IMPLEMENTASI DEEP LEARNING MENGGUNAKAN MODEL DENSENET121 DAN YOLOV9 UNTUK KLASIFIKASI SPESIES LEBAH TANPA SENGAT DI LEMBAH SUHITA BANDAR LAMPUNG

(Skripsi)

Oleh

LEO FETRI HENDLI
NPM 2215061020

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

IMPLEMENTASI DEEP LEARNING MENGGUNAKAN MODEL DENSENET121 DAN YOLOV9 UNTUK KLASIFIKASI SPESIES LEBAH TANPA SENGAT DI LEMBAH SUHITA BANDAR LAMPUNG

Oleh

LEO FETRI HENDLI

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar SARJANA TEKNIK

Pada

Jurusan Teknik Elektro
Fakultas Teknik

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

ABSTRAK

IMPLEMENTASI DEEP LEARNING MENGGUNAKAN MODEL DENSENET121 DAN YOLOV9 UNTUK KLASIFIKASI SPESIES LEBAH TANPA SENGAT DI LEMBAH SUHITA BANDAR LAMPUNG

Oleh

LEO FETRI HENDLI

Identifikasi spesies lebah tanpa sengat secara manual masih bergantung pada keahlian khusus dan berpotensi menimbulkan kesalahan. Penelitian ini merancang sistem klasifikasi spesies lebah tanpa sengat di kawasan Lembah Suhita, Bandar Lampung. Sistem yang dirancang menggunakan arsitektur DenseNet121 dan YOLOv9 pada 1.470 citra yang terbagi ke dalam 7 kelas. Pengujian dilakukan pada ukuran input 224×224 dan 640×640 dengan teknik augmentasi yang sama, yaitu *flipping*, perubahan kecerahan, dan *random cropping*. Pada penelitian ini, DenseNet121 diterapkan menggunakan pendekatan *hybrid prediction* berbasis *embedding* dan indeks DLS, sedangkan YOLOv9 menggunakan data beranotasi *bounding box*. Pengembangan model dimulai dari melakukan pengumpulan data citra di Lembah Suhita, dilanjutkan dengan pra pemrosesan citra, pelatihan model, dan evaluasi model. Hasil menunjukkan bahwa model DenseNet121 dengan input 224×224 memberikan performa terbaik dengan akurasi (94,29%), DenseNet121 dengan input 640×640 menghasilkan akurasi (92,14%), YOLOv9 dengan input 640x640 menghasilkan akurasi (89%) dan YOLOv9 dengan input 224×224 menghasilkan akurasi (93%).

Kata Kunci : *Computer vision*, Klasifikasi gambar, DenseNet121, YOLOv9.

ABSTRACT

IMPLEMENTATION OF DEEP LEARNING USING DENSENET121 AND YOLOV9 MODELS FOR CLASSIFICATION OF STINGLESS BEE SPECIES AT LEMBAH SUHITA, BANDAR LAMPUNG

By

LEO FETRI HENDLI

Manual species identification still relies heavily on specialized expertise and is prone to errors. This study design a classification system for stingless honey bee species in the Suhita Valley area, Bandar Lampung. This system uses the DenseNet121 and YOLOv9 architectures on 1,470 images which are divided into 7 classes. Experiments were conducted using input sizes of 224×224 and 640×640 with the same augmentation techniques, namely flipping, brightness adjustment, and random cropping. In this study, DenseNet121 was implemented using a hybrid prediction approach based on embeddings and a DLS index, while YOLOv9 utilized data annotated with bounding boxes. The model development process began with image data collection in Suhita Valley, followed by image preprocessing, model training, and model evaluation. The results show that DenseNet121 with a 224×224 input achieved an accuracy of 94.29%, DenseNet121 with a 640×640 input achieved an accuracy of 92.14%, YOLOv9 with a 640×640 input achieved an accuracy of 89%, and YOLOv9 with a 224×224 input achieved an accuracy of 93%. In this study, the test results on the data used showed that the DenseNet121 model with 224x224 input has higher accuracy compared to other models tested.

Key words: Computer vision, Image Classification, DenseNet121, YOLOv9.

Judul Skripsi	:	IMPLEMENTASI DEEP LEARNING MENGGUNAKAN MODEL DENSENET121 DAN YOLOV9 UNTUK KLASIFIKASI SPESIES LEBAH TANPA SENGAT DI LEMBAH SUHITA BANDAR LAMPUNG
Nama Mahasiswa	:	Leo Fetri Hendli
Nomor Pokok Mahasiswa	:	2215061020
Program Studi	:	Teknik Informatika
Jurusan	:	Teknik Elektro
Fakultas	:	Teknik

MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama	Pembimbing Pendamping
Yessi Mulyani, S.T., M.T.	Rio Ariestia Pradipta, S.Kom, M.T.I.
NIP. 197312262000122001	NIP. 198603232019031013

2. Mengetahui

Ketua Jurusan Teknik Elektro	Ketua Program Studi Teknik Informatika
Herlinawati, S.T., M.T.	Yessi Mulyani, S.T., M.T.
NIP. 197103141999032001	NIP. 197312262000122001

MENGESAHKAN

1. Tim Penguji

Ketua : **Yessi Mulyani, S.T., M.T.** .....

Sekretaris : **Rio Ariestia Pradipta, S.Kom, M.T.I.** .....

Penguji : **Dr. Eng. Ir. Mardiana, S.T., M.T., IPM** .....

2. Dekan Fakultas Teknik

Dr. Ir. Ahmad Herison, S.T., M.T.
NIP. 196910302000031001

Tanggal Lulus Ujian Skripsi : 03 Februari 2026

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi berjudul "Implementasi Deep Learning Menggunakan Model Densenet121 Dan Yolov9 Untuk Klasifikasi Spesies Lebah Tanpa Sengat Di Lembah Suhita Bandar Lampung" sepenuhnya merupakan hasil karya saya sendiri. Apabila di kemudian hari terbukti pernyataan ini tidak benar, saya siap menerima sanksi sesuai dengan ketentuan hukum yang berlaku.

Bandar Lampung, 20 April 2026

Penulis,

Leo Fetri Hendli

NPM. 2215061020

RIWAYAT HIDUP

Penulis dilahirkan di Muara Enim, Sumatera Selatan pada tanggal 23 Februari 2005, sebagai anak kedua dari pasangan Bapak Hendri dan Ibu Nelly. Penulis menyelesaikan pendidikan formal di SDN 1 Muara Enim pada tahun 2016, kemudian melanjutkan ke SMPN 1 Muara Enim dan lulus pada tahun 2019, serta menamatkan pendidikan menengah atas di SMA Negeri 1 Unggulan Muara Enim pada tahun 2022. Pada tahun 2022, penulis diterima sebagai mahasiswa Program Studi Teknik Informatika Fakultas Teknik Universitas Lampung melalui jalur seleksi SNMPTN. Selama masa perkuliahan, penulis aktif berpartisipasi dalam berbagai kegiatan, antara lain:

1. Mengikuti kegiatan Studi Independen Bersertifikat dari Kementerian Pendidikan dan Budaya Batch 7 di mitra MIKTI pada tahun 2024.
2. Menjadi Ketua Divisi Kewirausahaan Himpunan Mahasiswa Teknik Elektro pada tahun 2024-2025.
3. Menjadi Ketua Umum UKM Buddha Universitas Lampung pada tahun 2025.
4. Menjadi Asisten Laboratorium Teknik Komputer pada tahun 2024-2026

MOTTO

“Datang, Lihat, dan Buktikan.”

(Buddha Gautama)

“Pada saat-saat tergelaplah kita harus fokus untuk melihat cahaya.”

(Aristoteles)

“*Many major sins in the name of Love*”

(Penulis)

PERSEMBAHAN

Dengan penuh rasa syukur kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, karya ini saya persembahkan kepada:

Kedua orang tua tercinta, yang telah menjadi sumber kekuatan, semangat, kasih sayang, dan doa dalam setiap langkah hidup saya.

Keluarga besar yang selalu memberi semangat dan dukungan tanpa henti.

Sahabat-sahabat seperjuangan yang senantiasa hadir dalam suka dan duka.

Serta almamater kebanggaan, Universitas Lampung, tempat saya menimba ilmu dan pengalaman berharga.

SANWACANA

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat, dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul "Implementasi Deep Learning Menggunakan Model Densenet121 Dan Yolov9 Untuk Klasifikasi Spesies Lebah Tanpa Sengat Di Lembah Suhita Bandar Lampung" dengan baik.

Penyusunan skripsi ini tidak terlepas dari dukungan, bimbingan, serta bantuan berbagai pihak yang telah memberikan kontribusi, baik dalam bentuk moril maupun materil. Oleh karena itu, dengan penuh rasa hormat dan ketulusan, penulis menyampaikan ucapan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Kedua orang tua tercinta beserta keluarga besar yang senantiasa memberikan doa, kasih sayang, semangat, dukungan, serta motivasi yang tiada henti kepada penulis, baik secara moril, materi maupun spiritual, sehingga penulis mampu menyelesaikan studi dan penelitian ini dengan penuh semangat.;
2. Bapak Dr. Hi. Ahmad Herison, S.T., M.T. selaku Dekan Fakultas Teknik, Universitas Lampung.;
3. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung.;
4. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika Fakultas Teknik Universitas Lampung, yang telah memberikan kesempatan, bimbingan, serta dorongan kepada penulis dalam menyelesaikan skripsi ini.;
5. Ibu Resty Annisa, S.ST., M.Kom, selaku Dosen Pembimbing Akademik, yang telah memberikan bimbingan, perhatian, motivasi, serta menjadi tempat berkeluh kesah selama penulis menjalani masa perkuliahan.;

6. Ibu Yessi Mulyani, S.T., M.T., selaku Dosen Pembimbing Utama, yang dengan sabar, telaten, dan penuh perhatian telah memberikan arahan, masukan, serta ilmu yang sangat berharga selama proses penelitian dan penulisan skripsi ini.;
7. Bapak Rio Ariestia Pradipta, S.Kom, M.T.I., selaku Dosen Pembimbing Pendamping, yang telah memberikan bimbingan dan arahan dalam melakukan penelitian ini.;
8. Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., IPM, selaku Dosen Penguji, yang telah memberikan kritik, saran, dan masukan yang membangun demi penyempurnaan hasil penelitian ini.;
9. Seluruh Dosen Program Studi Teknik Informatika Fakultas Teknik Universitas Lampung, yang telah memberikan ilmu, wawasan, dan pengalaman berharga selama masa perkuliahan.;
10. Rekan satu tim penelitian, Arswendo Erza Sadewa, Ghebi Armando, atas kerja sama yang hebat, rasa kebersamaan, serta kontribusinya dalam pengembangan Sistem Klasifikasi Gambar Lebah ini.;
11. Seluruh rekan Rectical 2022 yang tidak dapat disebutkan satu per satu, atas kebersamaan, dukungan, dan semangat yang diberikan selama masa perkuliahan hingga penyusunan skripsi ini.;
12. Seluruh rekan Asisten Laboratorium Teknik Komputer atas canda tawa, semangat, dan bantuannya selama ini.;
13. Seluruh rekan grup “Natar”, grup “ Yang AI AI Aja”, grup “Beo”, serta anggota kelas PSTI D yang telah memberikan motivasi untuk menyelesaikan skripsi ini, menjadi teman dalam setiap perkuliahan, menjadi teman dalam project kelompok, dan menjadi teman pada setiap perkembangan.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, sehingga kritik dan saran yang membangun sangat diharapkan. Semoga karya ini dapat memberikan manfaat serta menjadi referensi bagi pengembangan penelitian di bidang *Computer Vision*. Penulis juga berharap seluruh pihak yang telah memberikan dukungan dan bantuan selama proses penyusunan skripsi ini senantiasa mendapatkan limpahan rahmat dan balasan kebaikan dari Tuhan Yang Maha Esa.

Bandar Lampung, 20 April 2026

Penulis,

Leo Fetri Hendli

NPM. 2215061020

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

I. PENDAHULUAN

1.1 Latar Belakang

Teknologi *computer vision* dapat dimanfaatkan pada bidang pertanian dan peternakan untuk berbagai keperluan. Salah satu cabang utama dari *computer vision* adalah *image classification*, yaitu proses pengenalan objek dalam gambar dan menetapkannya ke dalam satu atau beberapa kelas. Dalam penelitian biologis, *image classification* dapat digunakan untuk mengidentifikasi spesies hewan, termasuk lebah madu [1]. Identifikasi ini penting untuk tujuan pemantauan populasi, evaluasi kesehatan koloni, dan pelestarian keanekaragaman hayati seperti koloni lebah.

Identifikasi dan pemantauan spesies lebah secara manual memerlukan keahlian khusus dan waktu yang tidak efisien, terutama dalam membedakan lebah asli dengan serangga peniru yang secara morfologis sangat mirip. Seiring berkembangnya teknologi, pendekatan berbasis kecerdasan buatan seperti *computer vision* dan *image classification* telah terbukti mampu mengklasifikasikan lebah dan penirunya dengan akurasi tinggi [2].

Salah satu tempat budidaya yang ada di Bandar Lampung yaitu Lembah Suhita. Lembah suhita merupakan tempat pusat edukasi dan pelatihan perlebahan. Pengunjung dapat mengenal proses pemeliharaan lebah, panen madu, serta pentingnya peran lebah bagi lingkungan. Proses identifikasi secara manual memerlukan waktu yang lama, tingkat ketelitian tinggi, dan sering kali bergantung pada keahlian individu. Selain itu, citra lebah madu memiliki ukuran tubuh yang kecil, warna serupa antarspesies, serta tekstur morfologi halus, sehingga sulit dibedakan secara visual oleh metode konvensional [3].

Kemajuan teknologi *computer vision* dan *deep learning* memungkinkan pengembangan sistem klasifikasi citra yang mampu mengenali pola visual secara otomatis. Salah satu arsitektur populer adalah *Convolutional Neural Network* (CNN), yang mampu mengekstraksi fitur hierarkis dari gambar. DenseNet121, salah satu varian CNN, dikenal unggul dalam mempertahankan informasi fitur melalui koneksi padat (dense connections) antar-layer, sehingga cocok digunakan untuk objek berukuran kecil atau dengan detail halus. Beberapa penelitian sebelumnya menunjukkan DenseNet121 mampu mencapai akurasi tinggi pada klasifikasi serangga dan objek kecil lainnya [4], serta pada sistem pengenalan hama atau serangga di lapangan [5].

Namun, model CNN murni hanya melakukan klasifikasi citra yang sudah terpotong (cropped image) dan tidak mampu mendeteksi posisi objek secara langsung di dalam gambar. Untuk mengatasi keterbatasan tersebut, digunakan algoritma deteksi objek YOLO (*You Only Look Once*), yang dirancang untuk mengenali dan menentukan lokasi objek dalam satu kali proses inferensi. YOLO dikenal efisien dalam mendeteksi banyak objek sekaligus dengan kecepatan tinggi, sehingga cocok digunakan pada kasus deteksi objek kecil seperti lebah madu di lingkungan alami [6]. Salah satu penelitian juga menggunakan YOLO untuk mendeteksi serangga kecil [7].

Salah satu varian terkini dari keluarga YOLO adalah YOLOv9, yang dilengkapi dengan *Generalized Efficient Layer Aggregation Network* (GELAN) dan *Programmable Gradient Information* (PGI) untuk meningkatkan efisiensi ekstraksi fitur dan stabilitas pelatihan. YOLOv9 terbukti mampu mendeteksi objek kecil secara lebih akurat dan cepat dibandingkan beberapa generasi YOLO lainnya [8].

Penelitian ini bertujuan untuk membandingkan performa DenseNet121 dan YOLOv9 dalam identifikasi spesies lebah madu. DenseNet121 digunakan untuk klasifikasi spesies berdasarkan citra hasil deteksi, sedangkan YOLOv9 berperan dalam mendeteksi posisi objek lebah dalam citra asli. Pendekatan ini diperkuat dengan mekanisme *hybrid prediction* berbasis *image embedding* (DenseLinkSearch) untuk meningkatkan akurasi pada kasus prediksi berkepercayaan rendah. Diharapkan sistem ini dapat menghasilkan model

identifikasi lebah madu yang akurat, dan andal, terutama untuk citra objek kecil di lingkungan alami.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, dapat dirumuskan beberapa pertanyaan penelitian sebagai berikut:

1. Bagaimana membangun model untuk mengklasifikasikan spesies lebah madu tanpa sengat menggunakan DenseNet121 dan YOLOv9?
2. Bagaimana kinerja model DenseNet121 dan YOLOv9 dalam mengklasifikasikan lebah madu tanpa sengat?
3. Bagaimana perbandingan akurasi antara metode DenseNet121 dan YOLOv9 dalam identifikasi lebah madu tanpa sengat?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disampaikan, maka tujuan penelitian ini adalah sebagai berikut :

1. Membangun model yang memanfaatkan DenseNet121 dan YOLOv9 untuk klasifikasi spesies lebah madu tanpa sengat.
2. Menerapkan dan menganalisa kinerja model DenseNet121 dan *YOLOv9* dalam mengklasifikasikan spesies lebah madu tanpa sengat.
3. Membandingkan akurasi antara metode DenseNet121 dan YOLOv9 dalam identifikasi spesies lebah madu tanpa sengat.

1.4 Manfaat Penelitian

Adapun manfaan penelitian ini adalah sebagai berikut :

1. Bagi peneliti, penelitian ini memberikan pengalaman dalam penerapan model CNN khususnya DenseNet121, dan YOLO v9, serta memperluas pemahaman tentang penggunaan *computer vision* untuk klasifikasi objek biologis seperti lebah.

2. Penelitian ini diharapkan menjadi penelitian yang berdampak untuk dimanfaatkan untuk menjadi referensi bagi penelitian selanjutnya dalam topik identifikasi spesies berbasis citra.

1.5 Batasan Penelitian

Adapun batasan dari penelitian ini adalah sebagai berikut :

1. Penelitian ini hanya dilakukan pada gambar 6 spesies lebah madu yang berasal dari kawasan Lembah Suhita, Bandar Lampung. Spesies lebah tersebut diantaranya yaitu Tetrigona apicalis, Tetragonula biroi, Heterotrigona itama, Geniotrigona thoracica, Tetrigona binghami, dan Tetrigona vidua, serta tambahan 1 kelas *unknown* diluar dari 6 spesies tersebut.
2. Gambar yang digunakan merupakan citra statis (*still image*), bukan video atau citra bergerak.
3. Evaluasi performa model dilakukan dengan metrik klasifikasi standar seperti akurasi, presisi, *recall*, dan *F1-score*.

1.6 Sistematika Penulisan

Adapun sistematika penulisan dari penelitian ini adalah sebagai berikut :

1. BAB I: PENDAHULUAN : Berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.
2. BAB II: TINJAUAN PUSTAKA : Membahas teori-teori yang mendasari penelitian, termasuk konsep dasar lebah, *computer vision*, klasifikasi citra, *deep learning*, CNN, DenseNet121, YOLO v9 dan studi terdahulu yang relevan.
3. BAB III: METODOLOGI PENELITIAN : Menjelaskan tahapan pelaksanaan penelitian, mulai dari pengumpulan data, *preprocessing*, perancangan arsitektur DenseNet121, dan YOLO v9, proses pelatihan dan evaluasi, serta alat dan bahan yang digunakan.

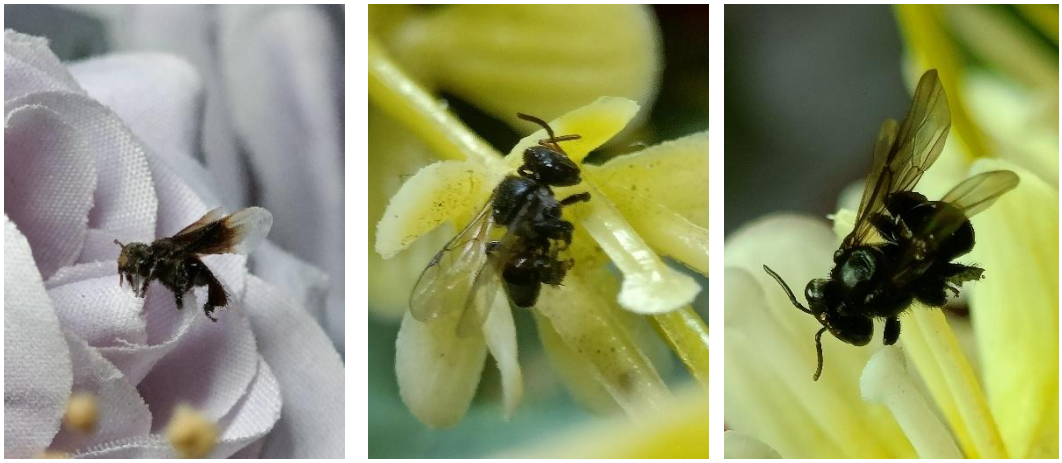
4. BAB IV: HASIL DAN PEMBAHASAN : Menyajikan hasil eksperimen klasifikasi lebah madu menggunakan DenseNet121, dan YOLO v9 dan menganalisis kinerja model berdasarkan data yang diperoleh.
5. BAB V: KESIMPULAN DAN SARAN : Berisi kesimpulan dari hasil penelitian dan saran untuk pengembangan penelitian lebih lanjut.
6. DAFTAR PUSTAKA : Memuat referensi atau sumber-sumber yang digunakan dalam penelitian ini, baik dari buku, jurnal ilmiah, maupun artikel terpercaya lainnya.

II. TINJAUAN PUSTAKA

2.1 Lebah Madu Tanpa Sengat

Lebah tanpa sengat atau kelulut merupakan kelompok lebah berukuran kecil yang termasuk dalam suku Meliponini, dan masih berkerabat dekat dengan lebah madu bersengat (Trigona sp.). Secara morfologis, tubuh lebah tanpa sengat terbagi menjadi tiga bagian utama, yaitu kepala, dada (thoraks), dan abdomen. Pada bagian thoraks terdapat dua pasang sayap dan tiga pasang kaki, dimana kaki belakang dilengkapi dengan keranjang serbuk sari (*pollen basket*) untuk mengangkut serbuk bunga. Di bagian kepala terdapat sepasang mata majemuk, tiga mata tunggal (ocelli), serta sepasang antena yang berfungsi sebagai organ peraba dan sensor lingkungan [9].

Lebah kelulut dikenal sebagai serangga eusosial, yaitu memiliki sistem kehidupan sosial yang kompleks, mirip dengan lebah madu (Apis), semut, dan rayap. Kehidupan sosial ini ditandai dengan adanya pembagian tugas yang jelas di dalam koloni. Dalam satu koloni biasanya terdapat satu atau lebih ratu lebah, ratusan lebah jantan (*drone*), serta ratusan hingga ribuan lebah pekerja.



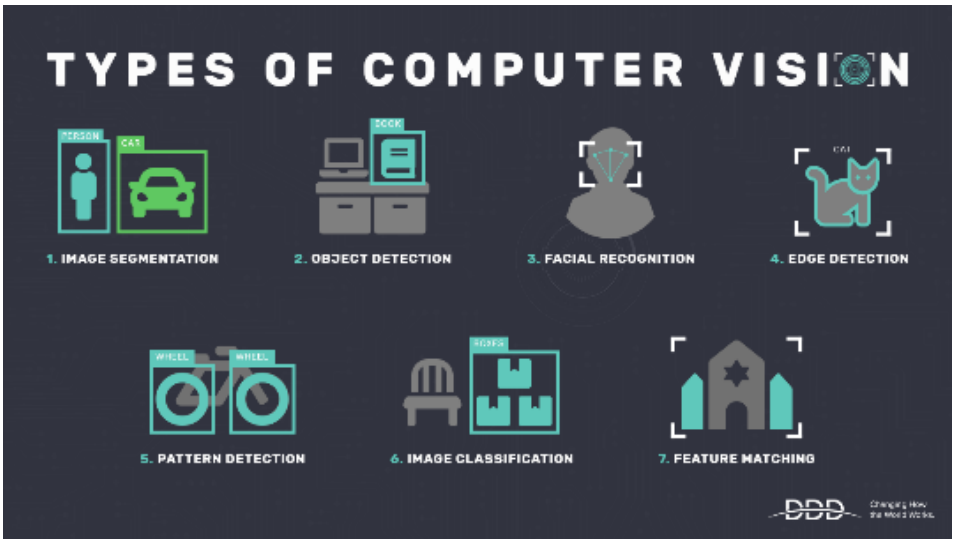
Gambar 1 Lebah tanpa sengat

2.2 *Computer Vision*

Computer vision adalah cabang dari kecerdasan buatan yang berfokus pada bagaimana komputer dapat "melihat" dan memahami gambar atau video layaknya manusia memproses penglihatan. Bidang ini mengembangkan teknik matematis dan algoritmik agar komputer mampu menganalisis dan mengenali objek dalam citra, merekonstruksi bentuk tiga dimensi (3D) dari gambar dua dimensi (2D), serta memahami konteks visual seperti membedakan objek utama dari latar belakang atau mengenali ekspresi wajah manusia [10].

Meskipun telah mengalami kemajuan pesat, kemampuan komputer dalam memahami gambar masih belum sebanding dengan kemampuan persepsi visual manusia, karena penglihatan merupakan masalah yang tidak pasti (*ill-posed problem*) yang memerlukan pendekatan berbasis fisika, probabilistik, dan pembelajaran mesin (machine learning) untuk menafsirkan informasi visual dengan tepat. Dengan demikian, *computer vision* dapat diartikan sebagai bidang yang mengajarkan komputer untuk meniru kemampuan penglihatan manusia melalui analisis, pemrosesan, dan interpretasi data visual secara otomatis [11].

Teknologi ini telah diaplikasikan secara luas dalam berbagai sektor, termasuk medis, manufaktur, keamanan, pertanian, dan konservasi. Dalam studi perilaku dan ekologi hewan, *computer vision* digunakan untuk mengamati spesies dan mengotomatisasi identifikasi morfologi makhluk hidup [1].



Gambar 2. Jenis Computer Vision

Berbagai jenis tipe teknologi *computer vision* telah tersedia, seperti segmentasi gambar, deteksi objek, pengenalan wajah, deteksi tepi, deteksi pola, klasifikasi gambar, dan pencocokan fitur. Setiap jenis memiliki fungsi spesifik dalam mengenali dan menganalisis elemen visual dalam gambar. Misalnya, *image classification* digunakan untuk mengelompokkan gambar ke dalam kategori tertentu, sementara object detection menandai lokasi objek seperti mobil atau buku dalam sebuah gambar.

2.2 *Image Classification*

Image classification atau klasifikasi citra merupakan proses untuk mengelompokkan gambar ke dalam kelas-kelas tertentu berdasarkan informasi visual yang terkandung di dalamnya, seperti warna, tekstur, bentuk, dan pola spasial. Dalam perkembangannya, model klasifikasi modern banyak mengandalkan teknik *deep learning*, khususnya *Convolutional Neural Networks* (CNN), yang mampu mengekstraksi fitur secara otomatis dan hierarkis dari citra input tanpa memerlukan perancangan fitur secara manual. Pendekatan ini memungkinkan model untuk mempelajari representasi visual dari tingkat rendah hingga tingkat tinggi, sehingga meningkatkan akurasi dan robustitas sistem klasifikasi. Image classification telah menjadi komponen inti dalam berbagai aplikasi, terutama pada sistem diagnosis medis berbasis citra, pengenalan objek, dan analisis citra skala besar. Seiring dengan kemajuan arsitektur jaringan saraf visual seperti ResNet, Inception, dan DenseNet, kinerja model klasifikasi citra terus mengalami peningkatan [12].

Meskipun demikian, tantangan utama dalam *image classification* masih terletak pada kebutuhan akan data berlabel dalam jumlah besar dan berkualitas tinggi untuk melatih model secara efektif, serta potensi terjadinya *overfitting* ketika data yang tersedia terbatas.

2.3 *Deep Learning*

Deep learning merupakan cabang dari *machine learning* yang mempelajari data secara bertingkat atau hierarkis. Metode ini dikembangkan dari konsep *artificial*

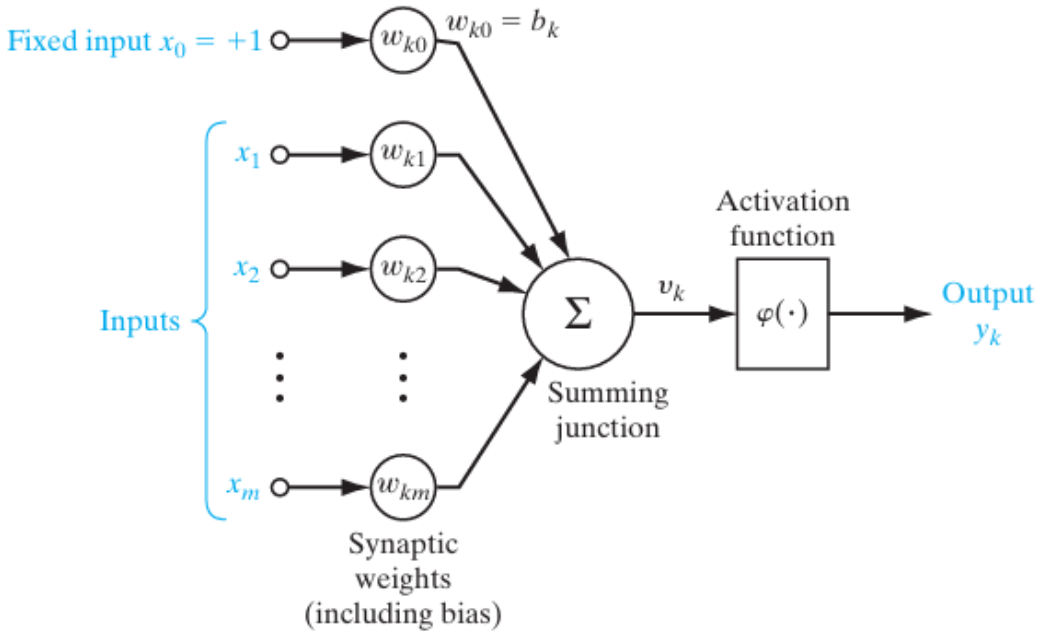
neural network (ANN) yang meniru cara kerja jaringan saraf manusia. Dalam *deep learning*, proses pembelajaran dilakukan melalui banyak lapisan (*layers*), dimana lapisan awal mengekstraksi fitur-fitur sederhana, sedangkan lapisan yang lebih dalam menghasilkan representasi fitur yang semakin kompleks [13]. Teknologi ini banyak digunakan dalam berbagai bidang seperti klasifikasi, *clustering*, segmentasi, maupun *recognition*.

Deep learning umumnya diimplementasikan melalui Convolutional Neural Networks (CNN), yang dirancang khusus untuk mengolah data visual. Dengan mekanisme ini, *deep learning* memungkinkan sistem klasifikasi citra bekerja secara end-to-end, mulai dari pengolahan citra mentah hingga menghasilkan prediksi kelas. Keunggulan utama *deep learning* terletak pada kemampuannya meningkatkan akurasi dan generalisasi model, terutama ketika didukung oleh dataset berukuran besar dan beragam, sehingga menjadikannya pendekatan dominan dalam berbagai aplikasi *computer vision* modern.

Deep learning sangat sesuai diterapkan pada jenis data yang tidak terstruktur, seperti teks, suara, dan citra. Namun, pendekatan *deep learning* tradisional sangat bergantung pada banyaknya data berlabel, dan hal ini menjadi kendala pada domain seperti klasifikasi otolith ikan, yang memerlukan keahlian manual untuk labeling [14]. Oleh karena itu, muncul kebutuhan untuk pendekatan alternatif yang lebih efisien dari segi data.

2.4 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (JST) adalah model komputasi yang terinspirasi dari cara kerja sistem saraf biologis pada manusia, khususnya otak. Model ini terdiri dari sejumlah unit pemroses sederhana yang disebut neuron tiruan, yang saling terhubung dan bekerja bersama untuk mengolah informasi. Setiap neuron menerima input, mengalikan dengan bobot tertentu, menjumlahkannya, lalu menghasilkan output melalui fungsi aktivasi [15].



Gambar 3. Struktur Jaringan Saraf Tiruan [15]

Gambar 3 menggambarkan model matematis sebuah neuron tiruan sebagai unit komputasi paling dasar dalam jaringan saraf tiruan. Setiap neuron menerima sejumlah sinyal masukan x1,x2,...,xn yang merepresentasikan fitur atau variabel dari suatu data. Selain itu, terdapat satu masukan tambahan bernilai tetap x0=+1 yang berfungsi sebagai bias. Bias ini berperan untuk menggeser batas keputusan (decision boundary) sehingga model tidak selalu terikat pada titik asal (origin) dalam ruang fitur. Dengan kata lain, bias meningkatkan fleksibilitas model dalam menyesuaikan pola data.

Setiap masukan dikalikan dengan bobot sinaptik masing-masing. Bobot mencerminkan tingkat kepentingan atau kontribusi suatu fitur terhadap keluaran neuron. Jika bobot bernilai besar dan positif, maka fitur tersebut memberikan pengaruh kuat dalam meningkatkan nilai aktivasi; sebaliknya, bobot negatif akan mengurangi atau menekan pengaruh masukan tersebut. Dalam proses pelatihan (training), nilai bobot inilah yang disesuaikan secara iteratif menggunakan algoritma optimasi (misalnya berbasis gradien) agar kesalahan prediksi dapat diminimalkan.

Seluruh hasil perkalian antara input dan bobot kemudian dijumlahkan pada bagian yang disebut summing junction. Proses ini menghasilkan suatu nilai agregasi yang sering disebut sebagai sinyal net atau potensial aktivasi. Secara konseptual, tahap ini merepresentasikan proses integrasi sinyal pada neuron biologis, di mana berbagai rangsangan yang masuk akan dikombinasikan sebelum menghasilkan respons.

Nilai agregasi tersebut selanjutnya dilewatkan ke fungsi aktivasi. Fungsi aktivasi memiliki peran yang sangat krusial karena memperkenalkan sifat nonlinier ke dalam model. Tanpa fungsi aktivasi nonlinier, jaringan saraf hanya akan menjadi model linear sederhana, meskipun terdiri dari banyak lapisan. Fungsi aktivasi memungkinkan jaringan mempelajari pola kompleks, seperti klasifikasi nonlinier atau hubungan yang tidak dapat direpresentasikan oleh garis lurus dalam ruang fitur. Contoh fungsi aktivasi yang umum digunakan meliputi sigmoid, ReLU, dan tanh, masing-masing dengan karakteristik matematis dan implikasi pelatihan yang berbeda.

Keluaran akhir neuron, yang dilambangkan sebagai yk, merupakan hasil transformasi dari fungsi aktivasi terhadap sinyal agregasi. Nilai ini dapat diinterpretasikan sebagai skor, probabilitas, atau sinyal yang akan diteruskan ke neuron lain pada lapisan berikutnya dalam jaringan. Dalam konteks jaringan multilayer, satu neuron tidak berdiri sendiri, melainkan menjadi bagian dari sistem yang saling terhubung, di mana keluaran dari satu neuron menjadi masukan bagi neuron lainnya.

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

Gambar 4 Rumus Jaringan Saraf Tiruan [15]

Gambar 4 menjelaskan bahwasannya setiap input xi terlebih dahulu dikalikan dengan bobotnya wi, di mana bobot menunjukkan seberapa penting atau besar

pengaruh input tersebut terhadap hasil akhir. Semua hasil perkalian tersebut kemudian dijumlahkan untuk menggabungkan seluruh pengaruh input menjadi satu nilai. Setelah proses penjumlahan, ditambahkan bias b yang berfungsi untuk menggeser hasil sehingga model menjadi lebih fleksibel dalam menyesuaikan pola data. Nilai yang telah diperoleh selanjutnya dimasukkan ke dalam fungsi aktivasi f(·), yang bertugas mengubah nilai tersebut menjadi output akhir y, misalnya dalam bentuk probabilitas atau nilai prediksi tertentu. Secara sederhana, alurnya adalah input dikalikan bobot, kemudian dijumlahkan, ditambahkan bias, dimasukkan ke fungsi aktivasi, dan akhirnya menghasilkan output. Itulah proses dasar satu neuron dalam neural network.

2.4 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan pengembangan dari Artificial Neural Network (ANN) konvensional dengan arsitektur yang terdiri atas puluhan hingga ratusan lapisan (*layers*). CNN memproses citra melalui serangkaian lapisan jaringan dan menghasilkan keluaran berupa kelas tertentu. Setiap lapisan dalam CNN melakukan proses pembelajaran, dimana hasil dari satu lapisan menjadi masukan bagi lapisan berikutnya. Pada tahap awal, lapisan CNN mengekstraksi fitur-fitur dasar seperti warna, kecerahan, dan tepi objek. Keunggulan utama CNN dibandingkan ANN adalah kemampuannya dalam mengekstraksi fitur yang relevan secara otomatis dari citra yang diberikan. Selain itu, CNN juga menerapkan konsep *weight sharing*, yang memungkinkan efisiensi waktu dan penggunaan memori selama proses komputasi [13].

$$s(i,j,c_2) = \sum_{c_1 \in \{C_1\}} \sum_{(k,l) \in \mathcal{N}} w(k,l,c_1,c_2) x(i+k, j+l, c_1) + b(c_2),$$

Gambar 5. Rumus CNN [10]

Keterangan :

s = output

x = input

w = nilai kernel/filter

b = bias

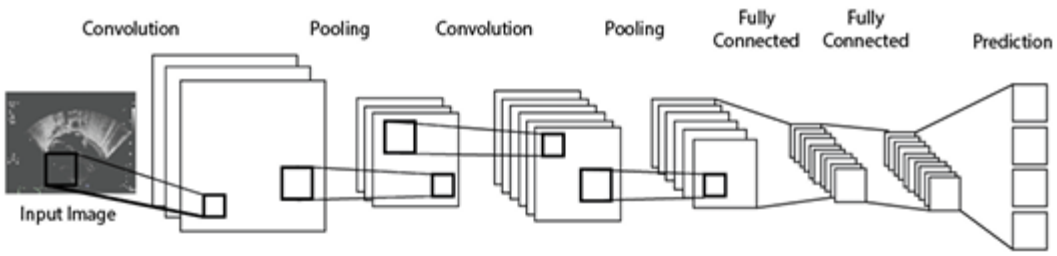
Misalkan:

$$X = \begin{bmatrix} 10 & 20 & 30 \\ 20 & 30 & 40 \\ 30 & 40 & 50 \end{bmatrix}, \quad w = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad b = 2$$

$$s = \sum W \cdot X = (1)(1)+(0)(2)+(-1)(1)+(1)(0)+(0)(1)+(-1)(2)+(1)(3)+(0)(1)+(-1)(0)$$

$$= 1+0-1+0+0-2+3+0+0 = 1$$

Kemudian tambahkan bias, 1 + b = 1 + 2 = 3. Jika menggunakan aktivasi bisa menambahkan ReLU, menjadi f (s) = f (3) = max (0, 3) = 3. Sehingga didapatkan output konvolusi nya adalah 3.



Gambar 6. Arsitektur Utama CNN [16]

a) *Convolution Layer*

Convolutional layer merupakan komponen utama CNN tempat sebagian besar proses komputasi terjadi. Lapisan ini menggunakan filter (kernel) berukuran tertentu, misalnya 5×5×3, dimana 5 menunjukkan tinggi dan lebar, sedangkan 3 menunjukkan jumlah channel (RGB). Filter tersebut bergerak ke seluruh area gambar dan melakukan operasi dot product antara nilai filter dan input untuk menghasilkan *activation map* (*feature map*) [16]. Dalam proses konvolusi terdapat dua parameter penting, yaitu stride dan padding:

- Stride menentukan seberapa jauh filter bergeser pada setiap langkah. Stride kecil (misalnya 1) menghasilkan detail lebih tinggi namun memerlukan komputasi lebih besar, sedangkan stride besar mempercepat proses tetapi mengurangi detail [16].
- Padding (zero padding) menambahkan nilai nol di tepi gambar agar ukuran output tetap terjaga [16].

Dalam konvolusi, sebuah filter (kernel) atau mask (misalnya ukuran 3×3 atau 5×5) digeser di seluruh area gambar. Pada setiap posisi, nilai piksel gambar dikalikan elemen filter satu per satu, lalu dijumlahkan hasilnya. Hasil penjumlahan ini menjadi satu nilai baru pada feature map (*activation map*). Misalkan punya gambar (*matrix pixel*), lalu digunakan *mask* (kernel) ukuran kecil, misalnya 3×3 dengan bias 0.1, seperti ini:

Gambar (bagian kecil)		Filter / Mask (3×3)
[10 20 30]		[0 -1 0]
[20 30 40]	x	[-1 5 -1]
[30 40 50]		[0 -1 0]

Kemudian, setiap elemen dikalikan satu per satu dan dijumlahkan:

(10×0)+(20×−1)+(30×0)+(20×−1)+(30×5)+(40×−1)+(30×0)+(40×−1)+(50 x0) = 30

Hasil akhirnya adalah satu angka baru, yang jadi nilai pixel hasil filter (masking).

b) Pooling Layer

Setelah melewati proses *convolution layer*, tahap berikutnya adalah *pooling* atau *sub-sampling layer*. Tujuan utama dari tahap ini adalah mengurangi jumlah parameter yang perlu dilatih serta menyederhanakan representasi data agar lebih mudah diolah dan mengurangi risiko *overfitting*. Beberapa metode pooling yang umum digunakan antara lain *max pooling*, yang

mengambil nilai maksimum dari area tertentu, dan *average pooling*, yang menghitung nilai rata-ratanya [16].

Gambar 7. Max Pooling [16]

Gambar 8. Average Pooling [16]

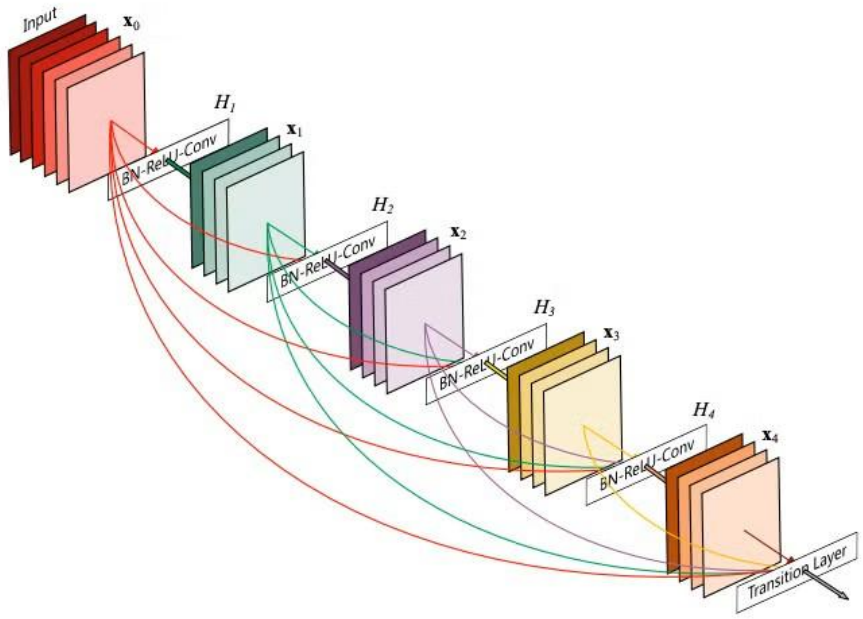
c) *Fully Connected layer*

Hasil dari *feature extraction layer* masih berbentuk array multidimensi, sehingga perlu dilakukan proses flattening atau reshape menjadi vektor satu dimensi agar dapat digunakan sebagai masukan pada *fully connected layer*. Lapisan ini terdiri atas beberapa *hidden layer*, fungsi aktivasi, lapisan output, serta fungsi loss yang berperan dalam proses pembelajaran dan klasifikasi akhir [16].

2.5 DenseNet121

DenseNet (*Densely Connected Convolutional Networks*) adalah salah satu arsitektur jaringan saraf tiruan konvolusional (*Convolutional Neural Network* - CNN) yang digunakan untuk klasifikasi citra. Konsep utamanya adalah koneksi yang lebih padat (*denser connections*) yang menghasilkan akurasi yang lebih tinggi. Prinsip sentral DenseNet adalah penggunaan kembali fitur (*feature reuse*) dan koneksi langsung (*direct connections*). Berbeda dengan jaringan lain, DenseNet menghubungkan setiap lapisan secara langsung dengan setiap lapisan sebelumnya

di dalam blok yang disebut *Dense Blocks*, dan menggunakan operasi penggabungan (*concatenation*) dari output lapisan-lapisan sebelumnya sebagai input. Arsitektur ini membantu mengurangi masalah vanishing-gradient, memperkuat penyebaran fitur, dan mengurangi jumlah parameter karena tidak perlu mempelajari peta fitur yang tidak penting [17].



Gambar 9. Arsitektur DenseNet [17]

DenseNet-121 adalah varian spesifik dari arsitektur DenseNet, dimana angka 121 menunjukkan total jumlah lapisan dalam jaringan saraf [17]. Arsitektur ini terdiri dari kombinasi berbagai lapisan, termasuk lima lapisan konvolusi dan pooling, tiga lapisan transisi (*transition layers*), satu lapisan klasifikasi, dan dua *Dense Blocks*. Dalam praktiknya, DenseNet-121 sering digunakan sebagai fondasi model *deep learning* untuk tugas klasifikasi citra, seperti yang ditunjukkan dalam penelitian untuk mendiagnosis kasus COVID-19 menggunakan citra Chest X-ray (CXR) [18]. Untuk meningkatkan kinerjanya, model DenseNet-121 sering diimplementasikan menggunakan konsep *Transfer Learning* dan *Fine Tuning* dengan bobot yang telah dilatih sebelumnya (*pre-trained weights*) dari dataset yang besar (seperti

ImageNet), yang memungkinkan model untuk mencapai akurasi lebih tinggi dan lebih cepat.

Arsitektur DenseNet121 pada umumnya menggunakan resolusi input 224×224 piksel karena model ini awalnya dirancang dan dilatih menggunakan dataset ImageNet, yang secara standar menerapkan ukuran citra tersebut sebagai input. Penggunaan resolusi 224×224 memungkinkan pemanfaatan bobot *pre-trained* ImageNet secara optimal dalam skema *transfer learning*, sehingga proses pelatihan menjadi lebih efisien dan stabil. Selain faktor kesesuaian dengan bobot awal, pemilihan ukuran 224×224 juga didasarkan pada pertimbangan teknis berupa keseimbangan antara kompleksitas komputasi dan kemampuan representasi fitur. Resolusi ini dinilai cukup untuk menangkap informasi visual penting pada berbagai objek, sekaligus menjaga kebutuhan memori dan waktu komputasi tetap efisien [19].

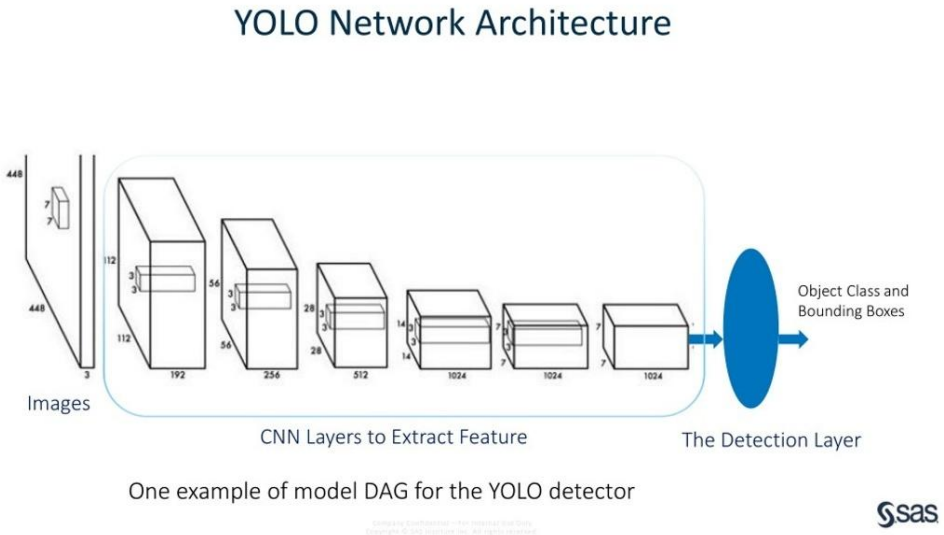
2.6 YOLO

You Only Look Once (YOLO) adalah algoritma yang sangat populer dan banyak digunakan dalam bidang deteksi objek (*object detection*), yang merupakan dasar bagi kecerdasan buatan (*artificial intelligence*). YOLO (*You Only Look Once*) merevolusi metode deteksi objek dengan mengubah konsep tugas deteksi menjadi masalah regresi tunggal yang terpadu, bukan lagi proses bertahap seperti pada metode konvensional. Pendekatan tradisional biasanya melibatkan beberapa tahap terpisah, seperti pembuatan proposal wilayah, klasifikasi objek, serta tahap pasca-pemrosesan seperti non-maximum suppression. YOLO menyederhanakan seluruh proses ini dengan mengintegrasikan deteksi dan klasifikasi ke dalam satu jaringan saraf tunggal yang secara langsung memprediksi *bounding box* dan probabilitas kelas dari keseluruhan citra dalam satu kali evaluasi [20].

Pendekatan ini dilakukan dengan membagi citra masukan menjadi beberapa sel grid, dimana setiap sel bertanggung jawab mendeteksi objek yang pusatnya berada di dalam area sel tersebut. Untuk setiap sel, jaringan memprediksi *bounding box* beserta skor kepercayaan dan probabilitas kelas secara bersamaan. Dengan cara ini, YOLO menggabungkan informasi lokasi dan klasifikasi secara langsung

berdasarkan fitur spasial yang diekstraksi oleh *convolutional backbone*. Output dari jaringan terdiri atas sejumlah tetap *bounding box* per sel, yang masing-masing mencakup koordinat, ukuran, nilai kepercayaan, serta probabilitas kelas bersyarat.

Sejak diperkenalkan versi pertamanya pada tahun 2015, algoritma YOLO terus mengalami peningkatan. Perkembangan ini telah menghasilkan beberapa versi lanjutan, seperti YOLO V2, YOLO V3, YOLO V4, YOLO V5, YOLOv6, YOLOv7 [21]. Versi-versi baru dalam seri YOLO yang lebih mutakhir termasuk YOLOv8, YOLOV9, YOLOv10, dan YOLOv11 [8].



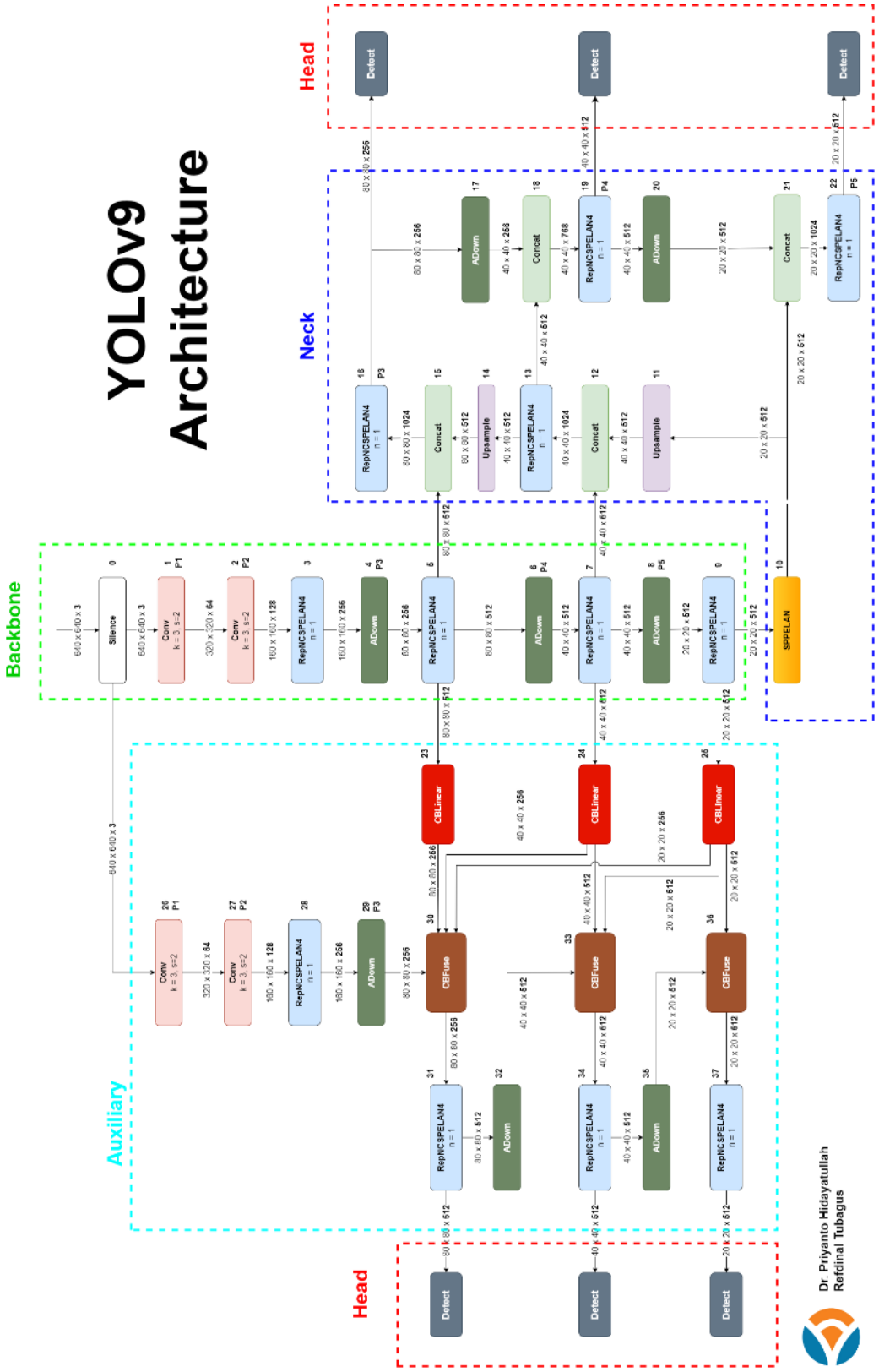
Gambar 10. Arsitektur YOLO

YOLOv9 merupakan peningkatan signifikan dalam deteksi objek real-time, memperkenalkan teknologi baru seperti *Programmable Gradient Information* (PGI) dan *Generalized Efficient Layer Aggregation Network* (GELAN). Model ini meningkatkan efisiensi, akurasi, dan fleksibilitas, serta menetapkan standar baru pada dataset MS COCO [22].

PGI membantu mengatasi masalah hilangnya informasi dalam jaringan saraf dalam, memastikan data penting tetap tersedia untuk pembaruan model yang akurat. GELAN meningkatkan pemanfaatan parameter dan efisiensi komputasi, memungkinkan integrasi berbagai blok komputasi tanpa mengorbankan kecepatan

atau akurasi. Arsitektur ini sangat penting untuk model ringan, yang biasanya rentan kehilangan informasi selama proses *feedforward*, sehingga YOLOv9 tetap mempertahankan performa deteksi yang tinggi meski dengan jumlah parameter yang lebih sedikit [22].

Pada YOLOv9 ini menggunakan AMP *(Automatic Mixed Precision)*. *Automatic Mixed Precision* (AMP) adalah teknik pelatihan *deep learning* yang menggunakan kombinasi presisi numerik rendah (seperti float16) dan presisi tinggi (float32) secara otomatis untuk meningkatkan efisiensi komputasi tanpa menurunkan performa model secara signifikan. Dalam implementasi PyTorch, AMP bekerja melalui mekanisme *autocasting* yang menjalankan operasi komputasi berat seperti konvolusi dan perkalian matriks pada presisi rendah agar lebih cepat dan hemat memori, sementara operasi yang sensitif terhadap stabilitas numerik tetap dijalankan pada presisi tinggi. Selain itu, AMP menggunakan *gradient scaling* untuk mencegah terjadinya *underflow* pada gradien selama proses *backpropagation*. Pada YOLOv9, AMP digunakan selama proses training untuk mempercepat pelatihan dan mengurangi penggunaan memori GPU, sehingga memungkinkan penggunaan *batch size* yang lebih besar dan waktu pelatihan yang lebih singkat tanpa mengubah arsitektur model maupun menurunkan akurasi deteksi secara signifikan [23].



Gambar 11. Arsitektur YOLO v9 [24]

2.7 Python

Python adalah bahasa pemrograman tingkat tinggi, berorientasi objek, dan bersifat interpreted dengan semantik yang dinamis. Struktur data bawaan yang canggih, dikombinasikan dengan pengetikan (*typing*) dan pengikatan (*binding*) yang dinamis, membuat Python sangat cocok untuk pengembangan aplikasi cepat (*Rapid Application Development*), serta dapat digunakan sebagai bahasa skrip atau penghubung (*glue*) untuk mengintegrasikan berbagai komponen. Sintaks Python yang sederhana dan mudah dipelajari menekankan keterbacaan, sehingga menurunkan biaya pemeliharaan program. Python mendukung modul dan paket, yang mendorong modularitas program dan penggunaan kembali kode. Interpreter Python beserta pustaka standarnya yang luas tersedia dalam bentuk sumber atau biner tanpa biaya untuk semua platform utama, dan dapat didistribusikan secara bebas [25].

2.8 PyTorch

PyTorch adalah sebuah *framework open-source* untuk *deep learning* dan *machine learning* yang dirancang agar fleksibel dan modular, sehingga cocok baik untuk penelitian maupun penerapan produksi. PyTorch mendukung dynamic computation graph, memungkinkan peneliti dan pengembang untuk membuat prototipe model secara cepat dan intuitif. Dengan integrasi yang mendalam dengan bahasa Python, PyTorch memudahkan pembangunan, pelatihan, dan pengujian model *deep learning* berskala besar. *Framework* ini banyak digunakan di dunia akademik maupun industri untuk berbagai aplikasi, seperti pengolahan bahasa alami (NLP), *computer vision*, *reinforcement learning*, dan *generative* AI. PyTorch juga dioptimalkan untuk performa tinggi pada CPU, GPU, dan akselerator perangkat keras khusus, serta mendukung pelatihan terdistribusi dan *deployment* pada platform cloud maupun perangkat mobile [26].

2.9 Augmentasi

Augmentasi gambar adalah teknik yang digunakan untuk meningkatkan kuantitas dan keragaman dari data pelatihan yang terbatas. Tujuan utamanya adalah untuk

mengatasi masalah data yang tidak mencukupi (*insufficient data* atau *small sample learning*) yang menghambat akurasi dan kemampuan generalisasi model *deep learning* [27]. Salah satu teknik dari augmentasi yaitu transformasi geometris. Transformasi geometris adalah upaya untuk memodifikasi hubungan spasial antar piksel. Tujuan utamanya adalah untuk meningkatkan variabilitas gambar dalam dataset pelatihan, meniru variasi yang dapat muncul pada objek dalam gambar yang diambil secara alami [28]. Metode transormasi geometris yang digunakan, diantaranya :

a) *Flip* (Pembalikan): Dapat dilakukan secara *horizontal* atau *vertikal*, tergantung pada karakteristik dataset pelatihan dan pengujian. Misalnya, dataset Cityscapes dapat diaugmentasi secara *horizontal* tetapi tidak secara *vertikal*. Operasi *flip* telah umum digunakan dalam berbagai tugas *computer vision* [28].
b) *Resize* : Mengubah ukuran (*resize*) input ke tinggi dan lebar yang diberikan [27]. *Resize* mengubah ukuran gambar awal menjadi ukuran seragam, seperti ukuran 224 tanpa mengubah ukuran pada gambar asli.
c) Perubahan kecerahan : Teknik augmentasi citra yang mengubah tingkat terang–gelap gambar dengan memodifikasi nilai intensitas piksel, tanpa mengubah bentuk, struktur, atau label objek di dalam citra [29].
d) *Random crop* : Teknik augmentasi spasial yang memotong sebagian area citra secara acak, kemudian diubah ukurannya kembali ke ukuran input model [29].

Augmentasi *on-the-fly* adalah metode dimana pembuatan data sintetis (data tambahan hasil modifikasi data asli) dilakukan secara langsung hanya pada saat proses pelatihan model berlangsung, bukan disimpan secara permanen [30].

Pada setiap model klasifikasi menerima data baru dan menghasilkan prediksi awal. Mekanisme ini terintegrasi dengan strategi *active learning* yang berfungsi mengevaluasi signifikansi sampel data, apabila strategi memutuskan bahwa label diperlukan, sistem akan meminta anotasi *ground truth* dari oracle dan menyimpan pasangan data tersebut ke dalam antrean memori multi-kelas. Tahap augmentasi diterapkan secara *on-the-fly* atau seketika, dimana proses pembuatan data sintetis

diinisiasi tepat sebelum pembaruan parameter model dilakukan. Data sintetis hasil transformasi tersebut ditampung dalam struktur memori sementara yang disebut antrean teraugmentasi. Selanjutnya, model diperbarui secara inkremental menggunakan himpunan data gabungan dari antrean asli dan antrean teraugmentasi. Pendekatan ini menawarkan efisiensi komputasi yang signifikan, karena alokasi memori untuk bersifat sementara dan segera dihapus pasca-pelatihan, sehingga tidak membebani kapasitas penyimpanan sistem secara permanen [30].

2.10 *Confusion Matrix*

Confusion Matrix adalah sebuah matriks yang digunakan untuk menganalisis sejauh mana sebuah classifier mampu mengenali data dari berbagai kelas. Matriks ini menampilkan informasi mengenai jumlah prediksi yang benar maupun yang salah yang dihasilkan oleh model klasifikasi untuk tiap kelas target [31].

Dalam Confusion Matrix terdapat empat komponen utama:

a) *True Positive* (TP): Jumlah sampel yang benar-benar diklasifikasikan sebagai positif oleh model.
b) *True Negative* (TN): Jumlah sampel yang benar-benar diklasifikasikan sebagai negatif oleh model.
c) *False Positive* (FP): Jumlah sampel yang keliru diklasifikasikan sebagai positif oleh model (false alarm).
d) *False Negative* (FN): Jumlah sampel yang keliru diklasifikasikan sebagai negatif oleh model (miss).

Keempat nilai ini digunakan untuk menghitung berbagai metrik evaluasi, seperti presisi, *recall*, akurasi, dan *F1-score* [31]. Berdasarkan nilai-nilai dari *Confusion Matrix*, metrik-metrik tersebut dapat dihitung melalui persamaan berikut :

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

$$Presisi = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$f1 - score = \frac{2 \times Presisi \times Recall}{Presisi \times Recall}$$

Gambar 12. Persamaan Confusion Matrix [31]

Berikut penjelasan terkait metrik evaluasi :

a) Akurasi mengukur seberapa banyak prediksi model yang benar dibandingkan dengan seluruh data yang diuji.
b) Presisi menunjukkan seberapa banyak prediksi positif model yang benar-benar positif. Presisi tinggi berarti model jarang memberikan false alarm (FP sedikit).
c) *Recall* mengukur seberapa banyak data positif yang berhasil dikenali oleh model. Recall tinggi berarti model jarang melewatkan kasus positif (miss kecil).
d) *F1-score* adalah rata-rata harmonik dari *precision* dan *recall*. Digunakan saat ingin menyeimbangkan keduanya, terutama jika dataset tidak seimbang.

2.11 Penelitian Terdahulu

Salah satu tantangan utama yang dihadapi dalam implementasi metode ini adalah keakuratan model, terutama pada domain khusus seperti identifikasi objek biologis, termasuk klasifikasi lebah madu. Berbagai studi terkini telah menunjukkan efektivitas metode ini, baik dalam domain medis, biologi, maupun pertanian. Berikut ini adalah beberapa penelitian yang relevan sebagai gambaran penggunaan klasifikasi citra menggunakan algoritma DenseNet121 dan YOLO v9:

1. Klasifikasi citra berbasis *deep learning* telah membantu dalam identifikasi penyakit daun tomat. Namun, CNN konvensional menunjukkan keterbatasan termasuk tingginya tuntutan komputasi dan kebutuhan akan data berlabel yang substansial, serta jejak memori yang besar. Oleh karena itu, arsitektur DenseNet121 diusulkan dan menunjukkan kinerja superior (akurasi validasi 87.47%) dibandingkan CNN (akurasi validasi 83.33%) [32].
2. Klasifikasi citra berbasis *deep learning* telah membantu dalam deteksi gulma untuk meningkatkan hasil panen dan mengurangi penggunaan pestisida dalam pertanian presisi. Namun, metode tradisional seringkali tidak efisien dan padat karya. Oleh karena itu, model DenseNet121 diusulkan dan menunjukkan kinerja superior (akurasi validasi 0.97) dibandingkan ResNet50 (0.91) dan model VGG lainnya [33].
3. Klasifikasi citra berbasis *deep learning* telah membantu dalam diagnosis kasus COVID-19 menggunakan citra rontgen dada (CXR). Namun, muncul tantangan karena kurangnya data yang cukup untuk menerapkan model yang akurat dalam memprediksi COVID-19. Oleh karena itu, model DenseNet-121 dengan *Transfer Learning* dan teknik *Early Stopping* diusulkan, mencapai akurasi 98.45% untuk kelas normal-sehat dan 98.32% untuk kelas COVID-19 [18].
4. Klasifikasi citra berbasis *deep learning* telah membantu dalam klasifikasi jenis hijab (pashmina, segi empat, dan hijab instan) untuk mendukung otomatisasi dalam industri fashion dan *e-commerce*. Namun, perkembangan industri menghadirkan tantangan dalam mengklasifikasikan jenis hijab karena variasi visual yang dinamis. Oleh karena itu, model DenseNet-121 diusulkan dan berhasil mencapai akurasi 0.89 [34].
5. Klasifikasi citra berbasis *deep learning* telah membantu dalam pengenalan citra akurat di bidang *computer vision*. Namun, model konvensional masih menghadapi masalah efisiensi pengenalan yang rendah dan akurasi yang buruk, serta masalah *bottleneck* komunikasi pada pelatihan terdistribusi. Oleh karena itu, sebuah strategi peningkatan untuk DenseNet (akurasi 97.2%) dan algoritma akselerasi paralel berbasis Gradient Quantization

(GQ-SDP) diusulkan, yang secara signifikan mengurangi waktu pelatihan dan memecahkan masalah bottleneck komunikasi data [35].

6. Klasifikasi citra berbasis AI telah membantu dalam identifikasi dan klasifikasi kendaraan (sepeda motor, mobil, bus, truk) untuk memantau lalu lintas dan memperkirakan tingkat polusi udara (emisi CO) di perkotaan. Namun, tantangan besar muncul ketika model mengalami kesulitan dengan benda-benda kecil seperti sepeda motor (mAP50:95 0.682), yang seringkali terlihat menyatu dengan latar belakang karena sudut kamera CCTV dan ukurannya yang kecil. Oleh karena itu, model YOLOv9 diusulkan, yang meskipun demikian mampu mendeteksi kendaraan secara keseluruhan dengan akurasi memuaskan (mAP50:95 0.826) dan memprediksi jumlah kendaraan dengan benar lebih dari 80% dalam kondisi malam yang ramai, dan 100% dalam kondisi malam yang sepi) [36].
7. Klasifikasi citra berbasis *deep learning* telah membantu dalam identifikasi dan klasifikasi jenis kerbau Toraja (bongga sori, bonga ulu, bulan, saleko, dan todi). Namun, banyak masyarakat Toraja kesulitan mengidentifikasi jenis kerbau yang sesuai untuk upacara adat Rambu Solo. Oleh karena itu, model YOLOv9 diusulkan, yang berhasil mencapai presisi sekitar 0.9 dan recall 0.8. Meskipun demikian, selama proses pelatihan, teramati pola *overfitting* dan underfitting yang menunjukkan perlunya peningkatan volume dan keragaman data pelatihan untuk mencapai generalisasi yang lebih seimbang [37].
8. Klasifikasi citra berbasis *deep learning* telah membantu dalam deteksi stempel pada dokumen yang dipindai untuk mengotomatisasi proses autentikasi dan memerangi pemalsuan dokumen. Namun, tantangan besar muncul dari variasi bentuk, warna, sudut, tumpang tindih dengan teks, serta stempel yang pudar atau buram dalam dataset. Oleh karena itu, model YOLOv9s diusulkan dan menunjukkan kinerja terbaik (mAP 98.7% dengan presisi dan recall 97.6%) dibandingkan YOLOv8s, YOLOv10s, dan YOLOv11s. Meskipun demikian, YOLOv9s memiliki waktu inferensi rata-rata dua kali lebih lama dari model lain karena arsitekturnya yang lebih kuat dan jumlah lapisan yang lebih banyak [38].

9. Klasifikasi citra berbasis *deep learning* telah membantu dalam deteksi berbagai spesies gulma (cocklebur, dandelion, common waterhemp, Palmer amaranth, dan common lambsquarters) untuk implementasi manajemen gulma yang spesifik spesies dan lokasi (SSSWM). Namun, akurasi dan presisi deteksi gulma tetap menjadi tantangan karena variabilitas biologis yang besar di antara spesies dan kondisi lingkungan tempat mereka tumbuh. Oleh karena itu, model YOLOv9 diusulkan dan menunjukkan akurasi tertinggi di antara semua model yang diuji (mAP@0.5 sebesar 0.935). Meskipun demikian, YOLOv9 memiliki waktu inferensi yang jauh lebih lambat (54.2 ms) dibandingkan YOLOv11 (13.5 ms) dan YOLOv8 (23 ms) [8].
10. Klasifikasi citra berbasis *deep learning* telah membantu dalam deteksi objek pada citra penginderaan jarak jauh untuk aplikasi pengawasan dan manajemen. Namun, tantangan besar muncul dari deteksi objek kecil, variasi skala, objek yang berdekatan, dan efek *motion blur*. Oleh karena itu, model YOLOv9 yang ditingkatkan dengan *Transformer Head* (YOLOv9-TH) diusulkan, yang menunjukkan peningkatan signifikan dalam mAP untuk deteksi objek kecil. Peningkatan ini menyebabkan model YOLOv9-TH-e mencapai 54.2% mAP50 pada dataset VisDrone2021 dan 92.3% mAP pada dataset DIOR [39].

III. METODE PENELITIAN

3.1 Waktu dan Tempat

Penelitian ini dilaksanakan pada bulan September 2025 sampai Januari tahun 2026, bertempat di Lembah Suhita, Bandar Lampung untuk proses *survey* dan pengambilan data citra lebah madu, dan di laboratorium teknik komputer Universitas Lampung untuk proses pemodelan dan analisis data menggunakan perangkat lunak pendukung.

Berikut adalah tabel alur pelaksanaan penelitian ini :

Tabel 1. Alur Pelaksanaan Penelitian

No	Tahapan	Bulan Pelaksanaan				
		September 2025	Oktober 2025	November 2025	Desember 2025	Januari 2026
1	Identifikasi Masalah					
2	Studi Literatur					
3	Pengumpulan Data					
4	Pra-pemrosesan Citra					
5	Pelatihan Model Klasifikasi					
6	Evaluasi dan Pengujian Akhir					

3.2 Bahan dan Alat

Adapun bahan yang digunakan, yaitu :

1. Dataset citra lebah tanpa sengat dari Lembah Suhita (berupa foto dalam format .jpg/.png) dan data gambar non lebah tanpa sengat sebagai tambahan yang berjumlah sebanyak 1.470 citra.

Adapun alat yang digunakan adalah sebagai berikut :

1. Perangkat Keras (*Hardware*)
   a) Laptop dengan spesifikasi AMD Ryzen 3 3250U, dan RAM 8 GB.
   b) Ponsel dengan kamera 48 MP sebagai alat pengambilan citra.
   c) GPU berbasis cloud NVIDIA Tesla (A100) yang tersedia pada Google Colaboratory untuk mempercepat proses pelatihan model *deep learning*.
2. Perangkat Lunak (*Software*)
   a) Sistem Operasi: Windows 11 sebagai *software environment* untuk menjalankan model dan sistem lain.
   b) Bahasa Pemrograman: Python versi 3.12.12 sebagai bahasa pemrograman yang digunakan untuk mengolah data, membuat model, dan visualisasinya.
   c) Platform: Google Colaboratory sebagai platform *open source* untuk melatih model, menjalankan kode python, dan mengolah datanya.
   d) Anotasi data: Roboflow untuk pelabelan *bounding box* pada citra

3.3 Metode

3.3.1 DenseNet121

DenseNet121 digunakan sebagai *feature extractor* untuk menghasilkan embedding citra yang kemudian di-index menggunakan algoritma pencarian efisien untuk mendukung prediksi atau pencarian berdasar kemiripan [40].

1) Feature Extraction

Pada tahap ini, digunakan arsitektur DenseNet121 yang telah ditransfer-belajar dari ImageNet. Model ini berfungsi mengekstraksi fitur-fitur penting dari citra lebah madu, sehingga menghasilkan feature map berukuran tinggi yang merepresentasikan karakteristik visual objek.

2) *Image Embedding*

Feature map yang dihasilkan oleh DenseNet121 kemudian diubah menjadi embedding vector, yaitu representasi numerik berdimensi tetap. Embedding ini digunakan sebagai dasar untuk perhitungan kemiripan antar-citra, pengambilan keputusan berbasis *similarity*, serta mendukung mekanisme prediksi hibrid.

3) *Hybrid Prediction*

Hybrid Prediction menggabungkan prediksi konvensional CNN dan prediksi berbasis kemiripan menggunakan *DenseLinkSearch* (DLS). Pada tahap awal, model menghasilkan probabilitas kelas melalui Softmax, dan jika tingkat kepercayaannya melewati ambang batas (CONF_THRESH), prediksi CNN langsung digunakan. Namun ketika model kurang yakin, sistem menghitung embedding citra dan membandingkannya dengan embedding data pelatihan menggunakan kNN. Jika ditemukan kemiripan yang tinggi (≥ SIM_THRESH), kelas terdekat dipilih sebagai prediksi melalui mekanisme DLS-Fallback. Apabila tidak ada kemiripan yang memadai, citra diklasifikasikan sebagai "*Unknown*". Pendekatan ini meningkatkan *robustnes* terhadap kasus ambigu sekaligus memungkinkan deteksi kelas di luar data pelatihan (*open-set*).

3.3.2 YOLO v9

1) *Feature Extraction*

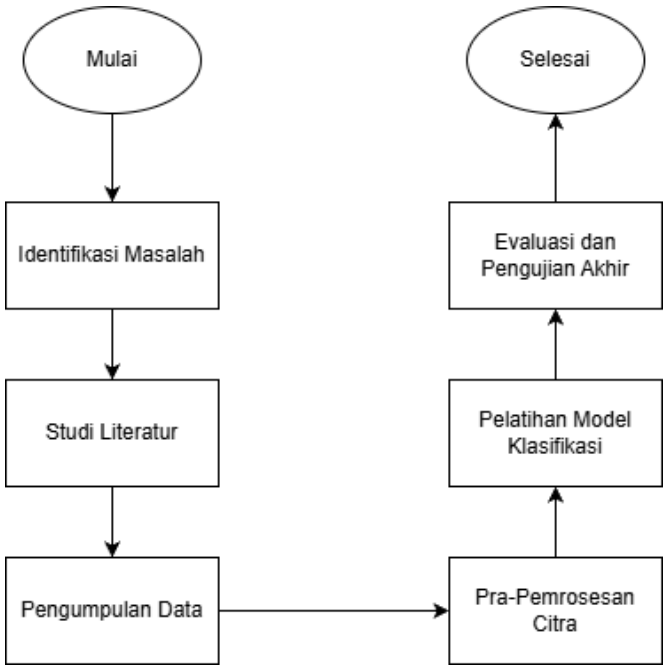
YOLOv9 melakukan ekstraksi fitur secara langsung melalui arsitektur *backbone-neck-head*. Model ini menghasilkan *bounding box*, prediksi kelas, dan *confidence score* dalam satu proses.

2) Prediksi Deteksi Objek

YOLO melakukan inferensi dengan memprediksi lokasi objek *(bounding box), confidence score,* dan kelas objek lebah. Pada metode YOLO, tidak digunakan *image embedding, denseLinkSearch (DLS), dan hybrid prediction.*

3.4 Alur Pelaksanaan

Berikut adalah alur pelaksanaan penelitian ini :



Gambar 13. Flowchart Alur Pelaksanaan

Tahap 1 : Identifikasi Masalah

Tahap ini bertujuan untuk merumuskan permasalahan yang akan dipecahkan dalam penelitian. Berdasarkan observasi awal dan studi pendahuluan, diketahui bahwa identifikasi jenis lebah secara manual memerlukan keahlian khusus, memakan waktu, dan rawan kesalahan. Selain itu, ketersediaan metode otomatis untuk klasifikasi citra lebah masih terbatas, terutama untuk membedakan jenis-jenis lebah yang memiliki kemiripan visual tinggi. Berdasarkan masalah-masalah ini, penelitian diarahkan untuk mengembangkan sistem klasifikasi citra lebah menggunakan kecerdasan buatan, sehingga dapat meningkatkan efisiensi, kecepatan, dan akurasi identifikasi lebah.

Tahap 2 : Studi Literatur

Pada tahap ini terdiri dari studi literatur serta observasi. Tahap studi literatur dilakukan melalui pengumpulan referensi dari berbagai sumber seperti buku, jurnal ilmiah, dan penelitian terdahulu yang relevan yang mendukung pengambangan sistem. Kegiatan yang dilakukan pada tahap studi literatur meliputi:

1. Mempelajari terkait spesies lebah madu tanpa sengat.

2. Mempelajari teori terkait *Computer Vision*, *Deep Learning*, dan *CNN*, khususnya model untuk klasifikasi gambar.
3. Mencari penelitian yang pernah dilakukan terkait *image classification* dan menemukan metode yang digunakan serta permasalahan yang terjadi.
4. Menganalisis berbagai model dan metode yang dilakukan pada penelitian sebelumnya dan mempelajari model yang akan digunakan.

Selain studi literatur, pada tahap ini juga dilakukan observasi awal terhadap kondisi lapangan dan ketersediaan data yang berkaitan dengan penelitian. Observasi dilakukan dengan meninjau sumber data citra lebah tanpa sengat yang diperoleh dari dokumentasi di kawasan Lembah Suhita, Bandar Lampung. Observasi ini bertujuan untuk memahami karakteristik citra yang digunakan, seperti variasi spesies lebah, kondisi pencahayaan, sudut pengambilan gambar, serta kualitas citra yang tersedia.

Berdasarkan hasil studi literatur dan observasi tersebut, penelitian ini menggunakan dua pendekatan model deep learning, yaitu DenseNet121 dan YOLOv9. Model DenseNet121 dipilih karena memiliki arsitektur jaringan yang mampu memanfaatkan konektivitas antar lapisan secara efisien sehingga dapat mengekstraksi fitur citra secara lebih mendalam dan mengurangi masalah *vanishing gradient*. Model ini juga banyak digunakan dalam tugas klasifikasi citra karena mampu menghasilkan performa yang baik meskipun dengan jumlah data yang relatif terbatas melalui pendekatan *transfer learning*.

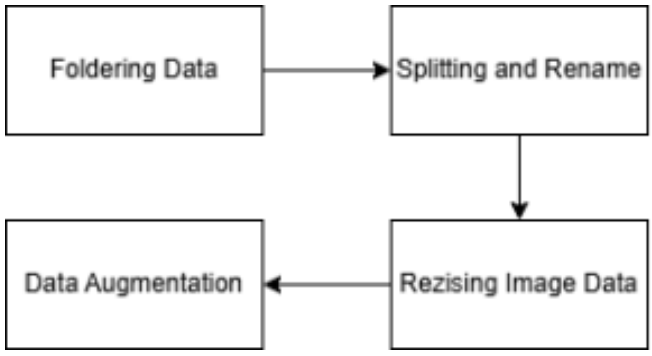
Sementara itu, YOLOv9 dipilih karena merupakan salah satu model deteksi objek yang memiliki kemampuan mendeteksi objek secara cepat dan akurat dalam satu tahap proses (*single-stage detector*). Arsitektur YOLOv9 dirancang untuk menghasilkan proses inferensi yang efisien sehingga cocok digunakan dalam pengolahan citra yang membutuhkan kecepatan deteksi yang tinggi. Dengan karakteristik tersebut, kedua model ini dinilai dapat saling melengkapi dalam penelitian, yaitu dengan membandingkan pendekatan klasifikasi citra menggunakan DenseNet121 dan deteksi objek menggunakan YOLOv9 dalam mengidentifikasi spesies lebah tanpa sengat.

Tahap 3 : Pengumpulan Data

Pada tahap pengumpulan data, fokus penelitian adalah memperoleh data citra yang akan digunakan sebagai bahan utama untuk proses pelatihan dan pengujian model. Dataset diambil dari kawasan lembah suhita, Bandar Lampung, yang menghimpun citra enam spesies lebah madu tanpa sengat. Dataset hanya menggunakan 6 spesies lebah dikarenakan pada lembah suhita hanya memiliki 6 jenis lebah tanpa sengat dari total 52 spesies lebah tanpa sengat di Indonesia. Untuk meningkatkan kemampuan model dalam menghadapi kondisi open-set, kemudian ditambahkan kelas *Unknown* yang berisi citra-citra di luar enam spesies tersebut. Penambahan data ini bertujuan agar sistem mampu mengenali gambar yang tidak termasuk dalam kelas yang telah didefinisikan, sehingga meningkatkan robustnes dan realibilitas model dalam kondisi nyata. Data yang digunakan dalam penelitian ini berjumlah 1.470 citra. Sebanyak 1.400 citra dimanfaatkan sebagai dataset pelatihan, pengujian, dan validasi, sedangkan 70 citra lainnya digunakan sebagai data pengujian akhir (testing). Seluruh dataset terbagi ke dalam tujuh kelas, di mana pada dataset pelatihan, pengujian, dan validasi masing-masing kelas terdiri dari 200 citra, sementara pada data testing setiap kelas terdiri dari 10 citra.

Gambar 14. Dataset Citra Lebah Apicalis, Biroi, Itama, Thoracica, Vidua, Binghami, dan Unknown

Tahap 4 : Pra Pemrosesan Citra



Gambar 15 Preprocessing Data

Tahap pada gambar menunjukkan alur preprocessing data citra sebelum digunakan pada proses pelatihan model agar memiliki struktur yang rapi, ukuran yang seragam, serta variasi data yang cukup sehingga model dapat belajar secara lebih optimal. Alur preprocessing tersebut meliputi proses foldering data, splitting and

rename, resizing image data, dan data augmentation. Penjelasan setiap tahap adalah sebagai berikut.

A. Foldering Data

Pada tahap awal preprocessing dilakukan proses pengelompokan data citra ke dalam struktur folder yang terorganisir. Setiap citra dimasukkan ke dalam folder sesuai dengan label atau kelasnya. Misalnya, setiap kelas gestur atau kategori objek disimpan dalam folder tersendiri yang dinamai berdasarkan label kelas tersebut. Tujuan dari proses foldering data ini adalah untuk mempermudah pengelolaan dataset serta memudahkan proses pemanggilan data pada tahap preprocessing dan pelatihan model. Dengan struktur folder yang terorganisir, proses pemrosesan data dapat dilakukan secara lebih sistematis dan mengurangi kemungkinan kesalahan dalam pengolahan dataset.

B. Splitting and Rename

Setelah data tersusun dalam folder sesuai kelasnya, tahap selanjutnya adalah melakukan pembagian dataset serta penyeragaman nama file. Dataset dibagi menjadi beberapa bagian, yaitu data latih (training), data validasi (validation), dan data uji (testing). Pembagian ini bertujuan agar model dapat dilatih, divalidasi, dan diuji menggunakan data yang berbeda sehingga performa model dapat dievaluasi secara objektif.

Selain pembagian dataset, pada tahap ini juga dilakukan proses rename atau penyeragaman penamaan file citra. Setiap file diberi nama dengan format yang konsisten, misalnya *label_001.png, label_002.png*, dan seterusnya. Penyeragaman nama file ini bertujuan untuk mempermudah proses pengolahan data selanjutnya serta memastikan kesesuaian antara file citra dan data anotasi yang digunakan dalam pelatihan model.

C. Resizing Image Data

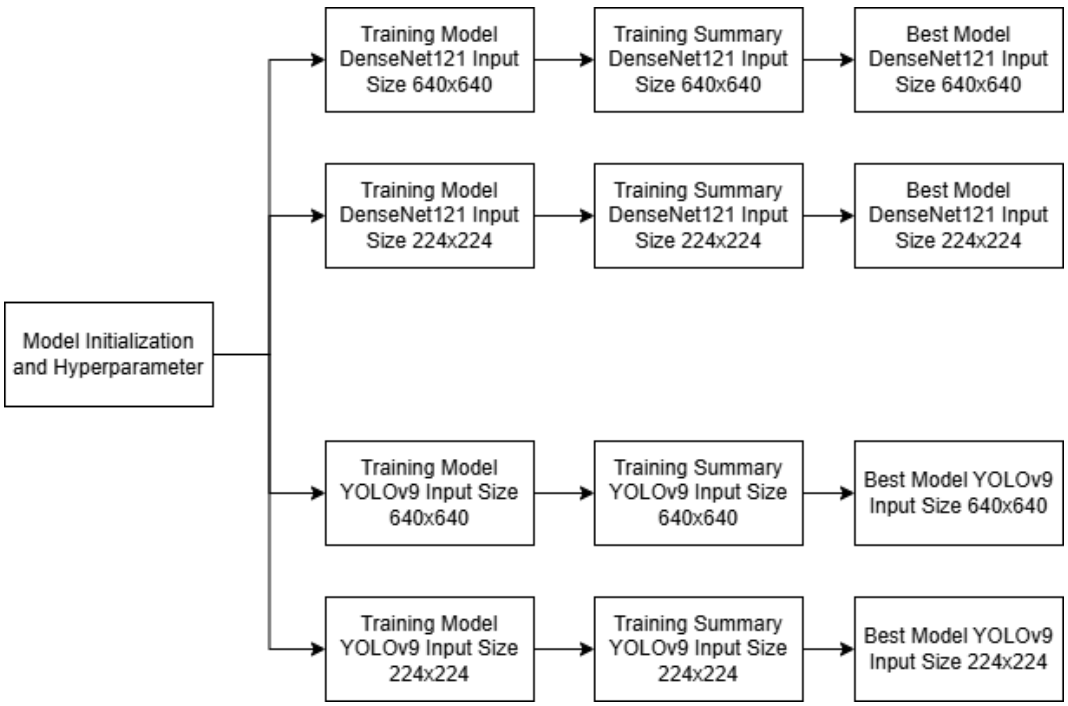
Pada tahap ini dilakukan proses perubahan ukuran (resize) pada seluruh citra dalam dataset. Setiap gambar diubah ke ukuran yang sama, misalnya 224×224 piksel atau ukuran lain yang sesuai dengan kebutuhan arsitektur model yang digunakan. Proses resizing bertujuan untuk menyeragamkan dimensi citra sehingga seluruh data

memiliki ukuran input yang konsisten saat dimasukkan ke dalam model. Selain itu, proses ini juga membantu mengurangi kompleksitas komputasi serta mempercepat proses pelatihan model.

D. Data Augmentation

Tahap terakhir adalah melakukan augmentasi data untuk meningkatkan variasi dataset, terutama pada data latih (training set). Augmentasi dilakukan dengan menerapkan berbagai transformasi pada citra, seperti rotasi, flipping, atau penyesuaian tingkat kecerahan. Transformasi ini bertujuan untuk mensimulasikan berbagai kondisi yang mungkin terjadi pada data nyata, seperti perubahan sudut pengambilan gambar, orientasi objek, maupun kondisi pencahayaan yang berbeda. Dengan adanya augmentasi data, jumlah variasi citra dalam dataset menjadi lebih banyak sehingga model dapat belajar mengenali pola secara lebih robust dan tidak mudah mengalami overfitting. Augmentasi ini hanya diterapkan pada data latih agar evaluasi model pada data validasi dan data uji tetap merepresentasikan kondisi data asli.

Tahap 5 : Pelatihan Model Klasifikasi



Gambar 16 Pelatihan Model Klasifikasi

Tahap ini merupakan proses pelatihan model (model training) menggunakan dua pendekatan model, yaitu DenseNet121 untuk klasifikasi citra dan YOLOv9 untuk deteksi objek berbasis bounding box. Penggunaan dua model ini bertujuan untuk membandingkan performa kedua metode klasifikasi citra tersebut dalam mengidentifikasi spesies lebah tanpa sengat.

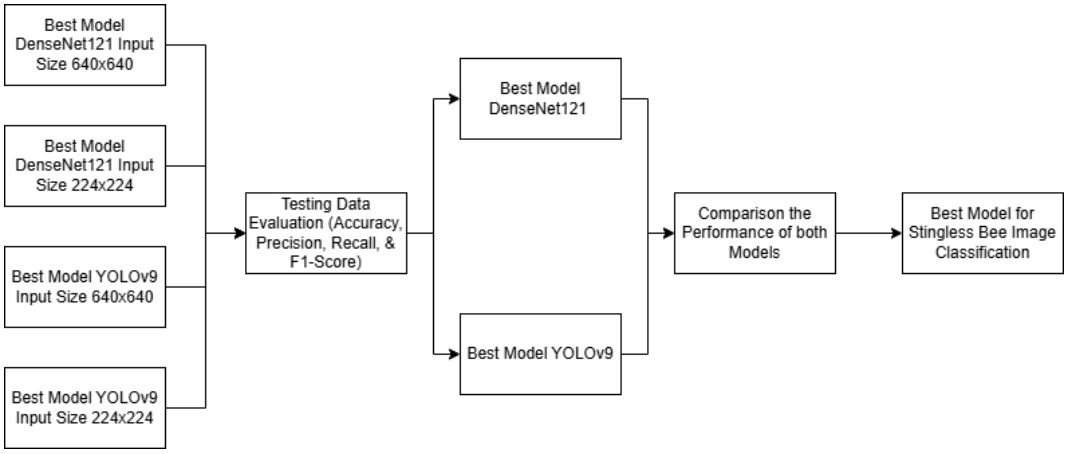
Model DenseNet121 diimplementasikan menggunakan pendekatan *transfer learning*, yaitu dengan menginisialisasi bobot awal menggunakan bobot pra-latih dari dataset ImageNet. Pendekatan ini dipilih karena model telah memiliki kemampuan dalam mengekstraksi fitur visual umum seperti bentuk, tekstur, dan pola pada citra. Dengan demikian, proses pelatihan dapat dilakukan lebih efektif meskipun jumlah dataset penelitian relatif terbatas. Pada tahap ini, lapisan *classifier* terakhir dari arsitektur DenseNet121 diganti agar sesuai dengan jumlah kelas spesies lebah yang digunakan pada penelitian. Setelah penyesuaian tersebut, model dilatih kembali (*fine-tuning*) menggunakan dataset pelatihan sehingga model dapat mempelajari karakteristik khusus dari setiap spesies lebah tanpa sengat.

Sementara itu, model YOLOv9 digunakan untuk melakukan klasifikasi spesies lebah berdasarkan lokasi objek pada citra. Pada model ini, data latih disiapkan dalam bentuk anotasi bounding box yang menunjukkan posisi objek lebah pada gambar beserta label kelasnya. Proses pelatihan dilakukan menggunakan konfigurasi bawaan dari arsitektur YOLOv9 dengan optimizer Stochastic Gradient Descent (SGD). Model kemudian dilatih menggunakan dataset pelatihan untuk mempelajari hubungan antara fitur visual dan posisi objek lebah pada citra.

Selama proses pelatihan, kedua model dilakukan proses fine-tuning menggunakan data latih dengan tujuan untuk menyesuaikan parameter model agar memperoleh performa terbaik pada data validasi. Evaluasi performa selama pelatihan dilakukan menggunakan data validasi untuk memantau proses pembelajaran model serta mencegah terjadinya *overfitting*. Model dengan performa terbaik pada data validasi kemudian disimpan sebagai model akhir yang akan digunakan pada tahap pengujian.

Pada penelitian ini juga dilakukan eksperimen terhadap ukuran input citra yang digunakan pada proses pelatihan model. Kedua model dilatih menggunakan ukuran 224x224 dan 640x640 untuk mencari ukuran yang terbaik sesuai dengan dataset dan augmentasi yang digunakan.

Tahap 6 : Evaluasi dan Pengujian Akhir



Gambar 17 Evaluasi dan Pengujian Akhir

Tahap ini merupakan evaluasi dan pengujian akhir model yang bertujuan untuk menilai kinerja model secara objektif setelah proses pelatihan selesai dilakukan. Evaluasi dilakukan menggunakan data uji (test set) yang bersifat independen, yaitu data yang tidak pernah digunakan pada tahap pelatihan maupun validasi. Penggunaan data uji yang terpisah bertujuan untuk mengukur kemampuan model dalam melakukan generalisasi terhadap data baru, sehingga hasil evaluasi dapat mencerminkan performa model ketika diterapkan pada kondisi nyata.

Pada tahap ini, model yang telah dilatih sebelumnya digunakan untuk melakukan prediksi terhadap seluruh citra pada data uji. Hasil prediksi tersebut kemudian dibandingkan dengan label sebenarnya untuk mengetahui tingkat kesesuaian antara prediksi model dan data asli. Berdasarkan perbandingan tersebut, dihitung beberapa metrik evaluasi utama untuk menilai kinerja model secara menyeluruh.

Metrik evaluasi yang digunakan dalam penelitian ini meliputi Accuracy, Precision, Recall, dan F1-score. Accuracy digunakan untuk mengukur rasio antara jumlah prediksi yang benar terhadap keseluruhan jumlah data uji. Metrik ini memberikan

gambaran umum mengenai tingkat ketepatan model dalam melakukan klasifikasi. Precision menunjukkan proporsi prediksi positif yang benar di antara seluruh prediksi yang diklasifikasikan sebagai positif oleh model. Nilai precision yang tinggi menunjukkan bahwa model memiliki tingkat kesalahan prediksi positif yang rendah.

Selanjutnya, Recall digunakan untuk mengukur kemampuan model dalam mendeteksi seluruh data positif yang sebenarnya terdapat dalam dataset. Nilai recall yang tinggi menunjukkan bahwa model mampu mengenali sebagian besar objek atau kelas yang relevan. Sementara itu, F1-score merupakan rata-rata harmonis (*harmonic mean*) antara precision dan recall, yang digunakan untuk memberikan keseimbangan antara kedua metrik tersebut. F1-score sangat berguna terutama ketika distribusi data antar kelas tidak seimbang.

Selain menghitung metrik evaluasi tersebut, pada tahap ini juga dihasilkan Confusion Matrix, yaitu tabel yang digunakan untuk menggambarkan kinerja model klasifikasi dengan membandingkan antara label sebenarnya dan label hasil prediksi model. Confusion matrix menunjukkan jumlah true positive, true negative, false positive, dan false negative untuk setiap kelas, sehingga dapat memberikan gambaran yang lebih rinci mengenai kesalahan klasifikasi yang dilakukan oleh model.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun kesimpulan dari penelitian ini adalah sebagai berikut:

1. Model DenseNet121 dan YOLOv9 berhasil diimplementasikan untuk membangun sistem klasifikasi spesies lebah madu tanpa sengat di kawasan Lembah Suhita, Bandar Lampung. Keberhasilan implementasi ini ditunjukkan oleh performa model pada data uji, di mana DenseNet121 mencapai akurasi sebesar 94,29% dan YOLOv9 mencapai akurasi hingga 93%.
2. Analisa penerapan kinerja model DenseNet121 dengan ukuran input 224×224 menunjukkan performa dengan nilai akurasi sebesar 94,29%, presisi 95%, *recall* 94%, dan *F1-score* 94% pada seluruh kelas pengujian. Sedangkan model DenseNet121 dengan ukuran input 640×640 mencapai akurasi 92,14%, presisi 93%, *recall* 92%, dan *F1-score* 92% pada konfigurasi data dan teknik augmentasi yang sama. Analisis *confusion matrix* menunjukkan DenseNet121 dengan ukuran input 224×224 hanya menghasilkan 8 kesalahan prediksi dari 140 citra uji, sedangkan model dengan ukuran input 640×640 menghasilkan 11 kesalahan prediksi dari jumlah citra uji yang sama.
3. Analisa penerapan kinerja model YOLOv9 dengan ukuran input 224×224 menunjukkan performa dengan nilai akurasi sebesar 93%, presisi 93%, *recall* 93%, dan *F1-score* 93% pada seluruh kelas. Sedangkan model YOLOv9 dengan ukuran input 640×640 mencapai akurasi 89%, presisi 90%, *recall* 89%, dan *F1-score* 89% pada konfigurasi dataset dan strategi augmentasi yang sama. Hasil analisis *confusion matrix* menujukkan model

YOLOv9 dengan ukuran input 224×224 hanya menghasilkan 10 kesalahan prediksi dari 140 citra uji, sedangkan model dengan ukuran input 640×640 menghasilkan 15 kesalahan prediksi dari jumlah citra uji yang sama.

4. Berdasarkan hasil evaluasi pada penelitian ini, model DenseNet121 dengan ukuran input 224×224 merupakan konfigurasi model dengan akurasi yang paling tinggi dengan tingkat akurasi sebesar 94,29%, presisi 95%, *recall* 94%, dan *F1-score* 94%.

5.2 Saran

Adapun saran dari penelitian ini adalah sebagai berikut :

1. Dataset yang digunakan masih memiliki kemiripan visual antarspesies yang tinggi dan kurangnya variasi jenis gambar. Oleh karena itu, diperlukan penambahan data dengan variasi morfologi yang lebih kaya untuk setiap spesies.
2. Kegagalan model mengenali lebah tanpa *background* disebabkan *background bias* dan *texture bias*, yaitu model lebih belajar tekstur dan latar dibandingkan bentuk objek. Solusi yang disarankan meliputi penggunaan citra tanpa *background*, augmentasi variasi *background*, serta *split training* agar model lebih fokus pada fitur bentuk dan struktur lebah.

DAFTAR PUSTAKA

[1] C. Trotter, H. J. Griffiths, and R. J. Whittle, “Surveying the deep: A review of computer vision in the benthos,” *Ecol. Inform.*, vol. 86, p. 102989, May 2025, doi: 10.1016/j.ecoinf.2024.102989.

[2] T. Bhuiyan, R. M. Carney, and S. Chellappan, “Artificial intelligence versus natural selection: Using computer vision techniques to classify bees and bee mimics,” *iScience*, vol. 25, no. 9, p. 104924, Sep. 2022, doi: 10.1016/j.isci.2022.104924.

[3] Y. Gao *et al.*, “Application of machine learning in automatic image identification of insects - a review,” *Ecol. Inform.*, vol. 80, p. 102539, May 2024, doi: 10.1016/j.ecoinf.2024.102539.

[4] X. Zhang and G. Chen, “An Automatic Insect Recognition Algorithm in Complex Background Based on Convolution Neural Network,” *Trait. Signal*, vol. 37, no. 5, pp. 793–798, Nov. 2020, doi: 10.18280/ts.370511.

[5] N. Kumar, Nagarathna, and F. Flammini, “YOLO-Based Light-Weight Deep Learning Models for Insect Detection System with Field Adaption,” *Agriculture*, vol. 13, no. 3, p. 741, Mar. 2023, doi: 10.3390/agriculture13030741.

[6] Q. Xiang, X. Huang, Z. Huang, X. Chen, J. Cheng, and X. Tang, “Yolo-Pest: An Insect Pest Object Detection Algorithm via CAC3 Module,” *Sensors*, vol. 23, no. 6, p. 3221, Mar. 2023, doi: 10.3390/s23063221.

[7] K. Bjerge, C. E. Frigaard, and H. Karstoft, “Object Detection of Small Insects in Time-Lapse Camera Recordings,” *Sensors*, vol. 23, no. 16, p. 7242, Aug. 2023, doi: 10.3390/s23167242.

[8] A. Sharma, V. Kumar, and L. Longchamps, “Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species,” *Smart Agric. Technol.*, vol. 9, p. 100648, Dec. 2024, doi: 10.1016/j.atech.2024.100648.

[9] S. A. Putri, *Mengenal Tentang Lebah Madu Tanpa Sengat*. Bogor: PT INDOCEMENT TUNGGAL PRAKARSA Tbk, 2024.

[10] Richard Szeliski, *Computer Vision: Algorithms and Applications 2nd Edition*, 2nd ed. Switzerland: Springer Nature Switzerland AG, 2022.

[11] M. Goyal, T. Knackstedt, S. Yan, and S. Hassanpour, "Artificial intelligence-based image classification methods for diagnosis of skin cancer: Challenges and opportunities," *Comput. Biol. Med.*, vol. 127, p. 104065, Dec. 2020, doi: 10.1016/j.compbiomed.2020.104065.

[12] P. Aggarwal, N. K. Mishra, B. Fatimah, P. Singh, A. Gupta, and S. D. Joshi, "COVID-19 image classification using deep learning: Advances, challenges and opportunities," *Comput. Biol. Med.*, vol. 144, p. 105350, May 2022, doi: 10.1016/j.compbiomed.2022.105350.

[13] Wahyudi Setiawan, *Deep Learning Menggunakan Convolutional Neural Network: Teori dan Aplikasi*, 1st ed. Malang: Media Nusa Creative, 2020.

[14] A. R. Sigurðardóttir, Þ. Sverrisson, A. Jónsdóttir, M. Gudjónsdóttir, B. Þ. Elvarsson, and H. Einarsson, "Otolith age determination with a simple computer vision based few-shot learning method," *Ecol. Inform.*, vol. 76, p. 102046, Sep. 2023, doi: 10.1016/j.ecoinf.2023.102046.

[15] S. Haykin, "Neural Networks and Learning Machines," 2009.

[16] Siti Nurmain, Annisa Darmawahyuni, Ade Iriani Sapitri, Muhammad Naufal Rachmatullah, Firdaus, and Bambang Tutuko, *Pengenalan Deep Learning dan Implementasinya*. Palembang: Unsri Press, 2021.

[17] Pytorch Team, "Densenet," PyTorch. Accessed: Oct. 29, 2025. [Online]. Available: https://pytorch.org/hub/pytorch_vision_densenet/

[18] T. Chauhan, H. Palivela, and S. Tiwari, "Optimization and fine-tuning of DenseNet model for classification of COVID-19 cases in medical imaging," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100020, Nov. 2021, doi: 10.1016/j.jjimei.2021.100020.

[19] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," Jan. 28, 2018, *arXiv*: arXiv:1608.06993. doi: 10.48550/arXiv.1608.06993.

[20] Richard Johnson, *YOLO Object Detection Explained: Definitive Reference for Developers and Engineers*. Singapura: HiTeX Press, 2025.

[21] M. Hussain, "YOLOv1 to v8: Unveiling Each Variant–A Comprehensive Review of YOLO," *IEEE Access*, vol. 12, pp. 42816–42833, 2024, doi: 10.1109/ACCESS.2024.3378568.

[22] Ultralytics Team, "YOLOv9: A Leap Forward in Object Detection Technology," Ultralytics. Accessed: Nov. 01, 2025. [Online]. Available: https://docs.ultralytics.com/models/yolov9/

[23] Pytorch Team, "Automatic Mixed Precision," PyTorch. Accessed: Jan. 21, 2026. [Online]. Available: https://docs.pytorch.org/xla/release/r2.7/perf/amp.html

[24] Priyanto Hidayatullah and Refdinal Tubagus, "YOLOv9 Architecture Explained," Stunning Vision AI. Accessed: Nov. 01, 2025. [Online]. Available: https://article.stunningvisionai.com/yolov9-architecture

[25] Python Software Foundation, "What is Python? – Executive Summary," Python.org. Accessed: Oct. 29, 2025. [Online]. Available: https://www.python.org/doc/essays/blurb/

[26] Pytorch Team, "PyTorch," PyTorch. Accessed: Oct. 29, 2025. [Online]. Available: https://pytorch.org/projects/pytorch/

[27] J. Ma, C. Hu, P. Zhou, F. Jin, X. Wang, and H. Huang, "Review of Image Augmentation Used in Deep Learning-Based Material Microscopic Image Segmentation," *Appl. Sci.*, vol. 13, no. 11, p. 6478, May 2023, doi: 10.3390/app13116478.

[28] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *Pattern Recognit.*, vol. 137, p. 109347, May 2023, doi: 10.1016/j.patcog.2023.109347.

[29] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, p. 100258, Dec. 2022, doi: 10.1016/j.array.2022.100258.

[30] K. Malialis, D. Papatheodoulou, S. Filippou, C. G. Panayiotou, and M. M. Polycarpou, "Data augmentation on-the-fly and active learning in data stream

classification," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec. 2022, pp. 1408–1414. doi: 10.1109/SSCI51031.2022.10022133.

[31] F. Yudistira and A. R. Isnain, "Analisis Sentimen Terhadap Seleksi CPNS Tahun 2024 Berbasis Media Sosial X Menggunakan Algoritma Naïve Bayes".

[32] H. A. Santoso, B. Fandhi Safsalta, N. Febrianto, G. Wilujeng Saraswati, and S.-C. Haw, "Comparative analysis of convolutional neural network and DenseNet121 transfer learning in agriculture focusing on crop leaf disease identification," *Appl. Comput. Inform.*, Jun. 2024, doi: 10.1108/ACI-03-2024-0132.

[33] K. Mohanappriya, D. C. Vennila, and R. R. Joshua, "Comparative Evaluation of CNN Models for Precision Agriculture in Deep Learning-Based Weed Detection," *J. Neonatal Surg.*, vol. 14, no. 29, 2025.

[34] D. L. R. Putri and Supatman, "Klasifikasi Citra Jenis Hijab Menggunakan Densenet-121," *J. Inform. Dan Tek. Elektro Terap.*, vol. 13, no. 1, Jan. 2025, doi: 10.23960/jitet.v13i1.5698.

[35] Y. Hou, Z. Wu, X. Cai, and T. Zhu, "The application of improved densenet algorithm in accurate image recognition," *Sci. Rep.*, vol. 14, no. 1, p. 8645, Apr. 2024, doi: 10.1038/s41598-024-58421-z.

[36] H. Suparwito, Bernardus Galih Hersa Prakoso, Rosalia Arum Kumalasanti, and Agnes Maria Polina, "Real-Time Vehicle Detection and Air Pollution Estimation Using YOLOv9," *J. Sisfokom Sist. Inf. Dan Komput.*, vol. 14, no. 1, pp. 23–30, Jan. 2025, doi: 10.32736/sisfokom.v14i1.2339.

[37] A. R. Manga', H. Herawati, and P. Purnawansyah, "Utilization of Deep Learning YOLO V9 for Identification and Classification of Toraja Buffalo Breeds," *Ilk. J. Ilm.*, vol. 17, no. 1, pp. 12–19, Apr. 2025, doi: 10.33096/ilkom.v17i1.2349.12-19.

[38] J. Bento, T. Paixão, and A. B. Alvarez, "Performance Evaluation of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 for Stamp Detection in Scanned Documents," *Appl. Sci.*, vol. 15, no. 6, p. 3154, Mar. 2025, doi: 10.3390/app15063154.

[39] M. Barr, “Coupling the Power of YOLOv9 with Transformer for Small Object Detection in Remote-Sensing Images,” *Comput. Model. Eng. Sci.*, vol. 143, no. 1, pp. 593–616, 2025, doi: 10.32604/cmes.2025.062264.

[40] S. Rahman, F. Humayara, S. M. E. Rabbi, and M. Rashid, “Efficient Medical Image Retrieval Using DenseNet and FAISS for BIRADS Classification”.

[41] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, “A review of convolutional neural networks in computer vision,” *Artif. Intell. Rev.*, vol. 57, no. 4, p. 99, Mar. 2024, doi: 10.1007/s10462-024-10721-6.

[42] M. Arif, C. Yong, and A. Mahalanobis, “Background Invariant Classification on Infrared Imagery by Data Efficient Training and Reducing Bias in CNNs,” Feb. 09, 2022, *arXiv*: arXiv:2201.09144. doi: 10.48550/arXiv.2201.09144.