ke format *ONNX* agar dapat dijalankan pada aplikasi *Android*.

2.4 Pengembangan Algoritma *YOLO*

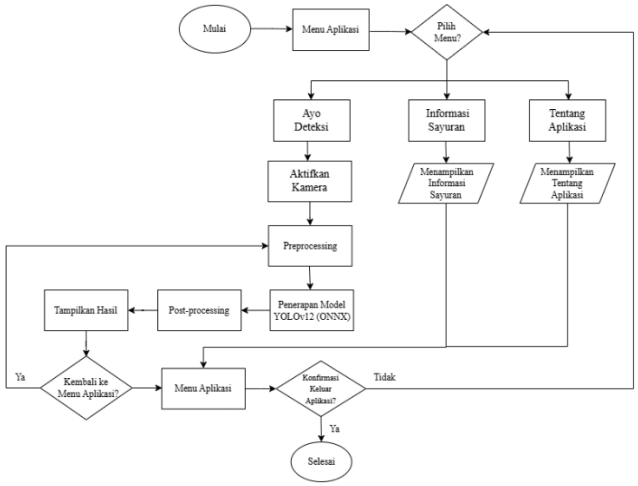
YOLO (You Only Look Once) merupakan kerangka kerja deteksi objek *real-time* yang menggunakan satu jaringan saraf konvolusional tunggal untuk memprediksi *bounding box* dan probabilitas kelas objek dalam satu kali proses maju *(single forward pass)*. Berbeda dengan metode dua tahap (*two-stage*) yang memisahkan proses pengajuan wilayah dan klasifikasi, *YOLO* menyatukan keduanya dalam satu alur terpadu sehingga proses inferensi menjadi lebih cepat dan efisien [23]. Karakteristik ini menjadikan *YOLO* cocok untuk diterapkan pada perangkat dengan sumber daya komputasi terbatas seperti smartphone.

YOLOv12 merupakan pengembangan terbaru keluarga *YOLO* dengan pendekatan *attention-centric* yang memanfaatkan modul *area attention* (A2C2f) dan *cross stage partial* (C3K2) untuk mengoptimalkan ekstraksi fitur tanpa membebani komputasi [24]. *YOLOv12* tersedia dalam lima varian, yaitu *nano*, *small*, *medium*, *large*, dan *extra-large*. Pada penelitian ini digunakan varian *small* (*YOLOv12s*) karena memberikan keseimbangan antara akurasi dan efisiensi yang sesuai untuk deteksi *real-time* pada perangkat *mobile*.

Pelatihan dilakukan pada *Google Colab* dengan *GPU NVIDIA Tesla T4* menggunakan 100 *epoch*, ukuran citra 640×640 piksel, dan *batch size* 16. Pada tahap inferensi, model menghasilkan *bounding box*, label kelas, dan nilai *confidence* yang disempurnakan melalui *Non-Maximum Suppression* (NMS), kemudian dievaluasi menggunakan *precision*, *recall*, *mAP50*, dan *mAP50-95* serta diukur kecepatan inferensinya.

2.5 Pengembangan Sistem

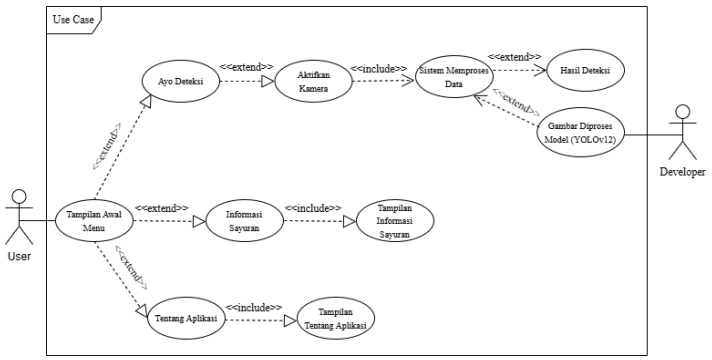
Pengembangan sistem mengikuti metode *Extreme Programming* (XP) yang iteratif dan terdiri atas empat tahap. Pada tahap *planning*, ditentukan kebutuhan dan fitur utama aplikasi, yaitu deteksi tingkat kesiapan panen, informasi sayuran, dan informasi tentang aplikasi. Tahap *design* menerjemahkan kebutuhan tersebut ke dalam *flowchart sistem* yang ditunjukkan pada Gambar 2 *use case diagram* pada Gambar 3, dan rancangan antarmuka yang dibuat menggunakan *Figma*. Pada *flowchart*, proses diawali saat pengguna memilih menu deteksi sehingga sistem mengaktifkan kamera, melakukan *preprocessing*, inferensi model *YOLOv12* berformat *ONNX*, *post-processing*, dan menampilkan hasil deteksi. Adapun *use case diagram* menggambarkan dua aktor, yaitu *user* yang mengakses fitur aplikasi dan *developer* yang menyediakan serta mengelola model deteksi yang digunakan sistem.



Gambar 2 *Flowchart* Sistem

Berdasarkan Gambar 2, proses dimulai ketika pengguna memilih menu deteksi sehingga sistem mengaktifkan kamera perangkat. Setiap *frame* yang diperoleh melalui tahap *preprocessing*

berupa penyesuaian ukuran menjadi 640×640 piksel dan normalisasi nilai piksel, kemudian diteruskan ke model *YOLOv12* berformat *ONNX* untuk proses inferensi. Keluaran model selanjutnya melewati tahap *post-processing* berupa penyaringan nilai *confidence* dan *Non-Maximum Suppression* sebelum ditampilkan kepada pengguna dalam bentuk *bounding box* beserta label kelas. Proses ini berlangsung berulang selama kamera aktif sehingga deteksi dapat ditampilkan secara berkelanjutan.



Gambar 3 *Use Case Diagram*

Gambar 3 memperlihatkan dua aktor yang terlibat dalam sistem. Aktor *user* memiliki akses terhadap tiga fungsi utama, yaitu melakukan deteksi kesiapan panen yang memerlukan pengaktifan kamera dan pemrosesan citra oleh sistem, melihat informasi sayuran, serta melihat informasi tentang aplikasi. Adapun aktor *developer* berperan menyediakan dan mengelola model *YOLOv12* yang digunakan sistem dalam memproses citra, sehingga tidak berinteraksi langsung dengan antarmuka pengguna.

Pada tahap *coding*, antarmuka dan logika aplikasi dibangun dengan bahasa *Java* dan *XML* pada *Android Studio*, dan model *ONNX* diintegrasikan menggunakan *ONNX Runtime* agar inferensi berjalan langsung pada perangkat. Tahap testing dilakukan melalui *black box testing* dan *usability testing*, dan apabila ditemukan kesalahan dilakukan perbaikan pada modul terkait. Tahap akhir, *release*, menghasilkan berkas *APK* yang dapat dijalankan pada perangkat *Android*.

2.6 Metode Pengujian Sistem

Pengujian sistem dilakukan melalui dua pendekatan. Pertama, *black box testing*, yang berfokus pada pengujian fungsionalitas aplikasi tanpa memperhatikan struktur kode, seperti proses membuka aplikasi, akses kamera, deteksi objek, serta penampilan hasil deteksi berupa label dan *bounding box*. Kedua, *usability testing*, yang dilakukan dengan menyebarkan kuesioner kepada sejumlah responden untuk menilai aspek kemudahan penggunaan, tampilan antarmuka, efisiensi, dan kepuasan pengguna. *Persentase* hasil *usability testing* dihitung menggunakan persamaan (1) [25].

$$Y = \frac{X}{Skor\ Ideal} \times 100\% \qquad (1)$$

Keterangan :
Y = Nilai presentase yang dicari (…%)
X = Jumlah nilai kategori jawaban dikalikan dengan frekuensi (Σ = N × R)
N = Nilai dari setiap jawaban
R = Frekuensi jawaban
Skor Ideal = Nilai tertinggi dikalikan jumlah sampel

2.7 Metrik Evaluasi Model

Kinerja model *YOLOv12* dievaluasi menggunakan metrik *precision*, *recall*, *F1-score*, dan *mean Average Precision* (mAP), yang dihitung berdasarkan nilai *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN). *Precision* mengukur ketepatan prediksi positif, *recall*

mengukur kemampuan model menemukan seluruh objek, dan *F1-score* merupakan rata-rata harmonik keduanya, sebagaimana ditunjukkan pada persamaan (2) – (4) [26], [27].

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

Selain itu, digunakan *Intersection over Union* (IoU) untuk mengukur tingkat tumpang tindih antara *bounding box* hasil prediksi dan *bounding box ground truth*, sebagaimana ditunjukkan pada persamaan (5).

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (5)$$

Nilai *mAP* diperoleh dengan menghitung rata-rata *Average Precision* (AP) dari seluruh kelas, sebagaimana ditunjukkan pada persamaan (6). Pada penelitian ini digunakan dua jenis *mAP*, yaitu *mAP50* yang dihitung pada ambang *IoU* sebesar 0,5, serta *mAP50-95* yang merupakan rata-rata *mAP* pada rentang ambang *IoU* 0,5 hingga 0,95.

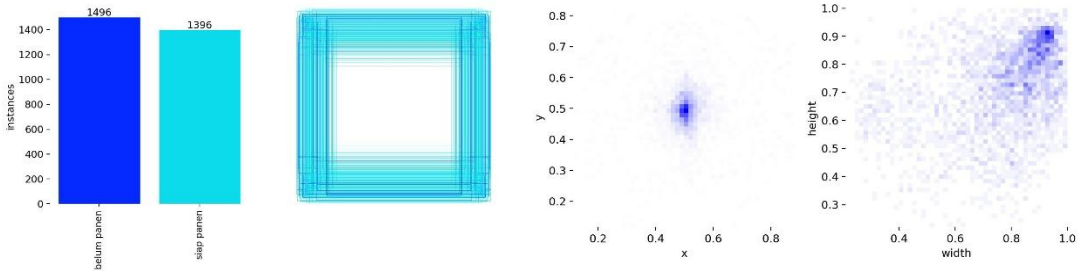
$$mAP = \frac{1}{N} \times \sum_{i=1}^{N} AP_i \quad (6)$$

Dengan N merupakan jumlah kelas dan AP_i adalah nilai *Average Precision* untuk kelas ke-i. Hasil evaluasi ini menjadi dasar untuk menilai kelayakan model sebelum diimplementasikan ke dalam aplikasi *Android*.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengumpulan *Dataset*

Dataset yang berhasil dihimpun berjumlah 3.556 citra kangkung potong berformat *.jpg* dari repositori publik *(Kaggle dan Roboflow)*, terbagi ke dalam dua kelas, yaitu "siap panen" dan "belum panen". Distribusi label pada *dataset* ditunjukkan pada Gambar 4.



Gambar 4 Distribusi Label *Dataset*

Gambar 4 menunjukkan bahwa jumlah *instance* pada kedua kelas relatif seimbang, sehingga model tidak mengalami kecenderungan *class imbalance* yang berarti selama pelatihan. Distribusi posisi dan ukuran *bounding box* memperlihatkan objek tersebar pada berbagai area citra dengan variasi ukuran yang cukup lebar, yang mencerminkan keragaman sudut dan jarak pengambilan gambar pada *dataset*.

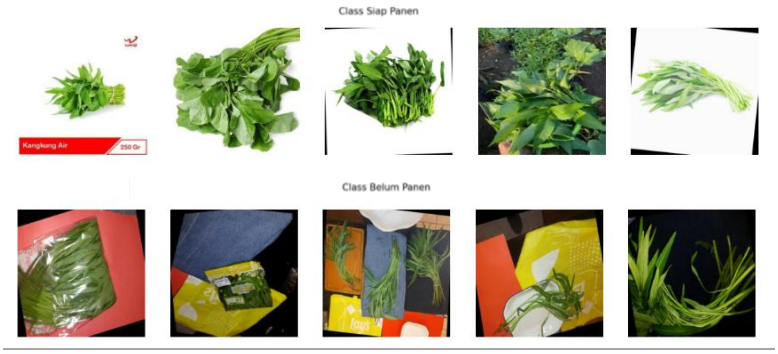
3.2 Implementasi Model *YOLOv12*

Implementasi model *YOLOv12* dilakukan melalui beberapa tahap, yaitu anotasi dan pelabelan, pra-pemrosesan, pembagian *dataset*, augmentasi data latih, pelatihan model, evaluasi, serta konversi model ke format *ONNX*. Seluruh proses pelatihan dilakukan menggunakan *Google Colab* dengan dukungan *GPU NVIDIA Tesla T4* dan pustaka *Ultralytics*.

1. Anotasi dan Pelabelan Data
   Seluruh citra dianotasi menggunakan *bounding box* pada *platform Roboflow*, kemudian diberi label ke dalam dua kelas, yaitu "siap panen" dan "belum panen". Proses anotasi

dilakukan pada citra mentah sebelum tahap pra-pemrosesan agar koordinat *bounding box* mengikuti penyesuaian ukuran citra yang diterapkan kemudian.

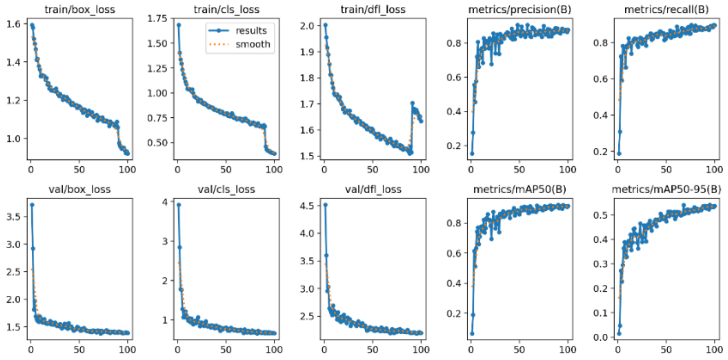
2. Pra-pemrosesan
   Citra yang telah dianotasi diseragamkan ukurannya menjadi 640×640 piksel dan dibersihkan dari gambar buram agar kualitas masukan model konsisten. Contoh citra untuk masing-masing kelas ditunjukkan pada Gambar 5.



Gambar 5 Contoh Kelas "Siap Panen" dan "Belum Panen"

Gambar 5 menampilkan lima contoh citra untuk masing-masing kelas. Pada kelas "siap panen", kangkung ditandai oleh daun yang telah berkembang lebar dengan warna hijau tua, batang yang tampak lebih tebal, serta kerapatan daun yang tinggi. Sebaliknya, citra pada kelas "belum panen" memperlihatkan daun yang masih berukuran kecil dan sempit dengan batang yang lebih langsing serta kerapatan daun yang rendah. Perbedaan karakteristik visual inilah yang menjadi dasar pembelajaran model dalam membedakan kedua kelas. Perlu dicatat bahwa citra pada *dataset* dihimpun dari repositori publik dan sebagian besar menampilkan kangkung dalam kondisi pascapotong dengan latar belakang yang bervariasi, sehingga variasi latar tersebut berpotensi turut memengaruhi proses pembelajaran model.

3. *Split Dataset*
   Dataset dibagi secara acak dengan proporsi 70:20:10 yang menghasilkan data latih 2.489 citra, validasi 711 citra, dan uji 356 citra. Pembagian ini bertujuan agar model dapat dilatih, dipantau performanya, dan diuji pada data yang belum pernah dilihat sebelumnya.
4. Augmentasi Data Latih
   Augmentasi diterapkan melalui rotasi, *flipping*, penyesuaian kecerahan, *zoom*, dan kontras untuk menambah variasi data sekaligus mengurangi *overfitting*. Augmentasi diterapkan hanya pada subset data latih sehingga tidak terjadi kebocoran data (*data leakage*) antara data latih, validasi, dan uji.
5. Pelatihan Model
   Model dilatih menggunakan varian *YOLOv12s* dengan bobot awal *pretrained (yolov12s.pt)* selama 100 *epoch*, ukuran citra 640×640 piksel, dan *batch size* 16 pada *Google Colab* dengan *GPU NVIDIA Tesla T4* (±2 jam 54 menit). Grafik hasil pelatihan ditunjukkan pada Gambar 6, yang memperlihatkan *loss* menurun konsisten serta *precision*, *recall*, dan *mAP* meningkat hingga stabil tanpa indikasi *overfitting* yang berarti.



Gambar 6 Grafik Hasil Pelatihan Model

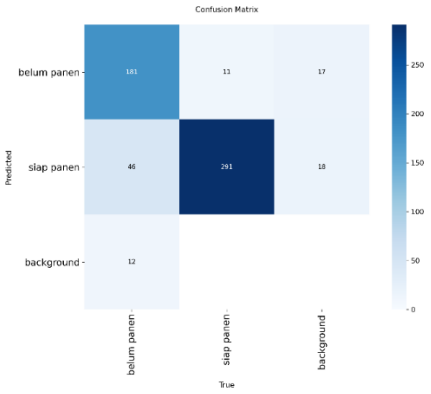
6. Evaluasi Model

Model terbaik (*best.pt*) dievaluasi pada 540 citra validasi (541 *instance*) menggunakan metrik *precision*, *recall*, *mAP50*, *mAP50-95*, dan *F1-score*, dengan hasil keseluruhan dan per kelas pada Tabel 1.

Tabel 1 Hasil Evaluasi Model YOLOv12

Metrik	Keseluruhan	Belum Panen	Siap Panen
Precision	0,841	0,871	0,811
Recall	0,888	0,787	0,990
mAP50	0,914	0,879	0,948
mAP50-95	0,539	0,451	0,626
F1-score	0,864	0,827	0,892

Model memperoleh *mAP50* 0,914 yang menunjukkan akurasi deteksi tinggi pada ambang *IoU* 0,5, sementara *mAP50-95* 0,539 mengindikasikan penurunan performa pada ambang yang lebih ketat. Hal ini umum terjadi pada objek dengan bentuk tidak beraturan seperti dedaunan kangkung. Pada evaluasi per kelas, kelas "siap panen" memperoleh *recall* sangat tinggi (0,990), sedangkan "belum panen" lebih rendah (0,787) sehingga sebagian objek "belum panen" terklasifikasi sebagai "siap panen"; kecenderungan ini perlu diperhatikan karena berpotensi menyebabkan panen sebelum waktunya, dan diduga disebabkan kemiripan visual antara kedua kelas. Pola tersebut diperkuat oleh *confusion matrix* yang ditunjukkan pada Gambar 7 serta hasil uji prediksi ditunjukkan pada Gambar 8 yang menampilkan *bounding box*, label kelas, dan nilai *confidence* tinggi.



Gambar 7 *Confusion Matrix*

Berdasarkan Gambar 7, kesalahan klasifikasi paling banyak terjadi pada objek kelas "belum panen" yang diprediksi sebagai "siap panen". Pola ini sejalan dengan nilai *recall*

kelas "belum panen" yang lebih rendah dibandingkan kelas "siap panen" dan mengindikasikan adanya kemiripan karakteristik visual antara tanaman yang mendekati usia panen dengan tanaman yang telah siap dipanen. Kecenderungan tersebut perlu diperhatikan karena kesalahan ke arah ini berpotensi mendorong pemanenan sebelum waktunya.

Gambar 8 Hasil Uji Prediksi Citra

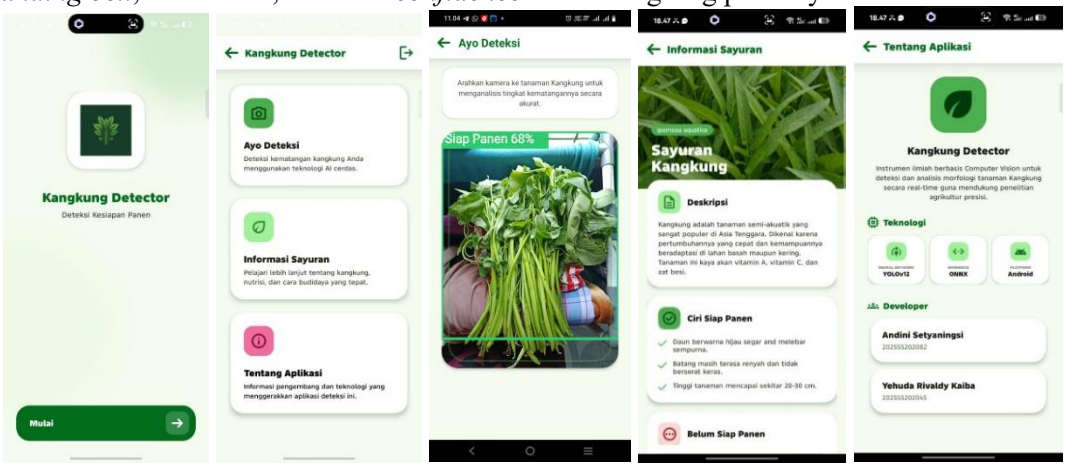
Gambar 8 memperlihatkan model mampu melokalisasi objek kangkung dengan *bounding box* yang tepat serta memberikan label kelas disertai nilai *confidence* yang tinggi pada sebagian besar objek. Hasil ini menunjukkan model telah mempelajari karakteristik visual pembeda antara kedua kelas pada data yang belum pernah dilihat sebelumnya. Kecepatan inferensi diukur dalam dua kondisi, yaitu pada *GPU Tesla T4* (format *PyTorch*) model memerlukan 0,2 *ms preprocessing*, 7,9 *ms inference*, dan 3,9 *ms postprocessing* per citra, sedangkan model *ONNX* pada *CPU (CPUExecutionProvider)* memerlukan 67,4 ms, 752,0 ms, dan 102,3 ms per citra atau setara ±1,3 *FPS*. Perbedaan ini menunjukkan kemampuan *real-time* sangat bergantung pada akselerasi perangkat keras, sehingga optimasi seperti kuantisasi diperlukan untuk perangkat berdaya terbatas.

7. Konversi Model ke ONNX

   Model terbaik (18,7 MB) dikonversi ke format *ONNX* (*opset* 18, 35,1 *MB*) agar dapat dijalankan langsung pada perangkat *Android* melalui *ONNX Runtime* tanpa koneksi ke *server* eksternal.

3.3 Implementasi Sistem

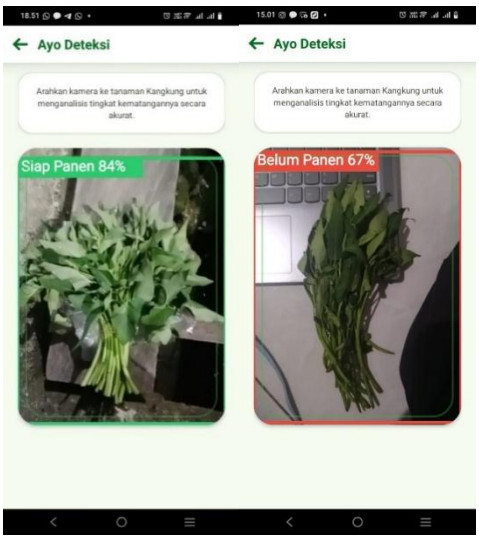
Model *YOLOv12* yang telah dikonversi ke format *ONNX* diintegrasikan ke dalam aplikasi *Android* bernama "Kangkung *Detector*" yang dikembangkan menggunakan *Android Studio*. Aplikasi terdiri atas beberapa halaman utama, yaitu *splash screen*, menu utama, halaman deteksi, halaman informasi sayuran, dan halaman tentang aplikasi, sebagaimana ditunjukkan pada Gambar 9. Pada halaman deteksi, aplikasi mengaktifkan kamera perangkat dan memproses setiap *frame* secara *real-time* menggunakan *ONNX Runtime*, kemudian menampilkan hasil deteksi berupa *bounding box*, label kelas, dan nilai *confidence* secara langsung pada layar.



(a) (b) (c) (d) (e)

Gambar 9 Implementasi Sistem

Aplikasi Kangkung *Detector* terdiri atas beberapa halaman utama. Gambar (a) merupakan tampilan *splash screen* yang menampilkan logo dan nama aplikasi beserta tombol "Mulai" untuk masuk ke menu utama. Gambar (b) merupakan tampilan menu utama yang memuat tiga menu, yaitu "Ayo Deteksi", "Informasi Sayuran", dan "Tentang Aplikasi". Gambar (c) merupakan tampilan halaman deteksi, di mana kamera diarahkan ke tanaman kangkung dan sistem menampilkan hasil deteksi secara *real-time* berupa label kelas dan nilai *confidence* (contoh: "Siap Panen 68%"). Gambar (d) merupakan tampilan halaman informasi sayuran yang memuat deskripsi kangkung serta ciri "Siap Panen" dan "Belum Panen". Gambar (e) merupakan tampilan halaman tentang aplikasi yang berisi penjelasan singkat aplikasi, teknologi yang digunakan (*YOLOv12*, *ONNX*, dan *Android*), serta identitas pengembang.



Gambar 9 Hasil Deteksi *Real-time* pada Perangkat Android

Untuk menguji kemampuan deteksi pada perangkat, dilakukan pengujian terhadap 20 citra uji yang terdiri atas 10 citra kangkung "siap panen" dan 10 citra "belum panen". Hasil pengujian menunjukkan aplikasi berhasil mendeteksi dengan benar 10 dari 10 citra pada kelas "siap panen" dan 9 dari 10 citra pada kelas "belum panen". Contoh hasil deteksi *real-time* pada perangkat *Android* ditunjukkan pada Gambar 10. Kesalahan deteksi yang terjadi diduga disebabkan oleh kondisi pencahayaan, sudut pengambilan gambar, atau kemiripan karakteristik visual antara kedua kelas.

3.4 Pengujian Sistem

Pengujian sistem dilakukan melalui dua pendekatan, yaitu *black box testing* untuk menguji fungsionalitas aplikasi dan *usability testing* untuk menilai pengalaman pengguna.

3.4.1 *Black Box Testing*

Black box testing dilakukan untuk memastikan setiap fitur aplikasi berjalan sesuai dengan fungsi yang dirancang tanpa memperhatikan struktur kode program. Hasil pengujian ditunjukkan pada Tabel 2.

Tabel 2 Hasil *Black Box Testing*

No	Fitur	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	*Splash Screen*	Membuka aplikasi	Splash screen tampil dengan logo "Kangkung *Detector*"	Sesuai Harapan
2	Menu Utama	Membuka halaman utama	Menu utama tampil dengan seluruh menu	Sesuai Harapan
3	Menu Ayo Deteksi	Menekan menu Ayo Deteksi	Halaman deteksi ditampilkan	Sesuai Harapan

4	Akses Kamera	Membuka halaman deteksi	Kamera aktif menampilkan tampilan *real-time*	Sesuai Harapan
5	Deteksi *Real time*	Mengarahkan kamera ke objek kangkung	Sistem melakukan deteksi secara *real-time*	Sesuai Harapan
6	*Bounding Box* & Label	Objek kangkung terdeteksi	*Bounding box*, label kelas, dan *confidence* tampil	Sesuai Harapan
7	Informasi Sayuran	Menekan menu Informasi Sayuran	Halaman informasi sayuran tampil	Sesuai Harapan
8	Tentang Aplikasi	Menekan menu Tentang Aplikasi	Halaman tentang aplikasi tampil	Sesuai Harapan
9	Tombol Kembali	Menekan tombol kembali	Sistem kembali ke menu sebelumnya	Sesuai Harapan

Berdasarkan Tabel 2, seluruh fitur aplikasi Kangkung *Detector* berjalan sesuai dengan fungsi yang dirancang dan memberikan keluaran yang sesuai dengan hasil yang diharapkan.

3.4.2 *Usability Testing*

Untuk mengetahui seberapa puas pengguna terhadap aplikasi yang dikembangkan, peneliti membuat sebuah survei menggunakan *Google Forms*. Survei ini berisi sepuluh pernyataan yang mencakup aspek kemudahan penggunaan, tampilan antarmuka, kemudahan navigasi, kecepatan proses deteksi, kejelasan informasi, serta kepuasan pengguna secara keseluruhan. Cara ini memudahkan peneliti memperoleh masukan dari banyak responden secara cepat dan rapi, sehingga dapat diketahui aspek mana yang sudah baik dan mana yang perlu diperbaiki. Survei disebarkan kepada 25 responden setelah mereka mencoba langsung aplikasi Kangkung *Detector*. Berdasarkan hasil perhitungan, diperoleh rata-rata interpretasi skor sebesar 92,08% dengan keterangan "Sangat Baik", di mana *persentase* tertinggi terdapat pada pernyataan P3 (95,2%) dan terendah pada P9 (88,8%). Dengan demikian, dapat disimpulkan bahwa aplikasi deteksi tingkat kesiapan panen kangkung potong menggunakan algoritma *YOLOv12* berbasis *Android* ini mudah digunakan, dapat diterima dengan baik oleh pengguna, dan telah sesuai dengan tujuan yang ingin dicapai.

4. KESIMPULAN

Berdasarkan hasil penelitian, algoritma *YOLOv12* berhasil diimplementasikan untuk mendeteksi tingkat kesiapan panen kangkung potong ke dalam dua kelas, "siap panen" dan "belum panen". Model *YOLOv12s* yang dilatih selama 100 *epoch* memperoleh *precision* 0,841, *recall* 0,888, *mAP50* 0,914, dan *mAP50-95* 0,539, dengan performa terbaik pada kelas "siap panen" (*recall* 0,990) dan lebih rendah pada "belum panen" (*recall* 0,787). Model dikonversi ke format *ONNX* dan diintegrasikan ke aplikasi Android "Kangkung *Detector*" sehingga deteksi berjalan langsung pada perangkat tanpa *server* eksternal. Pada pengujian, seluruh fitur berfungsi sesuai rancangan (*black box*), deteksi terhadap 20 citra mengenali 19 citra dengan benar, dan *usability testing* terhadap 25 responden memperoleh rata-rata 92,08% (kategori "sangat baik"). Dengan demikian, *YOLOv12* berhasil diimplementasikan pada perangkat *Android* untuk mendeteksi tingkat kesiapan panen kangkung potong dengan akurasi yang baik pada data uji, sehingga berpotensi membantu petani menentukan waktu panen secara lebih objektif dan konsisten.

5. SARAN

Untuk pengembangan penelitian selanjutnya, disarankan agar jumlah data diperbanyak dan diseimbangkan, khususnya pada kelas "belum panen", untuk meningkatkan *recall* kelas tersebut dan menekan kesalahan klasifikasi yang berpotensi menyebabkan panen sebelum waktunya. Selain itu, perlu dilakukan pengukuran kecepatan inferensi berupa latensi dan *frame*

per second secara langsung pada beberapa perangkat *Android* dengan spesifikasi berbeda agar kemampuan deteksi *real-time* pada perangkat dapat dibuktikan secara kuantitatif. Penelitian selanjutnya juga dapat menambahkan variasi tingkat kematangan, misalnya kelas "lewat panen", atau jenis kangkung lain agar sistem mampu mengenali kondisi yang lebih beragam. Terakhir, model perlu diuji dan dioptimalkan pada kondisi lapangan nyata seperti variasi pencahayaan alami, latar belakang kompleks, dan objek yang saling menutupi (*occlusion*), serta mempertimbangkan teknik optimasi seperti *quantization* agar dapat berjalan ringan pada perangkat dengan spesifikasi rendah.

DAFTAR PUSTAKA

[1] A. R. Al-Abbas, Ł. Gierz, B. S. Falih, dan M. A. J. Al-Sammarraie, "Application of you only look once algorithms to crop production management using unmanned aerial vehicles and computer vision systems," *Adv. Sci. Technol. Res. J.*, vol. 20, no. 1, hal. 1–13, 2026, doi: 10.12913/22998624/207678.

[2] N. P. Sari, "Analisis Performa Algoritma CNN dalam Klasifikasi Citra Medis Berbasis Deep Learning," *J. Komput.*, vol. 2, no. 2, hal. 87–92, 2024, doi: 10.70963/jk.v2i2.113.

[3] M. Yan, Z. Sun, Y. Xu, C. Gong, dan C. Kang, "A Review of Lightweight Object Detection Technologies for Densely Occluded Scenarios in Agricultural Fields," *Agronomy*, vol. 16, no. 11, hal. 1–37, 2026, doi: 10.3390/agronomy16111059.

[4] N. Sary, H. Handoko Syahputra Pasaribu, R. Juliana Situmeang, dan R. Darmawan Darus, "Implementasi Algoritma YOLOv11 Dan Roboflow Untuk Deteksi Tingkat Kematangan Anggur Berbasis Web," *Media Teknol. Inf. Dan Komput. J.*, vol. 9, no. 2, hal. 264–274, 2025, doi: 10.47002/metik.v9i2.1121.

[5] S. E. Prasetyo, G. Wijaya, A. Kwan, P. T. Informasi, F. I. Komputer, dan U. I. Batam, "Klasifikasi Kematangan Buah Pisang Menggunakan YOLOv12 Berbasis Deep Learning," *STORAGE – J. Ilm. Tek. dan Ilmu Komput.*, vol. 5, no. 1, hal. 30–39, 2026, doi: 10.55123.

[6] I. U. Haq, F. Felicetti, D. L. Carnì, dan F. Lamonaca, "A Metrologically Guided YOLOv12 Framework with Augmentation-Free Training and Two-Phase Optimization for Multiclass Tomato Leaf Disease Detection," *Appl. Sci.*, vol. 16, no. 5, hal. 1–26, 2026, doi: 10.3390/app16052252.

[7] A. Rahmawati, F. Fadhlilah, Y. K. Fanaba, K. M. Salma, dan D. Indah, "PERTUMBUHAN DAN HASIL TANAMAN KANGKUNG (Ipomoea reptans) PADA PEMBERIAN BEBERAPA KONSENTRASI EKOENZIM," *Fruitset Sains J. Pertan. Agroteknologi*, vol. 12, no. 6, hal. 382–389, 2025, doi: https://doi.org/10.35335/fruitset.v12i6.6106.

[8] C. V. Nanda, V. K. Sari, dan M. N. Khozin, "RESPON PERTUMBUHAN TANAMAN KANGKUNG (Ipomoea reptans Poir) PADA BERBAGAI DOSIS PUPUK NPK," *Agribios*, vol. 20, no. 2, hal. 295, 2022, doi: 10.36841/agribios.v20i2.1943.

[9] Z. Zulkifli, R. Rosnina, K. Khaidir, M. Martina, dan R. Riani, "Budidaya Hidroponik Tanaman Kangkung Dengan Sistem Nft (Nutrient Film Technique) Bagi Masyarakat Desa Lancang Garam Kecamatan Banda Sakti Kota Lhokseumawe," *J. Malikussaleh Mengabdi*, vol. 2, no. 1, hal. 177, 2023, doi: 10.29103/jmm.v2i1.9166.

[10] D. P. K. Sorong, "Produksi Tanaman Sayuran Menurut Distrik dan Jenis Tanaman di Kota Sorong (kuintal)," Badan Pusat Statistik. Diakses: 23 Juni 2026. [Daring]. Tersedia pada: https://sorongkota.bps.go.id/id

[11] A. Chandrashekhar, B. Satyanarayana, R. R. Gorrepati, P. Vasanthi, dan K. L. Prasanna, "An efficient YOLOv12-based framework for detecting extremely small-scale objects," *Sci. Rep.*, vol. 16, no. 1, hal. 1–19, 2026, doi: 10.1038/s41598-025-31803-7.

[12] L. T. Ramos dan A. D. Sappa, "A comprehensive analysis of YOLO architectures for tomato leaf disease identification," *Sci. Rep.*, vol. 15, no. 1, hal. 1–26, 2025, doi: 10.1038/s41598-025-11064-0.

[13] Z. Yang, Z. Khan, Y. Shen, dan H. Liu, "GTDR-YOLOv12: Optimizing YOLO for Efficient and Accurate Weed Detection in Agriculture," *Agronomy*, vol. 15, no. 8, hal. 1–25, 2025, doi: 10.3390/agronomy15081824.

[14] I. P. A. Pradana dan N. A. S. ER, "Implementasi Algoritma Yolo untuk Deteksi Buah Durian dan Manggis," *J. Nas. Teknol. Inf. dan Apl.*, vol. 2, no. 4, hal. 879–886, 2024.

[15] M. S. Nuha dan R. Alexandro H., "Pemanfaatan Yolo untuk Pengenalan Kesegaran Buah Mangga," *Joutica*, vol. 7, no. 1, hal. 513, 2022, doi: 10.30736/jti.v7i1.747.

[16] F. R. B. Putra, A. Fadlil, dan R. Umar, "Application of Forward Chaining Method, Certainty Factor, and Bayes Theorem for Cattle Disease," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 14, no. 1, hal. 365–

374, 2024, doi: 10.18517/ijaseit.14.1.18912.
[17] M. R. Setyawan, F. Rahardika, B. Putra, R. Soekarta, dan Y. I. Manurung, "Deteksi Kesehatan Kucing Menggunakan Arsitektur VGG-16 Berbasis Website," *J. Pustaka Data*, vol. 5, no. 2, hal. 391–397, 2025, doi: https://doi.org/10.55382/jurnalpustakadata.v5i2.1449.
[18] F. Rahardika *dkk.*, "Implementasi Machine Learning Dalam Mengidentifikasi Tanaman Hias menggunakan Metode CNN," *Framew. J. Ilmu Komput. dan Inform.*, vol. 03, no. 01, hal. 1–12, 2024, doi: https://doi.org/10.33506/jiki.v3i1.4138.
[19] M. R. Setyawan, F. R. Bahari Putra, A. Ilham, dan D. A. Suseno, "Detection of Curcuma and Turmeric Differences Utilizing Fuzzy Tsukamoto Android-Based CNN Model," *Ilk. J. Ilm.*, vol. 17, no. 3, hal. 276–291, 2025, doi: 10.30865/jurikom.v12i3.8696.
[20] R. Soekarta, F. Rahardika, B. Putra, S. Aras, M. R. Setyawan, dan A. Ilham, "Fake Reviews Detection in Tokopedia Using Fine-Tuned BERT-Indonesia," *Ingénierie des Systèmes d'Information*, vol. 31, no. 2, hal. 401–409, 2026, doi: https://doi.org/10.18280/isi.310208.
[21] R. B. Fajar Putra, M. Surahmanto, dan Haris, "Implementation of Machine Learning Classification of Obesity Weight using Dicision Tree," *Int. J. Inf. Syst. Technol.*, vol. 8, no. 2, hal. 110–116, 2024, doi: https://doi.org/10.30645/ijistech.v8i2.354.
[22] A. Rokhim dan A. Alimin, "Implementasi Metode Extreme Programming Pada Sistem Pelayanan Pengaduan Masyarakat Berbasis Android," *Spirit*, vol. 15, no. 2, hal. 67–78, 2023, doi: 10.53567/spirit.v15i2.309.
[23] M. El-Geneedy, H. El-Din Moustafa, H. Khater, S. Abd-Elsamee, dan S. A. Gamel, "Advanced real-time detection of acute ischemic stroke using YOLOv12, YOLOv11, and YOLO-NAS: a comparative study for multi-class classification," *Sci. Rep.*, vol. 15, no. 1, hal. 1–15, 2025, doi: 10.1038/s41598-025-18997-6.
[24] T. Yerulan dan Z. Duisebekov, "Comparative Study of Lightweight YOLOv12 Models for Real-Time Underwater Object Detection," *J. Telemat.*, vol. 134, no. 5, hal. 127–137, 2025, doi: 10.32743/unitech.2025.134.5.20045.
[25] A. T. A. Putro, A. Wibowo, dan S. Sutikno, "Evaluasi Usability pada Aplikasi Sistem Pencatatan Pegawai Menggunakan Metode Usability Testing dan USE Questionnaire," *J. Masy. Inform.*, vol. 15, no. 1, hal. 125–148, 2024, doi: 10.14710/jmasif.15.2.67263.
[26] S. Zhao, C. Ye, C. Lian, L. Mei, L. Wu, dan J. Chen, "Fine-Grained Detection and Sorting of Fresh Tea Leaves Using an Enhanced YOLOv12 Framework," *Foods*, vol. 15, no. 3, hal. 1–18, 2026, doi: 10.3390/foods15030544.
[27] M. R. Setyawan, F. R. B. Putra, dan A. Ramadhani, "Sentiment Analysis towards Jokowi Post-Presidential Term Using CNN-BiLSTM with Multi-head Attention on Platform X," *Ilk. J. Ilm.*, vol. 17, no. 2, hal. 150–161, 2025, doi: https://doi.org/10.33096/ilkom.v17i2.2843.150-161.