Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.1148092
Vol. 11, No. 4, Agustus 2024, hlm. 723-734 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

DETEKSI DAN KLASIFIKASI HAMA *POTATO BEETLE* PADA TANAMAN KENTANG MENGGUNAKAN *YOLOV8*

Daniel Geoffrey Manurung[*1], Mohammad Ryan Pinasthika[2], Muhammad Azka Obila Vasya[3], Rania Aprilia Dwi Setya Putri[3], Agustinus Parasian Tampubolon[4], Rakan Fadhil Prayata[5], Septia Khoirin Nisa[6], Novanto Yudistira[7]

[1,2,3,4,5,6,7]Universitas Brawijaya, Malang
Email: [1]danielgeoffrey@student.ub.ac.id, [2]ryanpinasthika@student.ub.ac.id, [3]azkaobila@student.ub.ac.id, [4]raniaapriliadsp@student.ub.ac.id, [5]agustinus1408t@student.ub.ac.id, [5]rakan.fadhil@student.ub.ac.id, [6]khoirinnisa8@student.ub.ac.id, [6]yudistira@ub.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 22 November 2023, diterima untuk diterbitkan: 8 Agustus 2024)

Abstrak

Kentang *(Solanum tuberosum L.)* adalah tanaman pangan penting dengan nilai ekonomi yang tinggi dan menyumbang gizi yang besar bagi manusia. Produksi kentang terhambat oleh serangan penyakit dan hama *potato beetle (Leptinotarsa decemlineata Say)*. Hama ini secara signifikan mempengaruhi hasil panen kentang maka perlu penanganan yang efektif untuk mencegah penurunan produksi yang berkelanjutan. Metode yang digunakan untuk mendeteksi dan mengklasifikasikan *potato beetle* adalah *Convolutional Neural Network* (CNN) yang merupakan algoritma jaringan syaraf tiruan yang efektif dalam pengolahan citra. Model YOLOv8 (*You Only Look Once)* diimplementasikan untuk mendeteksi objek pada gambar dengan mengidentifikasi posisi dan kelas dari *potato beetle*. Data yang digunakan untuk pelatihan adalah menggunakan framework *PyTorch* yang telah dipecah menjadi data *training*, data *validation*, dan data *test*. Hasil penelitian menunjukkan model yang dikembangkan memiliki tingkat akurasi yang memadai dalam mendeteksi dan mengklasifikasikan *potato beetle*. Evaluasi model melibatkan pengukuran *Mean Average Precision* (mAP) dan *F1-Score* berdasarkan konsep *Precision* dan *Recall*. Secara keseluruhan, model ini mencapai mAP50 sebesar 81,8%, yang mengindikasikan tingkat keseluruhan akurasi deteksi yang baik. Namun perlu perbaikan dalam mAP50-95 untuk mengukur akurasi deteksi pada tingkat lebih ketat. Model ini mampu mengklasifikasikan objek dengan *precision* sebesar 78,1% dan *recall* sebesar 89,8%. Dalam evaluasi kelas objek individu, model ini berhasil mendeteksi objek guk (*potato beetle*) dengan tingkat akurasi mencapai *precision* sebesar 88,1% dan *recall* sebesar 90,3%. Sedangkan objek lich juga mendapatkan hasil *precision* 72,8% dan *recall* 76,8%. Namun, perlu diperhatikan bahwa mAP50-95 pada objek lich menunjukkan penurunan yang lebih tinggi dibandingkan dengan objek guk.

Kata kunci: *kentang, potato beetle, CNN, YOLOv8.*

POTATO BEETLE DETECTION AND CLASSIFICATION ON POTATO USING YOLOV8

Abstract

Potato (Solanum tuberosum L.) is a vital food crop with high economic value and a significant source of nutrition for humans. The production of potatoes is hindered by the infestation of the potato beetle (Leptinotarsa decemlineata Say), a pest that significantly impacts potato yields. Effective measures are needed to prevent sustained decreases in production. The Convolutional Neural Network (CNN) algorithm, which is effective in image processing, is utilized for the detection and classification of potato beetles. The YOLOv8 (You Only Look Once) model is implemented to detect objects in images, identifying the position and class of potato beetles. The data used for training is divided into training, validation, and testing datasets using the PyTorch framework. The research results indicate that the developed model achieves an adequate level of accuracy in detecting and classifying potato beetles. Model evaluation involves measurements of Mean Average Precision (mAP) and F1-Score based on Precision and Recall concepts. Overall, the model achieves an mAP50 of 81,8%, indicating a good overall detection accuracy level. However, there is room for improvement in mAP50-95 to measure detection accuracy at stricter levels. This model is capable of classifying objects with a precision of 78,1% and a recall of 89,8%. In the evaluation of individual object classes, the model successfully detects potato beetles with a precision

of 88,1% and a recall of 90,3%. Meanwhile, the "lich" object achieves a precision of 72,8% and a recall of 76,8%. It should be noted that the mAP50-95 for the "lich" object shows a higher decrease compared to the "guk" object.

Keywords: *potato, potato beetle, CNN, YOLOv8*

1. PENDAHULUAN

Kentang adalah salah satu tanaman yang memiliki nilai ekonomi tinggi dan banyak dikonsumsi oleh masyarakat, tidak terbatas pada negara tertentu saja. Kentang juga memiliki sejumlah kandungan gizi yang baik. Seperti tanaman lainnya, kentang juga memiliki permasalahan yang menghambat produksinya, yaitu penyakit dan hama. *Potato beetle* merupakan salah satu hama yang menjangkit tanaman ini adalah. *Potato beetle* adalah hama utama tanaman kentang yang dapat ditemukan di sebagian besar Amerika Serikat.

Hama ini cukup mempengaruhi jumlah produksi kentang. Perlu adanya penanganan yang lebih baik untuk masalah ini. Jika tidak maka penurunan produksi akan terus berlanjut. Salah satu yang cara yang bisa dilakukan adalah pendeteksian lebih awal adanya hama *potato beetle* pada tanaman kentang. Masalahnya, kehadiran penyakit atau hama yang menjangkit tanaman ini masih diidentifikasi secara manual (Prayitna, 2022). Seiring berkembangnya zaman, banyak sekali yang bisa dilakukan oleh komputer secara otomatis, seperti pengidentifikasian citra. Terinspirasi oleh keberhasilan *deep learning* dalam melakukan identifikasi berdasarkan citra digital, penerapan *deep learning* juga akan memberikan dampak yang signifikan dalam mengidentifikasi hama *potato beetle* (Alim, 2020).

Deep learning merupakan bidang pembelajaran mesin (*machine learning*) yang mengimplementasikan metode pembelajaran. Data tersebut dimasukkan pada lapisan-lapisan pembelajaran berkelanjutan yang terus meningkat seiring proses pembelajaran. *Convolutional Neural Network* (CNN) merupakan jenis algoritma jaringan saraf tiruan yang biasa digunakan untuk melakukan klasifikasi maupun pengenalan objek pada data berupa gambar (Abdillah & Rasyad, 2022). Pada prinsipnya, CNN bekerja dengan memanfaatkan proses konvolusi dengan cara menggerakkan filter berukuran tertentu ke sebuah gambar, lalu komputer mendapatkan informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan filter yang digunakan (Lina, 2019). YOLO (*You Only Look Once*) adalah sebuah algoritma deteksi objek yang berbasis *Convolutional Neural Network* (CNN). Algoritma ini dikenal karena kemampuannya untuk mendeteksi objek secara cepat dan akurat dalam gambar atau video dalam satu proses *end-to-end* (Sugandi & Hartono, 2022).

Deep learning sudah terbukti baik dalam melakukan identifikasi hama serangga. CNN telah diimplementasikan dalam mengidentifikasi hama tanaman kedelai dengan hasil yang baik (Gupta, Sharma, & Jain, 2021). Variasi arsitektur ResNet-50 juga berhasil dengan baik mengidentifikasi penyakit daun dan hama serangga pada tanaman apel dengan penerapan *transfer learning* (Zhang dkk, 2023).

Pada penelitian ini diimplementasikan pengolahan citra untuk mendeteksi penyakit pada tanaman kentang menggunakan algoritma YOLO. YOLO memiliki beberapa versi yang berbeda dalam hal performa dan fitur. Versi terbaru dari YOLO adalah YOLOv8, yang dikembangkan oleh Ultralytics. YOLOv8 merupakan model deteksi objek dan segmentasi gambar yang canggih dan mutakhir. YOLOv8 membangun kesuksesan versi-versi sebelumnya dengan memperkenalkan fitur-fitur dan perbaikan baru untuk meningkatkan kinerja dan fleksibilitas. YOLOv8 didesain untuk menjadi cepat, akurat, dan mudah digunakan, sehingga cocok untuk berbagai macam tugas deteksi objek dan pelacakan, segmentasi instans, klasifikasi gambar, dan estimasi pose.

Penelitian ini mengusulkan dan meninjau performa penggunaan YOLOv8 untuk deteksi dan klasifikasi sebagai solusi awal dalam menangani permasalahan hama *potato beetle* pada subjek tanaman kentang. *Dataset* yang digunakan terdiri dari sekumpulan foto *potato beetle* sebagai objek utama dengan background berupa lingkungan sekitar *potato beetle*, yang mencakup variasi tanaman kentang beragam. Kondisi keberagaman tanaman kentang dan lingkungan pada latar belakang visual diharapkan dapat memberikan kondisi yang tepat sehingga model dapat terimplementasi secara universal untuk deteksi hama *potato beetle* dengan latar belakang visual beragam. Penelitian menggunakan *dataset* yang diperoleh dari Kaggle yang kemudian akan dipisah menjadi data *training* dan data *testing*. Model YOLOv8 dilatih menggunakan framework PyTorch dan dievaluasi performanya menggunakan pengukuran *Mean Average Precision* (mAP) dan *F1-Score* yang didasarkan pada konsep *Precision* dan *Recall*.

Meskipun penelitian ini bertujuan memberikan solusi awal dalam deteksi visual hama *potato beetle*, perlu ditekankan pula pentingnya implementasi praktis di lapangan. Perlu dipertimbangkan bahwa metode yang diusulkan akan lebih solutif jika juga diimplementasikan pada perangkat keras. Kendati demikian, fokus dan batas penelitian hanya pada pengembangan model yang dapat dengan efektif mendeteksi hama dalam berbagai kondisi visual. Hasil penelitian dapat memberikan landasan penting untuk pengembangan lebih lanjut, termasuk integrasi dengan perangkat keras khusus untuk aplikasi di lapangan. Selain itu, penelitian ini juga

memperkenalkan implementasi terbaru dari teknologi deteksi objek, YOLOv8, yang merupakan model *state-of-the-art* yang dikembangkan oleh Ultralytics, yang secara signifikan meningkatkan performa deteksi dan klasifikasi objek pada gambar.

2. KAJIAN PUSTAKA

2.1 Kentang dan Potato Beetle

Kentang (*Solanum tuberosum L.*) adalah tanaman pangan yang memiliki nilai gizi tinggi dan lengkap. Tanaman kentang termasuk bahan pangan alternatif pengganti beras. Kentang dapat digolongkan sejenis tanaman herba semusim (*annual crop*) yang kaya akan tepung setelah jagung, gandum, dan padi (Sastrahidayat, 2011).

Sebagai bahan pangan, kentang memiliki kandungan gizi yang dinilai cukup baik, yaitu mengandung protein berkualitas tinggi, asam amino esensial, mineral dan kandungan gizi lainnya. Kentang menyumbang nutrisi penting untuk diet (Beals, 2019). Hal tersebut dapat meningkatkan kebutuhan konsumsi terhadap kentang untuk beberapa tahun kedepan. Dengan adanya peningkatan tersebut, diperlukan pula peningkatan terhadap produksi tanaman kentang. Namun, pertumbuhan tingkat produksi kentang di Indonesia terhambat karena organisme pengganggu tanaman atau hama (Sidauruk, Manalu, & Purba, 2022).

Potato beetle, juga dikenal sebagai *Colorado potato beetle* (*Leptinotarsa decemlineata Say*), adalah hama utama tanaman kentang. Hama ini berasal dari Pegunungan Rocky dan menyebar dengan cepat di tanaman kentang di seluruh Amerika sejak 1859 (Balaško dkk, 2021).

Gambar 1. Colorado potato beetle

Potato beetle memiliki panjang sekitar 10 mm dengan tubuh berwarna kuning cerah/oranye dan lima garis coklat tebal di sepanjang tubuhnya. Telur *potato beetle* berukuran sekitar 1,5mm, mengalami perubahan warna dari kuning setelah oviposisi menjadi oranye untuk telur dewasa yang siap menetas. Hama potato beetle dikenal karena kemampuannya untuk mengembangkan resistensi terhadap insektisida dan pestisida. Tidak jarang infestasi potato beetle sepenuhnya menghancurkan panen tanaman kentang dalam ketiadaan langkah-langkah pengendalian (Alyokhin, Benkovskaya, & Udalov, 2022).

Selain tanaman kentang, *potato beetle* juga menyukai tanaman lain dalam keluarga *nightshade* seperti tomat, paprika, dan terong. Hama ini juga dikenal sebagai hama tanaman dalam keluarga nightshade (Rondon dkk, 2021).

Potato beetle adalah hama yang serius yang dapat merusak tanaman kentang dengan cara memakan daunnya. Serangan oleh hama *potato beetle* yang parah dapat merusak seluruh tanaman kentang. Oleh karena itu, mengendalikan *potato beetle* adalah prioritas bagi petani sayuran karena berbagai tanaman yang dapat diinvestasi oleh hama ini (Wen dkk, 2020).

2.2 CNN

Convolutional Neural Network (CNN) adalah jenis neural network yang umumnya digunakan untuk menganalisis data berupa gambar. CNN sangat efektif dalam mendeteksi dan mengenali objek dalam gambar karena dirancang khusus untuk memproses data dengan struktur topologi *grid*, seperti gambar. Konsep konvolusi digunakan untuk mengekstraksi fitur-fitur penting dari gambar tersebut (Lina, 2019). Metode CNN telah menjadi landasan utama dalam berbagai aplikasi pengenalan gambar dan visi komputer, karena kemampuannya yang unggul dalam memahami hubungan spasial antar piksel dalam gambar (Herlambang, 2020).

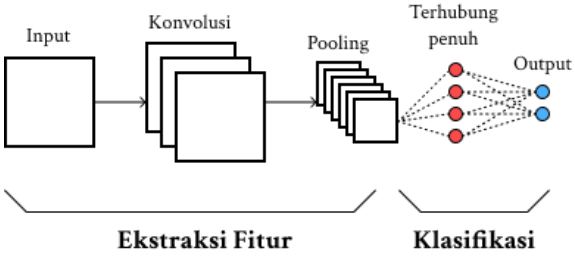
CNN telah menjadi komponen penting dalam bidang visi komputer dan pengolahan citra. CNN pertama kali diperkenalkan pada tahun 1980-an, namun popularitasnya meledak pada tahun 2010-an berkat kemajuan perangkat keras dan perangkat lunak, serta ketersediaan *dataset* yang besar. CNN mendapat nama dari lapisan konvolusi, yang merupakan komponen kunci dalam arsitektur ini.

Selain lapisan konvolusi, CNN seringkali juga mengandung lapisan *pooling*, yang berfungsi untuk mengurangi dimensi data. Lapisan *pooling* membantu mengurangi *overfitting* dan meningkatkan efisiensi komputasi. CNN juga biasanya memiliki lapisan terhubung penuh di bagian akhir untuk mengambil keputusan berdasarkan representasi-fitur yang telah dipelajari sebelumnya.

CNN telah sukses digunakan dalam berbagai aplikasi, termasuk klasifikasi gambar, deteksi objek, pengenalan wajah, analisis medis, dan masih banyak lagi. Kemampuan CNN dalam mengekstraksi fitur-fitur secara otomatis dari data visual, serta kapasitasnya untuk menangani masalah yang besar dan kompleks, menjadikannya salah satu alat kunci dalam pengolahan citra dan visi komputer. Dengan perkembangan yang terus berlanjut, CNN terus menjadi area penelitian yang sangat menarik dalam dunia kecerdasan buatan.

Arsitektur CNN sama dengan jaringan syaraf tiruan (JST) secara tradisional. CNN terdiri atas sekumpulan *neuron* yang menyusun suatu lapisan.

Masing-masing *neuron* akan memiliki bobot, bias, serta fungsi aktivasi. Berbeda dengan JST yang merepresentasikan kumpulan lapisan pada neuron satu dimensi secara vertikal, lapisan pada CNN disusun secara tiga dimensi, sehingga lapisan pada CNN memiliki lebar, tinggi, dan kedalaman. Arsitektur CNN secara umum dapat dilihat pada Gambar 2.



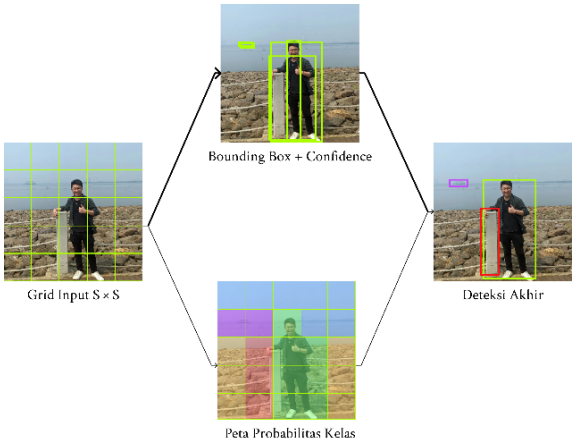
Gambar 2. Arsitektur CNN

Pada CNN, proses klasifikasi dilakukan melalui penggunaan lapisan terhubung penuh, yang merupakan jenis jaringan saraf tiruan (JST) di mana setiap neuron tersusun dalam lapisan dan saling terhubung. Agar dapat diproses dalam lapisan ini, feature map yang dihasilkan pada lapisan konvolusi harus diubah menjadi vektor yang panjang, yang kemudian dimasukkan ke dalam JST untuk proses klasifikasi. Tahap pelatihan dilakukan untuk mencari arsitektur jaringan terbaik dengan melakukan penyetelan pada beberapa *hyperparameter*, termasuk jumlah epoch, jumlah lapisan konvolusi, ukuran *filter*, ukuran *max pooling*, dan nilai *dropout*. Proses ekstraksi fitur dilakukan dengan menggunakan operasi konvolusi dan *pooling* pada lapisan konvolusi dan lapisan *pooling*. lapisan *pooling* digunakan dalam mengurangi dimensi spasial dari *feature map* yang ada. Konfigurasi arsitektur CNN tidak terbatas pada penggunaan satu lapisan konvolusi atau pooling saja, melainkan dapat menggunakan lebih dari dua atau lebih lapisan.

2.3 YOLOv8

YOLO (*You Only Look Once*) adalah sebuah pendekatan baru untuk deteksi objek yang memformulasikan deteksi objek sebagai masalah regresi terhadap *bounding box* dan probabilitas kelas yang terpisah secara spasial. YOLO menggunakan satu jaringan syaraf tunggal yang memprediksi *bounding box* dan probabilitas kelas secara langsung dari gambar penuh dalam satu evaluasi. Seluruh *pipeline* deteksi adalah satu jaringan sehingga dapat dioptimalkan secara *end-to-end* langsung pada kinerja deteksi. YOLO membagi citra input menjadi *grid* dan setiap sel *grid* bertanggung jawab untuk mendeteksi objek di dalamnya. Kelebihan YOLO adalah sangat cepat karena hanya memerlukan satu evaluasi jaringan, melihat seluruh gambar saat pelatihan sehingga mengkodekan informasi kontekstual, dan belajar representasi objek yang sangat umum sehingga baik dalam generalisasi. Kekurangannya adalah masih lemah dalam ketelitian lokalisasi terutama untuk objek kecil, kesulitan menggeneralisasi objek dengan konfigurasi tidak biasa, dan menggunakan fitur yang relatif kasar untuk prediksi *bounding box* (Redmon dkk, 2016).

Algoritma YOLO akan membagi data input menjadi beberapa kotak dan memprediksi setiap *bounding box* serta probabilitas untuk setiap kotak (Aprilino, 2022). Dalam algoritma YOLO, Pertama-tama, sistem YOLO membagi citra input ke dalam *grid* S × S. Jika pusat dari sebuah objek jatuh di dalam salah satu sel *grid*, maka sel *grid* itu bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel kotak yang ada akan memprediksi bounding box beserta *confidence score*-nya. *Confidence score* merupakan persentase tingkat keyakinan dan akurasi dari model bahwa terdapat objek pada kotak tersebut. Setiap prediksi untuk *bounding box* memiliki lima nilai, yaitu *Pc*, *bx*, *by*, *bh*, dan *bw*. Untuk koordinat (*bx*,*by*) merepresentasikan posisi dari pusat *bounding box* relatif terhadap batas sel *grid*, dan koordinat (*bh*,*bw*) merepresentasikan tinggi dan lebar dari *bounding box* relatif terhadap gambar secara keseluruhan. Pc merepresentasikan confidence score dari *bounding box* tersebut. Jika terdapat duplikasi *bounding box*, digunakan *Non-max Suppression* (NMS) untuk menghilangkan duplikat tersebut (Aprilino, 2022). *Non-max Suppression* (NMS) merupakan teknik *post-processing* yang digunakan pada algoritma YOLO untuk mengurangi jumlah duplikat *bounding box* juga untuk meningkatkan kualitas deteksi. Ilustrasi algoritma YOLOv8 dapat dilihat pada Gambar 3.

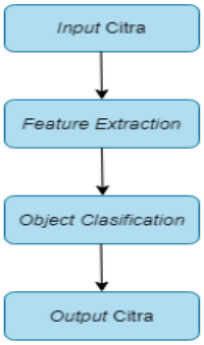


Gambar 3. Ilustrasi algoritma YOLOv8

Mirip dengan CNN, YOLO menggabungkan dua tahap penting dalam proses deteksi objek dalam citra. Tahap pertama, yang dikenal sebagai feature extraction, dilakukan oleh *convolutional lapisan* pada bagian awal jaringan. Pada tahap ini, jaringan memindai citra input dan mengekstraksi fitur-fitur penting seperti tepi, tekstur, dan pola yang membantu dalam mengidentifikasi objek. Selanjutnya, tahap kedua adalah object classification, yang terjadi di convolutional lapisan bagian akhir. Pada tahap ini, YOLO mengklasifikasikan objek yang telah

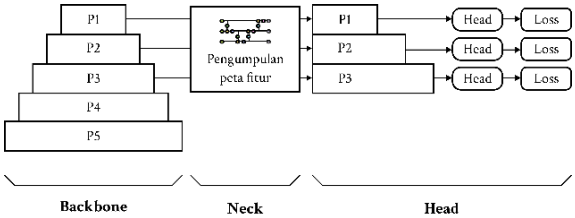
diidentifikasi oleh *feature extraction* ke dalam kelas-kelas yang telah ditentukan, sambil menghitung confidence score untuk setiap kelas. Kemudian, YOLO dapat menggabungkan hasil feature extraction dan object classification untuk mengidentifikasi objek dan menentukan lokasi serta kelasnya dalam gambar.

Sejak diluncurkannya, YOLO telah mengalami berbagai pembaruan versi. YOLOv1 memulai dengan konsep dasarnya namun memiliki kesulitan dalam mendeteksi objek kecil dan yang bertumpuk. YOLOv2 memperkenalkan *anchor boxes* untuk membantu deteksi objek dengan berbagai ukuran, didukung dengan pendekatan *multiscale* dan menggunakan Darknet-19. YOLOv3 lebih lanjut meningkatkan pendekatan multiskala dengan tiga ukuran *anchor box* yang berbeda untuk setiap skala, memungkinkan deteksi yang lebih baik . YOLOv4, dengan CSP Darknet 53, menggabungkan kelebihan versi sebelumnya dan menambahkan *CIOU loss*, memberikan kecepatan dan akurasi yang lebih baik . YOLOv5, dari tim *Ultralytics*, menggunakan CSPNet berbasis *PyTorch* dan *CSPNet* untuk meningkatkan kecepatan serta akurasi, dengan fokus pada augmentasi data dan *tuning hyperparameter*. YOLOv6, dengan *backbone Transformer* bernama YOLOS, merupakan implementasi pertama teknologi *Transformer* dalam deteksi objek, meningkatkan performa pada objek kecil. Selanjutnya, YOLOv7 menggabungkan kelebihan YOLOv5 dan YOLOv6 dengan menggunakan YOLOS dan CSPNet, menawarkan performa yang optimal pada berbagai *dataset*. Lalu yang berikutnya, alur deteksi YOLOv8 dapat dilihat pada Gambar 4.



Gambar 4. Alur Deteksi YOLO

YOLOv8 unggul dibandingkan pendahulunya dalam beberapa aspek utama (Terven & Cordova-Esparza, 2023) YOLOv8 menggunakan jaringan dasar yang lebih cepat dan lebih akurat, yaitu COCO untuk meningkatkan deteksi objek. YOLOv8 memperkenalkan deteksi objek tanpa anchor, meningkatkan prediksi kotak pembatas. Ini lebih efisien berkat feature map yang lebih besar dan jaringan konvolusi yang lebih efisien, sehingga mengurangi waktu pelatihan dan risiko *overfitting*. Selain itu, YOLOv8 menawarkan antarmuka pengguna yang ramah, memudahkan integrasi ke dalam aplikasi, menjadikannya pilihan unggul untuk tugas deteksi objek.



Gambar 5. Arsitektur YOLOv8

Dapat dilihat pada Gambar 5, arsitektur YOLOv8 dapat digambarkan dalam bentuk piramida. Arsitektur YOLOv8 menggunakan beberapa komponen kunci untuk menjalankan tugas deteksi objek. *Backbone* adalah serangkaian lapisan konvolusi yang mengekstrak fitur yang relevan dari gambar masukan. Lapisan SPPF dan lapisan konvolusi berikutnya memproses fitur pada berbagai skala, sementara lapisan *Upsample* meningkatkan resolusi peta fitur. Modul C2f menggabungkan fitur tingkat tinggi dengan informasi kontekstual untuk meningkatkan akurasi deteksi. Terakhir, modul Deteksi yang berada pada komponen head menggunakan serangkaian lapisan konvolusi dan linear untuk memetakan fitur berdimensi tinggi ke kotak pembatas keluaran dan kelas objek. Arsitektur secara keseluruhan dirancang untuk menjadi cepat dan efisien, tetapi tetap mencapai akurasi deteksi yang tinggi. Mengenai legenda diagram, persegi panjang mewakili lapisan, dengan label yang menjelaskan jenis lapisan dan parameter yang relevan. Panah menggambarkan aliran data antara lapisan, dengan arah panah menunjukkan aliran data dari satu lapisan ke lapisan berikutnya.

3. METODE PENELITIAN

Dalam implementasi penelitian kali ini, alur pengerjaan dibagi menjadi tahapan sebagai berikut.

3.1 Analisis kondisi

Pada tahapan ini, dilakukan analisis terhadap kebutuhan apa saja yang harus terpenuhi untuk keberlanjutan penelitian. Gambar-gambar pada *dataset* merepresentasikan gambar latar belakang agraria dengan objek berupa *potato beetle* dalam tahapan larva atau dewasa. Latar memiliki tingkat pencahayaan dan kondisi cuaca yang berbeda, sehingga memiliki keragaman yang dinamis. Resolusi setiap gambar pada *dataset* sudah seragam, yaitu berukuran 1280×720 pixel. Dan terakhir, *dataset* juga dilengkapi dengan karakteristik *bounding box* objek setiap gambar, berupa koordinat titik maksimum dan minimum dari sumbu *x* dan sumbu *y bounding box*.

3.2 Pengumpulan Data

Pada penelitian ini, *dataset* yang digunakan sebagai sumber informasi untuk YOLOv8 adalah

gambar dari larva ataupun kumbang dewasa spesies *colorado potato beetle*. *Dataset* terdiri atas 1810 gambar hama *potato beetle* dengan visual latar belakangnya. Pengumpulan data dilakukan dengan mengambil *dataset* dari situs Kaggle menggunakan perintah API. Setiap citra digital pada *dataset* juga sudah dilengkapi dengan anotasi untuk *bounding box* beserta labelnya, sehingga tidak perlu dilakukan proses anotasi lagi. Gambar 6 adalah beberapa contoh gambar dari *dataset* yang diperoleh, beserta *bounding box*-nya.

Gambar 6. Contoh gambar pada *dataset*

3.3 Implementasi sistem

Pada penelitian ini, deteksi objek akan menggunakan algoritma YOLOv8. Alur kerja sistem akan diawali dengan memasukkan data gambar sebagai input, kemudian dilanjutkan ke tahapan deteksi objek berupa *potato beetle* dalam tahapan larva ataupun dewasa, setelah terdeteksi akan dilanjutkan dengan memotong gambar objek pada bagian objek yang terdeteksi, kemudian dilanjutkan dengan tahap klasifikasi, dan terakhir, akan ditampilkan hasil prediksi klasifikasi dan nilai *confidence* hasil prediksi.

Sistem melakukan *training dataset* pada algoritma YOLOv8 dengan menspesifikasikan beberapa parameter pada fungsi *training*. Dengan memperhatikan kekuatan komputasi sistem, untuk penelitian kali ini, digunakan tiga variasi kompleksitas model, yaitu YOLOv8-n, YOLOv8-s, dan YOLOv8-m dengan banyak *epoch* 10. Pemilihan model YOLOv8 dan banyaknya *epoch* akan menentukan seberapa baik model yang akan dihasilkan melalui proses *training* ini.

Selanjutnya dilakukan Deteksi-*validation* yang mendeskripsikan deteksi yang dilakukan sistem dengan konfigurasi untuk data *validation.* Di mana model yang digunakan diambil dari model terbaik yang didapat setelah melakukan proses training sebelumnya.

Deteksi pada data *testing* mendeskripsikan deteksi yang dilakukan sistem dengan konfigurasi untuk data *test*. Model yang digunakan juga diambil dari model terbaik yang didapat setelah melakukan proses training sebelumnya. Proses ini akan menghasilkan gambar yang telah terprediksi menggunakan model yang dipilih.

Pada penampilan hasil prediksi, sistem akan menampilkan seluruh gambar yang berhasil diprediksi oleh proses deteksi data *testing*. Gambar yang telah terprediksi akan terdapat *bounding box* di sekitar objek yang berhasil dideteksi oleh model disertai hasil prediksi berupa nama kelas dan persentase kemungkinannya.

Gambar 7. Contoh hasil prediksi

Algoritma YOLOv8 bekerja dengan membagi gambar menjadi sel-sel kecil, kemudian memprediksi kelas dan *bounding box* dari setiap sel. *Bounding box* terbaik kemudian dipilih untuk setiap sel, dan kelas yang diprediksi dari *bounding box* tersebut ditambahkan ke daftar prediksi. Daftar prediksi kemudian dikembalikan sebagai hasil dari algoritma. Berikut adalah penjelasan lebih lanjut dari algoritma dalam YOLOv8.

Arsitektur

Arsitektur YOLOv8 terdiri dari tiga bagian utama:

- Backbone

Backbone adalah bagian pertama dari arsitektur YOLOv8, yang bertanggung jawab untuk mengekstrak fitur dari gambar *input*. *Backbone* YOLOv8 menggunakan model ResNet-50, yang telah terbukti sangat efektif untuk tugas-tugas pembelajaran mesin visual.

- Neck

Neck atau leher adalah bagian kedua dari arsitektur YOLOv8, yang bertanggung jawab untuk menghubungkan *backbone* dengan kepala. *Neck* YOLOv8 menggunakan dua buah convolutional block, yaitu FPN (*feature pyramid network*) dan PAN (*path aggregation network*). FPN berfungsi untuk memperluas cakupan visual dari model, sedangkan PAN berfungsi untuk meningkatkan presisi deteksi.

- Head

Head atau kepala adalah bagian ketiga dari arsitektur YOLOv8, yang bertanggung jawab untuk melakukan deteksi objek. *Head* YOLOv8 menggunakan tiga buah *convolutional block*, yang masing-masing bertanggung jawab untuk memprediksi *bounding box*, *confidence*, dan kelas objek.

Anchor-free detection

YOLOv8 menggunakan sistem deteksi *anchor-free*, yang merupakan salah satu peningkatan utama

dari versi sebelumnya. Sistem deteksi *anchor-free* tidak memerlukan *anchor*, yang merupakan kotak persegi panjang yang digunakan untuk memprediksi *bounding box* objek. Dengan tidak menggunakan *anchor*, YOLOv8 dapat lebih akurat dalam memprediksi *bounding box* objek, terutama untuk objek yang berukuran kecil atau tidak beraturan.

Sistem deteksi *anchor-free* bekerja dengan memprediksi *bounding box* objek secara langsung, tanpa menggunakan anchor. Untuk melakukan ini, YOLOv8 menggunakan teknik bernama *IoU-aware bounding box regression*. Teknik ini menggunakan informasi IoU (*Intersection over Union*) antara *bounding box* yang diprediksi dengan *bounding box* milik *ground truth* untuk meningkatkan akurasi prediksi.

Convolutional blocks

YOLOv8 menggunakan *convolutional blocks* yang diperbarui, yang dirancang untuk meningkatkan kinerja dan efisiensi. *Convolutional blocks* baru ini menggunakan teknik-teknik seperti *residual connections*, *group convolutions*, dan *depth wise separable convolutions*.

- Residual connections

  Residual connections adalah teknik yang menghubungkan *output* dari lapisan sebelumnya ke *input* dari lapisan selanjutnya. Teknik ini dapat membantu model untuk belajar lebih baik, terutama untuk tugas-tugas yang kompleks.
- Group convolutions

  Group convolutions adalah teknik yang membagi *filte*r konvolusi menjadi beberapa grup. Teknik ini dapat membantu mengurangi parameter model, sehingga meningkatkan efisiensi.
- Depthwise separable convolutions

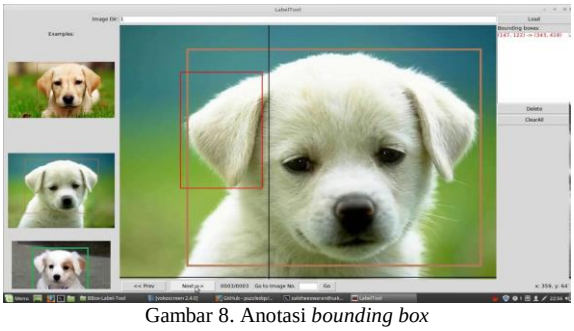
  Depthwise separable convolutions adalah teknik yang membagi filter konvolusi menjadi dua bagian, yaitu *depthwise convolution* dan *pointwise convolution*. Teknik ini dapat membantu meningkatkan kinerja model, terutama untuk tugas-tugas yang membutuhkan kecepatan komputasi yang tinggi.

Mosaic augmentation

YOLOv8 menggunakan teknik augmentasi bernama *mosaic*, yang diterapkan selama pelatihan. *Mosaic* berfungsi untuk menggabungkan empat gambar input menjadi satu gambar, yang dapat membantu model untuk belajar mengenali objek dalam berbagai konteks.

3.4 Training

Pada algoritma YOLOv8, data harus diberi anotasi letak *bounding box* yang mengenkapsulasi objek yang ingin dideteksi. Penerapan anotasi *bounding box* dapat dilihat pada Gambar 8.

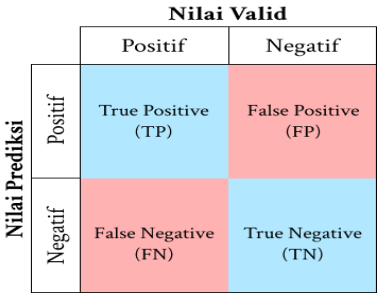


Gambar 8. Anotasi *bounding box*

Dataset akan disusun menjadi tiga kategori data, yakni data *training*, data *validation*, dan data *testing*. Dari *dataset*, terdapat total 1810 gambar, dengan perbandingan pembagian sebanyak 1267 gambar untuk data *training*, 362 gambar untuk data *validation*, dan 181 gambar untuk data *testing*. Data *training* akan difungsikan untuk melatih model dalam tugas deteksi dan klasifikasi objek, sementara data *validation* akan digunakan untuk mengevaluasi kemampuan pembelajaran model serta mencegah *overfitting*. Selain itu, data *validation* akan berperan sebagai alat ukur seberapa baik model berkinerja dalam tahap belajar. Terakhir, data *testing* akan digunakan untuk mengukur metrik akhir dari model terhadap data yang sebelumnya belum pernah dilihat, memberikan gambaran lengkap tentang performa model pada situasi yang sesungguhnya.

3.5 Uji Coba dan Evaluasi Sistem

Karena sistem melakukan deteksi dan klasifikasi, diterapkan dua metode evaluasi kinerja sistem. Pengujian akan hasil deteksi dievaluasi menggunakan mAP (*mean average precision*), sedangkan hasil klasifikasi dievaluasi menggunakan *F1-Score*.



Gambar 9. *Confusion matrix*

Kedua metode evaluasi tadi berbasis pada konsep yang sama, yaitu *confusion matrix*. Berbasis pada hasil prediksi berbanding dengan hasil valid, *confusion matrix* merupakan tabel dengan empat kombinasi antara nilai prediksi dengan nilai valid.

Nilai mAP didapatkan dari rata-rata hasil interpolasi jumlah titik koordinat untuk (*recall*, *precision*) bagi setiap kelas. *Recall* merupakan proporsi nilai positif valid yang diprediksi dengan

benar, dan didefinisikan dengan membagi TP dengan penjumlahan TP dan FN(1).

$$Recall = \frac{TP}{TP+FN} \quad (1)$$

Precision merupakan proporsi prediksi positif yang benar, didefinisikan dengan membagi TP dengan penjumlahan TP dan FP(2).

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

Nilai *precision* dihitung dengan cara membagi total sampel positif yang diklasifikasikan dengan benar dengan total sampel positif yang diprediksi, sedangkan *recall* adalah rasio dari total keseluruhan sampel positif yang diklasifikasikan dengan benar kemudian dibagi dengan total sampel positif.

Nilai *precision* dan *recall* setiap data untuk masing-masing kelas kemudian diinterpolasikan menjadi titik-titik koordinat dengan *precision* pada sumbu Y. Kemudian, nilai Y pada seluruh titik untuk suatu kelas akan dijumlahkan dan dirata-ratakan dengan total data pada kelas tersebut (3) untuk memperoleh *average precision* (AP).

$$Average\ Precision = \frac{1}{n}\sum_{k=0}^{k=n} interpolate_k \quad (3)$$

mAP untuk evaluasi deteksi objek kemudian dapat ditentukan dengan mencari rata-rata AP dari seluruh kelas yang ada, sehingga dapat didefinisikan dengan membagi penjumlahan AP untuk seluruh kelas dengan jumlah kelas yang ada(4).

$$Mean\ Average\ Precision = \frac{1}{n}\sum_{k=0}^{k=n} AP_k \quad (4)$$

Definisi dasar mAP memiliki variasi lain berdasarkan nilai IoU (*intersection over union*) gambar. Metrik IoU mengkodekan sifat-sifat bentuk dari objek yang dibandingkan, misalnya lebar, tinggi, dan lokasi dua kotak pembatas, ke dalam properti wilayah dan kemudian menghitung ukuran yang dinormalisasi yang berfokus pada area (atau volume).

mAP dapat divariasikan berdasarkan *threshold* nilai IoU dari input yang diberikan pada model. Penelitian menggunakan variasi mAP50 dan mAP50-95 dalam metrik penilaian model.

mAP50 adalah mAP yang dihitung pada ambang IoU sebesar 0,50 (5). Metrik digunakan dalam menilai metrik mAP model yang hanya mempertimbangkan deteksi "mudah".

$$mAP50 = \frac{1}{n}\sum_{k=0}^{k=n} AP_{50}^{k} \quad (5)$$

mAP50−95 adalah mAP yang dihitung pada berbagai ambang IoU, mulai dari 0,50 hingga 0,95 (6).Metrik memberikan pandangan komprehensif tentang kinerja model pada berbagai tingkat kesulitan deteksi.

$$mAP50-95 = \frac{1}{n}\sum_{k=0}^{k=n} AP_{50-95}^{k} \quad (6)$$

Klasifikasi objek dievaluasi menggunakan *F1-Score* yang menilai kemampuan prediksi klasifikasi model berdasarkan performanya untuk seluruh kelas yang ada. *F1-Score* dapat didefinisikan dengan mencari *harmonic mean* dari nilai *precision* dan *recall*(7).

$$F1-Score = \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}} = \frac{2\cdot Precision\cdot Recall}{Precision+Recall} \quad (7)$$

F1-Score adalah salah satu metrik evaluasi klasifikasi yang digunakan untuk mengukur performa model dalam memprediksi kelas yang benar

Pada YOLOv8, juga dikenal Loss beserta fungsinya yaitu loss function. Untuk YOLOv8, fungsi *loss*-nya terdiri dari tiga komponen, yaitu *classification loss*, *localization loss*, dan *confidence loss*. Jika suatu objek terdeteksi, *classification loss* di setiap sel adalah kuadrat *loss* dari probabilitas kondisional kelas untuk setiap kelas (8).

$$\sum_{i=0}^{s^2} \mathbb{1}_i^{obj} \sum_{c\,\in\,kelas}\left(p_i(c) - \hat{p}_i(c)\right)^2 \quad (8)$$

Pada persamaan 8 untuk *classification loss*, variabel $1_{i\,j}^{obj}$ merepresentasikan nilai 1 apabila sebuah objek berada pada threshold *boundary box* untuk, jika tidak maka 0. Kemudian $\hat{P}_i(c)$ menunjukkan probabilitas kelas bersyarat untuk kelas *c* pada sel *i*..

Localization loss mengukur kesalahan dalam lokasi dan ukuran kotak pembatas yang diprediksi (9). Hanya dihitung kotak yang bertanggung jawab untuk mendeteksi objek.

Pada persamaan 9, variabel $1_{i\,j}^{obj}$ merepresentasikan nilai 1 apabila sebuah objek berada pada threshold *boundary box* untuk, jika tidak maka 0. Kemudian λ_{coord} akan menambahkan bobot *loss* pada koordinat *boundary box*.

$$\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right] + \lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[\left(\sqrt{x_i}-\sqrt{\hat{x}_i}\right)^2+\left(y_i-\sqrt{\hat{y}}_i\right)^2\right] \quad (9)$$

Untuk *confidence loss*, Jika suatu objek terdeteksi di dalam kotak, didefinisikan sebagai kuadrat loss dari nilai *confidence* objek.

$$\sum_{i=0}^{s^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2 \quad (10)$$

Pada persamaan (10), Variable $\hat{C}_i$ mendefinisikan nilai *confidence* dari box *j* pada sel *i*. Kemudian, variabel $1_{i\,j}^{obj}$ merepresentasikan nilai 1 apabila sebuah objek berada pada threshold *boundary box* untuk, jika tidak maka 0.

Ketiga *loss function* tersebut akan digabungkan menjadi satu definisi *loss function* utuh untuk YOLOv8 (11).

$$Loss = \lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right] + \lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[\left(\sqrt{x_i}-\sqrt{\hat{x}_i}\right)^2+\left(y_i-\right.\right.$$

$$\sqrt{\hat{y}_i}\Big)^2\Big] + \sum_{i=0}^{s^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left(C_i - \hat{C}_i\right)^2 + \sum_{i=0}^{s^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{noobj}\left(C_i - \hat{C}_i\right)^2 + \sum_{i=0}^{s^2}\mathbb{1}_{i}^{obj}\sum_{c\,\in\,kelas}\left(p_i(c) - \hat{p}_i(c)\right)^2 \quad (11)$$

Pada pengujian kali ini, model diklasifikasikan ke dalam tiga kelas, yaitu *guk*, *lich* dan *background*. Kelas *guk* menggambarkan objek *potato beetle*, kelas *lich* menggambarkan objek larva dan kelas background menggambarkan objek yang tidak termasuk kelas *guk* atau *lich*. Pada pengujian, digunakan tiga variasi arsitektur YOLOv8 berdasarkan ukurannya, yaitu *nano* (YOLOv8-n), *small* (YOLOv8-s), dan *medium* (YOLOv8-m).

Dalam konteks penelitian yang dilakukan, nilai *confusion matrix* yang terletak pada titik horizontal dan vertikal yang sama merupakan nilai *True Positive* (TP) dari suatu label. Nilai yang terletak dalam satu baris yang sama dengan *True Positive* dari suatu label, merupakan nilai *False Positive* (FP) dari suatu label. Nilai yang terletak dalam satu kolom yang sama dengan *True Positive* suatu label, merupakan nilai *False Negative* (FN) dari suatu label. Lalu nilai yang terletak dalam satu diagonal yang sama dengan *True Positive* dari suatu label merupakan nilai *True Negative* (TN) dari suatu label. C*onfusion matrix* menggunakan tingkat kecerahan yang menandai banyaknya data. Semakin gelap, maka semakin banyak data.

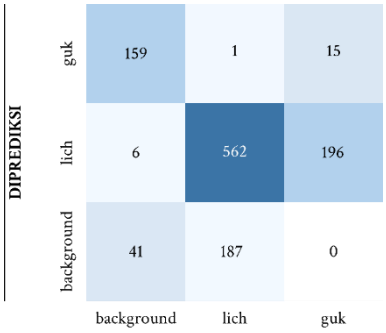
Pengujian model akan dilakukan terhadap tiga variasi model YOLOv8, yaitu YOLOv8-n, YOLOv8-s, dan YOLOv8-m. Ketiga variasi ini berbeda dengan satu sama lain secara kompleksitas kedalaman dan lebar arsitekturnya. Pada implementasinya, tingkat kompleksitas setiap model akan berkaitan langsung dengan jumlah parameter yang ada pada model.

Tabel 1. Hasil evaluasi YOLOv8

Model	Kelas	Recall	Precision	mAP50	mAP50-95
YOLOv8-n	all	74,8%	82,2%	78,3%	36,3%
	guk	77,2%	90,8%	84,1%	45,7%
	lich	72,3%	73,6%	72,6%	27%
YOLOv8-s	all	83,3%	75,4%	81,8%	38,9%
	guk	89,8%	78,1%	88,7%	48%
	lich	76,8%	72,8%	74,9%	29,8%
YOLOv8-m	all	76,1%	88,2%	82,5%	39,7%
	guk	76,7%	95,2%	88,4%	48,3%
	lich	75,5%	81,3%	76,7%	31,1%

Pada pengujian model YOLOv8-n, dilakukan pelatihan dengan 10 *epoch* dan total parameter mencapai 3.006.038 parameter. Secara keseluruhan, model mencapai *recall* 74,8% dan *precision* 82,2%, yang mengidentifikasi kemampuan model dalam mendeteksi objek dengan akurasi yang baik. Nilai *mean average precision* untuk mAP50 secara keseluruhan 78,3% dan mAP50-95 36,3%%.

Kemudian, evaluasi kelas dengan objek *guk* mencapai *precision* 90,8%, *recall* 77,2%, mAP50 84,1%, mAP50-95 45,7%. Untuk kelas *lich* mencapai *precision* 73,6%, *recall* 72,3%, mAP50 72,6%, mAP50-95 sekitar 27%. Hasil pengujian model YOLOv8-n dapat digambarkan dengan *confusion matrix* pada Gambar 10.

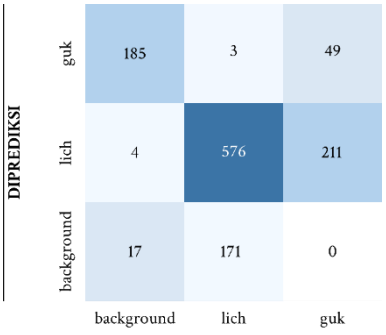


Gambar 10. Confusion matrix YOLOv8-n

Pada pengujian model YOLOv8-s, dilakukan pelatihan dengan 10 *epoch* dan total parameter mencapai 11.126.358 parameter. Secara keseluruhan, model mencapai *recall* 83,3% dan *precision* 75,4%, yang mengidentifikasi kemampuan model dalam mendeteksi objek dengan akurasi yang baik. Nilai *mean average precision* untuk mAP50 secara keseluruhan 81,8% dan mAP50-95 38,9%.

Kemudian, evaluasi kelas dengan objek *guk* mencapai *precision* 78,1%, *recall* 89,8%, mAP50 88,7%, mAP50-95 48%. Untuk kelas *lich* mencapai *precision* 72,8%, *recall* 76,8%, mAP50 74,9%, mAP50-95 29,8%. Hasil pengujian model YOLOv8-s dapat digambarkan dengan *confusion matrix* pada Gambar 11.

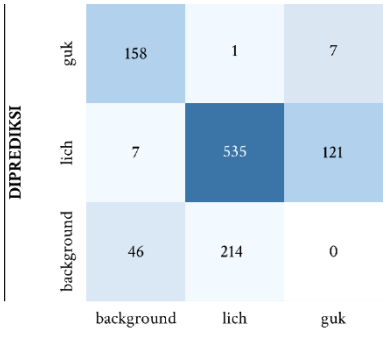
Terakhir pada pengujian model YOLOv8-m, dilakukan pelatihan dengan 10 *epoch* dan total parameter mencapai 25.840.918 parameter. Secara keseluruhan, model mencapai *recall* 76,1% dan *precision* 88,2%, yang mengidentifikasi kemampuan model dalam mendeteksi objek dengan akurasi yang baik. Nilai *mean average precision* untuk mAP50 secara keseluruhan 82,5% dan mAP50-95 39,7%.



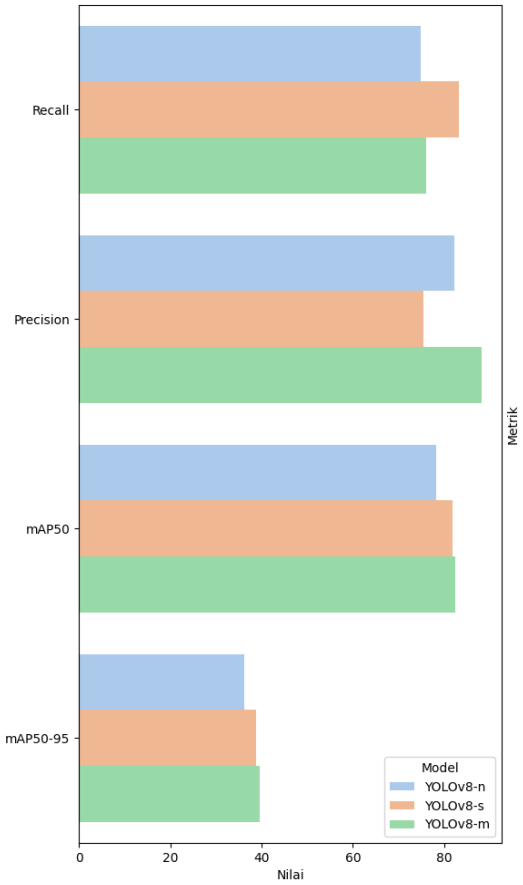
Gambar 11. Confusion matrix YOLOv8-s

Kemudian, evaluasi kelas dengan objek *guk* mencapai *precision* 95,2%, *recall* 76,7%, mAP50 88,4%, mAP50-95 48,3%. Untuk kelas *lich* mencapai *precision* 81,3%, *recall* 75,5%, mAP50 76,7%, mAP50-95 sekitar 31,1%. Hasil pengujian model

YOLOv8-m dapat digambarkan dengan *confusion matrix* pada Gambar 12.



Gambar 12. Confusion matrix YOLOv8-m



Gambar 13. Perbandingan model YOLOv8

Berdasarkan metrik-metrik yang ada, model terbaik adalah YOLOv8-m. Model ini mencapai tingkat precision tertinggi (88,2%), menunjukkan kemampuannya dalam mengidentifikasi objek dengan sangat tepat. Selain itu, YOLOv8-m juga memiliki nilai *recall* yang cukup baik (76,7%), menunjukkan kemampuannya dalam mendeteksi sebagian besar objek yang seharusnya diidentifikasi. Selain itu, nilai *mean average precision* (mAP50) secara keseluruhan juga tertinggi (82,5%), menunjukkan konsistensi model dalam memberikan prediksi yang baik.

Hasil evaluasi kelas objek *guk* dan *lich* menunjukkan bahwa YOLOv8-m memiliki performa yang secara keseluruhan lebih konsisten dan unggul dibandingkan dengan model lainnya pada Gambar 13. Meskipun demikian, ditemukan bahwa performa YOLOv8-m terkait dengan metrik mAP50-95 masih berada pada tingkat yang rendah. Oleh karena itu, penelitian selanjutnya dapat memperbaiki hasil ini dengan melakukan *fine tuning* pada model tersebut untuk mencapai hasil yang lebih baik.

4. KESIMPULAN

Dalam implementasi penelitian kali ini, alur pengerjaan dibagi menjadi tahapan yang mencangkup analisis kondisi, implementasi sistem, pelatihan, uji coba, dan evaluasi. Dalam pengujian, dilakukan komparasi antara tiga variasi model YOLOv8 dalam deteksi *potato beetle*. Dari ketiga variasi model YOLOv8, didapatkan bahwa model YOLOv8-m memberikan hasil terbaik dengan hasil keseluruhan setiap kelas untuk *recall* 76,1%, untuk *precision* 95,2%, untuk mAP-50 82,5%, dan untuk mAP50-95 39,7%. Secara keseluruhan, penelitian ini telah menghasilkan model yang mampu mendeteksi dan mengklasifikasikan objek dengan baik, dengan potensi perbaikan pada evaluasi kelas objek yang lebih ketat.

Untuk penelitian berikutnya, disarankan untuk meningkatkan jumlah parameter *epoch* guna memaksimalkan kemampuan model dalam mempelajari *dataset* yang diberikan. Selain itu, terdapat ruang untuk melakukan pengujian terhadap variasi YOLOv8 lain dengan tingkat kompleksitas arsitektur yang lebih tinggi, sehingga memungkinkan penilaian yang lebih komprehensif terhadap performa dan kehandalan model. Selain itu, dalam implementasi praktis pada perangkat, penelitian selanjutnya juga dapat mempertimbangkan *deployment model* untuk menangani permasalahan di lapangan secara efektif.

DAFTAR PUSTAKA

PRAYITNA, D.S., 2022. Deteksi Penyakit Daun Tomat dengan Algoritma You Only Look Once (YOLO). Bandung.

ALIM, M.M.F., 2020. Identifikasi Penyakit Tanaman Tomat menggunakan Algoritma Convolutional Neural Network dan Pendekatan Transfer Learning. Semarang.

ABDILLAH, M., RASYAD, S. dan ALFARIZAL, N., 2022. Implementasi Sistem Pendeteksi Penggunaan Masker Berbasis Raspberry Pi 4 Menggunakan Metode Convolution Neural Network (CNN) pada Proses Screening Protokol Kesehatan Covid 19. Jurnal Teknika, 16(1), pp.9-15.

LINA, Q., 2019. apa itu Convolutional Neural Network?. Medium, [Daring] Tersedia di: <https://medium.com/@16611110/apa-

ituconvolutional-neural-network-836f70b193a4> [Diakses 17 September 2023].

SUGANDI, A.N dan HARTONO, B., 2022. Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO, 13(1), pp. 183-188. Prosiding The 13th Industrial Workshop and National Seminar. Bandung, 13-14 Juli 2022. https://doi.org/10.35313/irwns.v13i01.4186

APRILINO, A., 2022. Implementasi Algoritma YOLO dan Tesseract OCR pada Sistem Deteksi Plat Nomor Otomatis. Jurnal Teknoinfo, 16(1), p.54. https://doi.org/10.33365/jti.v16i1.1522.

TERVEN, J. dan CORDOVA-ESPARZA, D.A., 2023. A Comprehensive review of YOLO: From YOLOv1 and beyond. arXiv preprint arXiv:1709.08439.

BALAŠKO, M.K., MIKAC, K.M., BAŽOK, R. dan LEMIC, D. 2021. Modern Techniques in Colorado Potato Beetle (Leptinotarsa decemlienata Say) Control and Resistance Management: History Review and Future Perpectives. Insects, 12(1), p.43. https://doi.org/10.3390/insects11090581.

WEN, G., KHELIFI, M., CAMBOURIS, A.N. dan ZIADI, N., 2020. Responses of the Colorado Potato Beetle (Coleoptera: Chrysomelidae) to the Chemical Composition of Potato Plant Foliage. Journal of Economic Entomology, 113(5), pp. 2278-2286. https://doi.org/10.1007/s11540-018-9405-0.

HERLAMBANG, M.F., 2020. Pengenalan Karakter Huruf Braille dengan Metode Convolutional Neural Network. Bandung.

REDMON, J., DIVVALA, S., GIRSHICK, R. dan FARHADI, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. [Daring] Washington: University of Washington, Allen Institute for AI, Facebook AI Research. Tersedia di https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf> [Diakses 17 September 2023].

ALYOKHIN, A., BENKOVSKAYA, G. dan UDALOV, M., 2022. Chapter 4 - Colorado potato beetle [Daring] Insect Pests of Potato: Global Perspective on Biology and Management, 2nd ed, pp.29-42. Cambridge: Academic Press.

BEALS, K.A., 2019. Potatoes, nutrition and health. American journal of potato research, 96(2), 102-100. https://doi.org/10.1007/s12230-018-09705-4.

SIDAURUK, L. MANALU, C.J.F. dan PURBA, T.H., 2022. Pengaruh Jenis dan Konsentrasi Pestisida Nabati Berbasis Lokal terhadap Persentasi Serangan Hama dan Produksi Kentang (Solanum tuberosum L.). Majalah Ilmiah METHODA, 12(2), 125-132. https://doi.org/10.46880/methoda.Vol12No2.pp125-132.

RONDON, S., FELDMAN, M., THOMPSON, A., OPPEDISANO, T. dan SHRESTHA, G., 2021. Identifying Resistance to the Colorado Potato Beetle (Leptinotarsa decemlineata Say) in Potato Germplasm: Review Update. Frontiers in Agronomy, 3, pp. 1-16. https://doi.org/10.3389/fagro.2021.642189.

ZHANG, X., LI, H., SUN, S., ZHANG, W., SHI, F., ZHANG, R. dan LIU, Q., 2023. Classification and Identification of Apple Leaf Diseases and Insect Pests Based on Improved ResNet-50 Model. Horticulutrae, 9(9). https://doi.org/10.3390/horticulturae9091046.

GUPTA, S., SHARMA, N. dan JAIN, S. 2021. Soybean Insect Classification Using Convolution Neural Network. International Journal of Creative Research Thoughts, 9(8).

SASTRAHIDAYAT., 2011. Tanaman Kentang dan Pengendalian Hama Penyakitnya. Malang: Universitas Brawijaya Press.

Halaman ini sengaja dikosongkan.