

**EVALUASI KINERJA *DEEP LEARNING*
UNTUK IDENTIFIKASI SPESIES KUPU-KUPU SUMATERA
BERBASIS AUGMENTASI DATA**

DISERTASI

Oleh

**RICO ANDRIAN
NPM: 2037061007**



**PROGRAM STUDI DOKTOR MIPA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026**

**EVALUASI KINERJA *DEEP LEARNING*
UNTUK IDENTIFIKASI SPESIES KUPU-KUPU SUMATERA
BERBASIS AUGMENTASI DATA**

Oleh

RICO ANDRIAN

Disertasi

Sebagai Salah Satu Syarat untuk Mencapai Gelar DOKTOR

Pada

**Program Studi Doktor MIPA
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026**

ABSTRAK

EVALUASI KINERJA DEEP LEARNING UNTUK IDENTIFIKASI SPESIES KUPU-KUPU SUMATERA BERBASIS AUGMENTASI DATA

Oleh

RICO ANDRIAN

Kupu-kupu merupakan salah satu indikator penting keanekaragaman hayati yang memiliki peran ekologis dalam proses penyerbukan dan pemantauan kualitas lingkungan. Identifikasi spesies kupu-kupu secara manual memerlukan keahlian taksonomi dan waktu yang relatif lama sehingga diperlukan sistem identifikasi otomatis yang akurat dan efisien. Penelitian ini bertujuan untuk mengevaluasi kinerja berbagai arsitektur *deep learning* dalam mengidentifikasi spesies kupu-kupu Sumatera berbasis citra digital serta menganalisis pengaruh teknik augmentasi data terhadap performa model.

Dataset yang digunakan terdiri atas delapan spesies kupu-kupu Sumatera yang berasal dari Taman Kupu-Kupu Gita Persada, Bandar Lampung. Penelitian membandingkan tujuh arsitektur *deep learning*, yaitu DenseNet201, ResNet50, Xception, MobileNet, InceptionV3, *Vision Transformer* (ViT), dan EfficientNetB0. Selain itu, diterapkan enam teknik augmentasi data, yaitu *Rotation*, *Flip*, *CutMix*, *Mixup*, *AugMix*, dan *LayerMix*. Evaluasi dilakukan menggunakan metrik akurasi klasifikasi, efisiensi komputasi, dan penggunaan memori.

Hasil penelitian menunjukkan bahwa seluruh model mampu melakukan identifikasi spesies kupu-kupu Sumatera dengan tingkat akurasi yang tinggi. DenseNet201 menghasilkan akurasi tertinggi sebesar 99,38% pada data asli, diikuti ResNet50 dan Xception sebesar 98,75%, serta MobileNet sebesar 97,50%. Penerapan augmentasi data terbukti meningkatkan kemampuan generalisasi model dan menghasilkan beberapa kombinasi arsitektur-augmentasi dengan akurasi tertinggi sebesar 99,38%, yaitu DenseNet201–*LayerMix*, MobileNet–*Flip*, MobileNet–*CutMix*, *Vision Transformer–Flip*, *Vision Transformer–CutMix*, Xception–*Flip*, Xception–*CutMix*, Xception–*Mixup*, Xception–*AugMix*, serta EfficientNetB0–*LayerMix*.

Hasil penelitian juga menunjukkan bahwa MobileNet memiliki keseimbangan terbaik antara akurasi, waktu inferensi, ukuran model, dan jumlah parameter sehingga sangat sesuai untuk implementasi pada perangkat bergerak. Penelitian ini berhasil menghasilkan sistem identifikasi otomatis spesies kupu-kupu Sumatera berbasis *deep learning* yang mendukung konservasi biodiversitas dan diimplementasikan dalam aplikasi mobile Butterfly Vision.

Kata kunci: *deep learning*, identifikasi spesies, kupu-kupu Sumatera, augmentasi data, *convolutional neural network*, *vision transformer*, aplikasi mobile.

ABSTRACT

PERFORMANCE EVALUATION OF DEEP LEARNING FOR SUMATRAN BUTTERFLY SPECIES IDENTIFICATION BASED ON DATA AUGMENTATION

By

RICO ANDRIAN

Butterflies are important biodiversity indicators that play significant ecological roles in pollination and environmental monitoring. Manual butterfly species identification requires taxonomic expertise and considerable time, creating a need for an accurate and efficient automated identification system. This study aims to evaluate the performance of various deep learning architectures for identifying Sumatran butterfly species from digital images and to investigate the impact of data augmentation techniques on model performance.

The dataset consisted of eight Sumatran butterfly species collected from the Gita Persada Butterfly Park, Bandar Lampung, Indonesia. Seven deep learning architectures were evaluated, including DenseNet201, ResNet50, Xception, MobileNet, InceptionV3, Vision Transformer (ViT), and EfficientNetB0. In addition, six data augmentation techniques were applied: Rotation, Flip, CutMix, Mixup, AugMix, and LayerMix. Model performance was assessed using classification accuracy, computational efficiency, and memory usage metrics.

The results demonstrated that all models were capable of accurately identifying Sumatran butterfly species. DenseNet201 achieved the highest accuracy of 99.38% on the original dataset, followed by ResNet50 and Xception with 98.75%, and MobileNet with 97.50%. Data augmentation significantly improved model generalization and produced several architecture–augmentation combinations that achieved the highest accuracy of 99.38%, namely DenseNet201–LayerMix, MobileNet–Flip, MobileNet–CutMix, Vision Transformer–Flip, Vision Transformer–CutMix, Xception–Flip, Xception–CutMix, Xception–Mixup, Xception–AugMix, and EfficientNetB0–LayerMix.

Furthermore, MobileNet provided the best balance between classification accuracy, inference time, model size, and parameter count, making it highly suitable for deployment on mobile devices. This study successfully developed an automated deep learning-based system for Sumatran butterfly species identification

and implemented it in the Butterfly Vision mobile application to support biodiversity conservation efforts.

Keywords: *deep learning, species identification, Sumatran butterflies, data augmentation, convolutional neural network, vision transformer, mobile application.*

HALAMAN PERSETUJUAN

Judul Disertasi : **EVALUASI KINERJA DEEP LEARNING
UNTUK IDENTIFIKASI SPESIES KUPU-KUPU
SUMATERA BERBASIS AUGMENTASI DATA**

Nama Mahasiswa : Rico Andrian

Nomor Pokok Mahasiswa : 2037061007

Jurusan : Doktor MIPA

Fakultas : Matematika dan Ilmu Pengetahuan Alam



1. Komisi Pembimbing

Prof. Admi Syarif, Ph.D.
NIP. 196701031992031003
Promotor

Faverisca R. Lumbantja, M.Si., Ph.D.
NIP. 198301102008121002
Ko-Promotor

Prof. Dr. Emantis Rosa, M. Biomed.
NIP. 195806151986032001
Ko-Promotor

2. Ketua Program Studi Doktor MIPA

Dr. Khoirina Nisa, M.Si.
NIP. 197407262000032001

HALAMAN PENGESAHAN

1. Tim Penguji

Ketua : Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 197110012005011002

Sekretaris : Dr. Khoirin Nisa, M.Si.
NIP. 197407262000032001

Anggota : Prof. Admi Syarif, Ph.D.
NIP. 196701031992031003



: Favorisen R. Lambanraja, M.Si., Ph.D.
NIP. 198301102008121002

: Prof. Dr. Emantis Rosa, M. Biomed.
NIP. 195806151986032001

: Dr. Aristoteles, S.Si., M.Si.
NIP. 198105212006041002

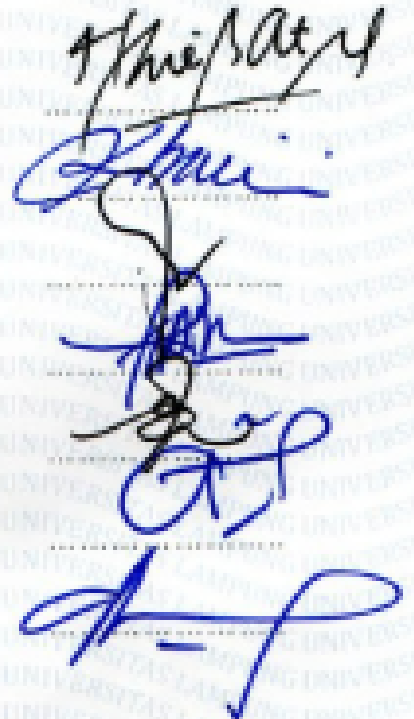
: Julian Supardi, M.T., Ph.D.
NIP. 197207102010121001

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 197110012005011002

Tanggal lulus ujian disertasi : 9 Juni 2026



PERNYATAAN ORISINALITAS DISERTASI

Saya dengan ini menyatakan dengan sesungguhnya bahwa sepanjang pengetahuan saya, disertasi ini merupakan hasil karya ilmiah asli saya sendiri dan belum pernah diajukan untuk memperoleh gelar akademik pada perguruan tinggi mana pun.

Selain itu, tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh pihak lain, kecuali yang secara jelas dikutip dan dicantumkan dalam daftar pustaka sesuai dengan kaidah penulisan ilmiah yang berlaku.

Demikian pernyataan ini saya buat dengan sebenar-benarnya untuk dapat dipergunakan sebagaimana mestinya.

Bandar Lampung, 12 Juni 2026



Yang menyatakan

Rico Andrian
NPM. 2037061007

RIWAYAT HIDUP



Rico Andrian lahir pada tanggal 27 Juni 1975. Penulis merupakan anak kelima dari lima bersaudara, pasangan Bapak Ing Sukaenda (alm.) dan Ibu Nuriah (almh.). Pendidikan dasar ditempuh di SD Negeri 4 Sukajawa, Bandar Lampung, dan diselesaikan pada tahun 1988. Pendidikan menengah pertama diselesaikan di SMP Negeri 3 Tanjungkarang. Pendidikan menengah atas diselesaikan di SMA Negeri 1 Garut, Jawa Barat, pada tahun 1993. Pendidikan Sarjana (S1) pada Program Studi Ilmu Komputer diselesaikan di Universitas Padjadjaran, Bandung, pada tahun 1999. Gelar Magister Komputer (M.Kom.) diperoleh dari Institut Pertanian Bogor (IPB), Bogor, Jawa Barat, pada tahun 2013. Saat ini, penulis berkarier sebagai dosen pada Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA), Universitas Lampung. Bidang minat penelitian yang ditekuni meliputi *Machine Learning* dan *Computer Vision* dengan fokus penerapan pada bidang biodiversitas. Penulis dapat dihubungi melalui surat elektronik pada alamat rico.andrian@fmipa.unila.ac.id.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga naskah disertasi ini, yang berjudul “Evaluasi Kinerja Deep Learning Untuk Identifikasi Spesies Kupu-Kupu Sumatera Berbasis Augmentasi Data”, dapat diselesaikan dengan baik. Disertasi ini disusun sebagai salah satu syarat untuk memperoleh gelar Doktor pada Program Studi Doktor MIPA UNILA, dengan fokus penelitian pada penerapan teknologi kecerdasan buatan dalam bidang kesehatan masyarakat.

Penelitian ini dilatarbelakangi oleh masalah stunting di Provinsi Lampung, yang menjadi masalah kesehatan masyarakat yang serius dan berdampak jangka panjang terhadap kualitas sumber daya manusia. Dengan memanfaatkan metode kecerdasan buatan, penelitian ini bertujuan untuk mengembangkan model prediksi dan analisis risiko stunting pada balita berbasis data antropometri, sehingga intervensi kesehatan dapat lebih tepat sasaran, efisien, dan berbasis bukti.

Penulis menyadari bahwa penyusunan naskah disertasi ini tidak lepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan penghargaan dan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. Ir. Lusmeilia Afriani, D.E.A.IPM. selaku Rektor Universitas Lampung.
2. Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.
3. Dr. Khoirin Nisa, M.Si. selaku Ketua Program Studi S3 MIPA Universitas Lampung.
4. Prof. Admi Syarif, Ph.D. selaku Promotor Disertasi.
5. Favorisen R. Lumbanraja, M.Si., Ph.D dan Prof. Dr. Emantis Rosa, M. Biomed selaku Ko-Promotor Disertasi.
6. Dr. Aristoteles, S.Si., M.Si. selaku dewan pembahas Internal Disertasi.
7. Julian Supardi, M.T., Ph.D. selaku pembahas/penguji Eksternal Disertasi.
8. Teman-teman Program Studi Doktor MIPA Unila.
9. Rekan-rekan Dosen Jurusan Ilmu Komputer, FMIPA Universitas Lampung.

7. Julian Supardi, M.T., Ph.D. selaku pembahas/penguji Eksternal Disertasi.
8. Teman-teman Program Studi Doktor MIPA Unila.
9. Rekan-rekan Dosen Jurusan Ilmu Komputer, FMIPA Universitas Lampung.
10. Rekan-rekan UPA. TIK Universitas Lampung.
11. Seluruh civitas akademika dari Unila yang tidak dapat saya sebutkan satu persatu yang telah memberikan bantuan, semoga Allah membalas amal kebaikan kita semua.

Semoga naskah disertasi ini dapat memberikan kontribusi ilmiah yang berarti, tidak hanya bagi pengembangan ilmu komputer khususnya bidang *machine learning* dan *computer vision*, tetapi juga bagi bidang biodiversitas di Indonesia.

Bandar Lampung, 12 Juni 2026

Penulis

Rico Andrian

DAFTAR ISI

	Halaman
ABSTRAK	i
ABSTRACT	iii
HALAMAN PERSETUJUAN	v
HALAMAN PENGESAHAN	vi
PERNYATAAN ORISINALITAS DISERTASI	i
RIWAYAT HIDUP	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	7
1.3 Tujuan Penelitian.....	7
1.4 Batasan Masalah.....	8
1.5 Manfaat Penelitian.....	9
1.6 Keterbaruan (<i>Novelty</i>)	10
TINJAUAN PUSTAKA	11
2.1 Identifikasi kupu-kupu menggunakan <i>deep learning</i>	11
2.1.1 <i>Deep Learning</i>	18
2.1.2 Arsitektur <i>deep learning</i> yang digunakan pada penelitian.....	23
2.1.3 <i>Hyperparameter</i>	25
2.1.4 <i>Confusion Matrix</i>	26
2.2 Augmentasi data pada <i>deep learning</i> untuk identifikasi kupu-kupu	28
METODOLOGI PENELITIAN	33
3.1 Identifikasi kupu-kupu Sumatera menggunakan <i>deep learning</i>	33
3.1.1 Pengumpulan Data	33
3.1.2 Pra-pemrosesan Data.....	34
3.1.3 Pembagian Data.....	35
3.1.4 Arsitektur <i>Deep Learning</i>	37

3.1.5	Konfigurasi <i>Hyperparameter</i>	39
3.1.6	Pengukuran Kinerja Model	41
3.2	Augmentasi data pada <i>deep learning</i> untuk identifikasi kupu-kupu Sumatera.....	43
3.2.1	Pembagian data untuk dataset yang menggunakan teknik augmentasi data	43
3.2.2	Teknik augmentasi data <i>flipping</i> pada dataset kupu-kupu Sumatera	45
3.2.3	Teknik augmentasi data <i>rotation</i> (rotasi) pada dataset kupu-kupu Sumatera	46
3.2.4	Teknik augmentasi data <i>CutMix</i> pada dataset kupu-kupu Sumatera.....	47
3.2.5	Teknik augmentasi data <i>Mixup</i> pada dataset kupu-kupu Sumatera	48
3.2.6	Teknik augmentasi data <i>AugMix</i> pada dataset kupu-kupu Sumatera.....	49
3.2.7	Teknik augmentasi data <i>LayerMix</i> pada dataset kupu-kupu Sumatera	50
3.3	Implementasi aplikasi mobile untuk identifikasi kupu-kupu	51
HASIL DAN PEMBAHASAN		55
4.1	Identifikasi Spesies Kupu-Kupu Sumatera Berbasis Citra Menggunakan <i>Deep Learning</i>	55
4.1.1	Desain Eksperimen.....	55
4.1.2	Hasil Evaluasi.....	55
4.1.3	Diskusi.....	65
4.2	Dampak Augmentasi Data terhadap Kinerja <i>Deep Learning</i> untuk Identifikasi Spesies Kupu-Kupu Sumatera.....	69
4.2.1	Hasil Eksperimen	69
4.3	Fitur aplikasi <i>mobile</i> untuk identifikasi kupu-kupu	73
SIMPULAN DAN SARAN		78
5.1	Simpulan.....	78
5.2	Saran.....	79
DAFTAR PUSTAKA		80
LAMPIRAN.....		90

DAFTAR GAMBAR

Gambar	Halaman
Gambar 1. Morfologi Kupu-kupu (Perveen & Khan, 2017).....	15
Gambar 2. Deep learning Family (Alzubaidi et al., 2021).....	18
Gambar 3. Perbedaan deep learning dan machine learning tradisional (Alzubaidi et al., 2021).....	19
Gambar 4. Lapisan Convolutional Neural Network (CNN).	21
Gambar 5. Tahapan Penelitian Identifikasi Kupu-kupu Sumatera dengan deep learning.....	34
Gambar 6. Contoh arsitektur CNN untuk klasifikasi citra (Alzubaidi et al.,)	37
Gambar 7. Metrik Evaluasi : (a) <i>Macro-Precision</i> , (b) <i>Macro-Recall</i> , (c) <i>Macro-F1-Score</i> , (d) <i>Micro-Precision</i> , (e) <i>Micro-Recall</i> , (f) <i>Micro-F1-Score</i> , (g) <i>Mean</i> , (h) <i>Standard deviation</i>	42
Gambar 8. Teknik augmentasi data flip horizontal pada citra original.....	45
Gambar 9. Teknik augmentasi data flip vertikal pada citra original.....	46
Gambar 10. Teknik augmentasi data rotasi 30^0 pada citra original.	47
Gambar 11. Teknik augmentasi data rotasi -30^0 pada citra original.....	47
Gambar 12. Teknik augmentasi data CutMix pada citra original.	48
Gambar 13. Teknik augmentasi data Mixup pada citra original.....	49
Gambar 14. Teknik augmentasi data AugMix pada citra original.....	50
Gambar 15. Teknik augmentasi data LayerMix (arithmetic, geometric, pixel dan element) pada citra original.....	51
Gambar 16. <i>Flowchart</i> pengembangan aplikasi <i>mobile</i> berbasis <i>deep learning</i>	52
Gambar 17. Kurva accuracy dan loss pada model deep learning.	57
Gambar 18. Kinerja confusion matrix 3 model tertinggi (DenseNet201, ResNet50, and Xception) and model terendah (EfficientNetB0).....	60
Gambar 19. Akurasi pada tujuh arsitektur deep learning.	63
Gambar 20. Grafik perbandingan akurasi tujuh arsitektur <i>deep learning</i> untuk data.....	70

Gambar 21. Antarmuka pengguna aplikasi Butterfly Vision..... 74

DAFTAR TABEL

Tabel	Halaman
Tabel 1. Penelitian-penelitian identifikasi kupu-kupu.....	11
Tabel 2. Pengukur kinerja klasifikasi.....	26
Tabel 3. Penelitian teknik augmentasi citra pada deep learning	28
Tabel 4. Pembagian dataset untuk pelatihan, validasi, dan pengujian	36
Tabel 5. Konfigurasi hyperparameter pelatihan yang digunakan	40
Tabel 6. Distribusi teknik augmentasi data pada setiap spesies.....	44
Tabel 7. Pembagian data pada dataset hasil augmentasi.....	44
Tabel 8. Kinerja pelatihan dan validasi pada epoch 20/20.....	58
Tabel 9. Perbandingan kinerja tiap spesies pada arsitektur dengan performa tiga terbaik. (DenseNet201, ResNet50, and Xception).....	61
Tabel 10. Precision, recall, F1-Score, and support per kelas pada EfficientNet-B0	62
Tabel 11. Kinerja statistik metrik pada skenario dua percobaan.	65
Tabel 12. Metriks komputasi pada tujuh arsitektur deep learning.	67
Tabel 13. Akurasi model deep learning pada data original dan augmentasi data .	71

PENDAHULUAN

1.1 Latar Belakang

Teknologi kecerdasan buatan (*Artificial Intelligence*) disingkat AI sudah masuk ke dalam aktifitas hidup manusia. Kecerdasan buatan mencakup *machine learning* dan *deep learning*. *Machine learning* (Pembelajaran Mesin) adalah bagian dari kecerdasan buatan yang menggunakan metode statistik yang memungkinkan mesin (komputer) untuk mengambil keputusan berdasarkan data. Algoritma *machine learning* dirancang agar dapat belajar dan kemampuannya meningkat seiring waktu ketika terdapat data baru dalam mengenali objek. *Deep learning* adalah jenis dari *machine learning* yang terinspirasi oleh fungsi sel otak manusia yang disebut neuron. Inspirasi tersebut mengandung konsep yang dinamakan *Neural Network* (NN) tapi pada *deep learning* NN-nya lebih dari dua lapisan (*layer*).

Algoritma yang baru pada saat ini dan cukup tangguh untuk mengenali objek adalah *deep learning*. *Deep learning* memungkinkan model komputasi untuk belajar representasi yang luar biasa kompleks, tajam, dan abstrak, mendorong kemajuan yang signifikan dalam berbagai masalah seperti pengenalan visual, deteksi objek, pengenalan ucapan, pemrosesan bahasa alami, analisis citra medis, pencarian obat-obatan dan genomik (Liu *et al.*, 2017). *Convolutional Neural Networks* (CNNs), model pembelajaran mendalam yang paling representatif, mampu memanfaatkan sifat yang mendasari sinyal alami: invariansi terjemahan, konektivitas lokal, dan hierarki komposisi (LeCun *et al.*, 2015).

Para peneliti sudah lebih dari satu dekade melakukan pengenalan spesies untuk melestarikan spesies yang terancam punah. Deteksi dan pengenalan spesies kupu-kupu secara otomatis akhir-akhir ini telah menarik perhatian para peneliti sehingga berbagai penelitian tentang proses deteksi objek dan pengenalan kupu-kupu telah dilakukan. Beberapa metode dan sistem dikembangkan untuk mengidentifikasi serangga secara akurat. Tugas identifikasi dan klasifikasi spesies kupu-kupu yang berbeda sangat kompleks, melelahkan, dan memakan waktu di antara semua spesies serangga. Identifikasi dan klasifikasi kupu-kupu tidak hanya penting di bidang *computer vision* tetapi juga tidak kalah pentingnya di bidang entomologi untuk melestarikan spesies kupu-kupu. Berbagai metode terutama pengolahan citra, *computer vision*, pembelajaran mesin tradisional, jaringan saraf, *deep learning*, *learning transfer* dan lainnya telah digunakan hingga sekarang untuk mendeteksi dan mengklasifikasikan kupu-kupu.

Identifikasi spesies kupu-kupu dibutuhkan oleh Taman Kupu-kupu Gita Persada untuk membedakan jenis spesies kupu-kupu. Metode yang biasa dilakukan oleh peneliti yang ada di Gita Persada dalam mengidentifikasi kupu-kupu hanya menggunakan mata manusia. Kemampuan mata manusia yang terbatas dalam membedakan corak pada kupu-kupu menjadi landasan dalam membuat klasifikasi kupu-kupu berbasis pengenalan pola (*pattern recognition*). Kupu-kupu termasuk ke dalam ordo *Lepidoptera*. *Lepidoptera* berarti sayap bersisik. Tubuh dan sayapnya yang berwarna. Warna pada sayap ditentukan oleh pigmen, struktur sisik dan jatuhnya sinar matahari. Corak pada kupu-kupu inilah yang membedakan antar spesies kupu-kupu. Lampung memiliki taman penangkaran kupu-kupu yang bernama Gita Persada. Gita Persada terletak di Jalan Wan Abdurrahman, Hutan, Kecamatan Hutan, Lampung, Indonesia. Gita Persada memiliki 211 spesies kupu-kupu yang dikembangbiakkan (Soekardi *et al.*, 2016). Penelitian terdahulu terkait dengan pengenalan pola untuk identifikasi kupu-kupu, salah satunya dilakukan oleh Andrian *et al.*, (2020) melakukan penelitian terhadap kupu-kupu yang ada di Taman Kupu-Kupu Gita Persada. Penelitian tersebut menggunakan 6 spesies kupu-kupu dan dataset yang digunakan berjumlah 600 citra sisi sayap bagian atas. Penelitian ini membuktikan bahwa dengan tahapan pra-pemrosesan (*scaling*, *segmentasi*, *grayscale*), citra kupu-

kupu dapat diekstraksi menggunakan pendekatan *Canny Edge Detection* maupun texture-based yaitu *Gray Level Co-occurrence Matrix* (GLCM). Keduanya menggunakan metode *K-Nearest Neighbor* (KNN) untuk klasifikasi, di mana penelitian berbasis deteksi tepi menghasilkan akurasi maksimal 80%, sementara penelitian berbasis tekstur (GLCM) menghasilkan akurasi yang lebih tinggi yakni 91,1%. Hal ini menegaskan bahwa pemilihan teknik ekstraksi fitur berpengaruh signifikan terhadap performa sistem identifikasi kupu-kupu.

Penelitian lain yang menggunakan klasifikasi *Convolutional Neural Network* (CNN) adalah identifikasi spesies kupu-kupu Malaysia, pengujian dilakukan menggunakan 120 citra kupu-kupu. Tingkat akurasi klasifikasi mencapai 97,5% (Arzar *et al.*, 2019). Dataset kupu-kupu Malaysia umumnya terdiri atas spesies tropis dari famili *Papilionidae*, *Nymphalidae*, *Pieridae*, dan *Lycaenidae*. Sebagian besar spesies memiliki pola warna yang kontras dan mudah dibedakan secara visual. Corak sayap didominasi oleh kombinasi warna hitam, kuning, biru, hijau, merah, dan putih dengan pola bercak besar atau pita warna yang jelas. Beberapa spesies memiliki ekor sayap (*swallowtail*) yang panjang sehingga menghasilkan karakter morfologi yang kuat untuk proses klasifikasi.

Penelitian identifikasi kelas kupu-kupu yang lain dilakukan pada kupu-kupu Turki. Penelitian ini mengusulkan model identifikasi otomatis kupu-kupu lapangan menggunakan *deep convolutional neural networks* dengan pendekatan *transfer learning*. Dataset berisi 44.659 citra dari 104 spesies kupu-kupu, namun karena distribusi data yang tidak seimbang, penelitian difokuskan pada 10 kelas dengan total 17.769 citra. Seluruh citra di-*resize* menjadi 224×224 piksel sebelum diproses. Tiga arsitektur CNN populer, yaitu VGG16, VGG19, dan ResNet50, dibandingkan dalam eksperimen. Hasil menunjukkan bahwa VGG16 memberikan performa terbaik dengan akurasi uji sekitar 79,5%, disusul VGG19 dengan 77,2%, sementara ResNet50 mengalami *overfitting* dengan akurasi uji hanya 70,2%. Studi ini menyimpulkan bahwa meskipun kondisi citra bervariasi (perbedaan sudut, jarak, *occlusion*, dan kompleksitas latar belakang), teknik *transfer learning* berbasis CNN tetap efektif untuk klasifikasi kupu-kupu lapangan dengan tingkat keberhasilan mendekati 80%. Dataset kupu-kupu Turki didominasi oleh spesies dari famili *Nymphalidae*, *Pieridae*, *Satyrinae*, dan

Lycaenidae yang hidup pada wilayah beriklim subtropis hingga temperate. Corak sayap banyak didominasi warna cokelat, abu-abu, krem, jingga pucat, dan hitam. Sebagian besar spesies memiliki pola *eyespot* (bintik menyerupai mata), garis melintang, pita tipis, dan bercak kecil yang sangat mirip antar spesies. (Almryad & Kutucu, 2020).

Penelitian spesies kupu-kupu India juga dilakukan menggunakan CNN pada tahun 2021. Penelitian ini membuat dataset kupu-kupu baru dengan 34024 citra kupu-kupu milik 315 spesies dari India. Metode yang diusulkan membuktikan efektivitas teknik augmentasi data baru pada dataset yang digunakan. Identifikasi kupu-kupu menggunakan gambar fotografi, kemudian membangun 11 model *deep learning* baru. Model *deep learning* yang dibangun untuk pengklasifikasian kupu-kupu menggunakan sebelas arsitektur baru *Deep Convolutional Neural Network* (DCNN) yaitu ResNet-18, ResNet-34, ResNet-50, ResNet-121 ResNet-152, AlexNet, DenseNet-121, DenseNet-161, VGG-16, VGG-19 dan SqueezeNet-v1.1. Hasil klasifikasi model yang berbeda dibandingkan dan teknik yang diusulkan mencapai akurasi top-1 maksimum (94,44%), top-3 akurasi (98,46%) dan akurasi 5 teratas (99,09%) menggunakan model ResNet-152, diikuti oleh model DenseNet-161 mencapai akurasi top-1 (94,31%), akurasi top-3 (98,07%) dan akurasi 5 besar (98,66%). Dataset kupu-kupu India memiliki keragaman morfologi yang sangat tinggi karena mencakup 315 spesies dari berbagai famili kupu-kupu tropis dan subtropis. Corak sayap menunjukkan variasi yang sangat luas mulai dari pola polos hingga pola kompleks yang terdiri atas garis radial, bercak simetris, pita warna, *eyespot*, pola mimikri, serta kombinasi warna metalik. Warna dominan meliputi hitam, kuning, jingga, merah, hijau, biru, dan putih. Banyak spesies memiliki karakter visual yang sangat khas sehingga model *deep learning* dapat membangun representasi fitur yang kuat. (Theivaprakasham, 2021).

Sebuah model *deep learning* membutuhkan banyak data agar model tersebut memiliki kinerja yang baik. Model *deep learning* berbeda dengan model *machine learning* tradisional. Model *deep learning* memiliki karakteristik yaitu semakin banyak data maka semakin meningkat juga kinerja model tersebut. Aturan tentang batas minimal berapa jumlah data yang dibutuhkan untuk melatih sebuah

model *deep learning* untuk mendapatkan akurasi tertentu belum ditemukan secara pasti. Jumlah data tersebut bergantung pada masalah yang akan diselesaikan. Contohnya pada kasus klasifikasi citra medis, saat menggunakan 100 data latih untuk klasifikasi citra otak manusia hanya mendapatkan akurasi sebesar 72,82% sedangkan pada kasus citra bahu manusia, dengan jumlah data yang sama mendapatkan akurasi 95,53% (Cho *et al.*, 2016). Model *deep learning* membutuhkan banyak data agar model tersebut memiliki kinerja yang baik maka dari itu dibutuhkan teknik untuk menambahkan data dari data yang sudah ada itu dengan teknik augmentasi data (*data augmentation*).

Data pelatihan yang beragam dan mencukupi merupakan faktor krusial untuk memastikan model *deep learning* mampu melakukan generalisasi dengan baik terhadap data baru. Klasifikasi spesies kupu-kupu yang datanya terbatas dan tidak seimbang antar kelas, pengumpulan data yang besar dan bervariasi sangat sulit dilakukan. Augmentasi data menjadi solusi penting untuk memperluas dan memperkaya distribusi data pelatihan secara artifisial. Augmentasi terbukti mampu meningkatkan kinerja model, mencegah *overfitting*, dan memperkuat model yang tangguh terhadap *noise* dan perubahan kondisi pencahayaan, sudut pandang, maupun latar belakang objek (Shorten & Khoshgoftaar, 2019) secara khusus menyatakan bahwa augmentasi bukan hanya praktik pelengkap, tetapi merupakan bagian integral dari *pipeline deep learning* modern karena mampu meningkatkan akurasi dan efisiensi pelatihan secara signifikan melalui strategi berbasis model dan optimisasi kebijakan augmentasi.

Penelitian oleh (Xu *et al.*, 2023), dilakukan telah komprehensif terhadap berbagai teknik augmentasi citra dalam konteks *deep learning*. Mereka mengkategorikan pendekatan augmentasi ke dalam tiga jenis utama, yaitu *model-free* (tanpa model), *model-based* (berbasis model), dan *optimizing policy-based* (berbasis kebijakan yang dioptimalkan). Studi ini memberikan panduan sistematis dalam memilih dan merancang metode augmentasi yang sesuai untuk meningkatkan kinerja model pembelajaran mendalam.

Penelitian lain melakukan pendekatan paling awal ditunjukkan oleh (Krizhevsky *et al.*, 2012) melalui pengembangan AlexNet pada dataset ImageNet. Mereka menggunakan teknik augmentasi seperti translasi (*translation*), *flipping*, dan

perubahan intensitas piksel untuk meningkatkan akurasi klasifikasi citra. Pendekatan ini sangat berpengaruh dan menjadi dasar berbagai metode augmentasi modern dalam CNN.

Kemajuan berikutnya diperkenalkan oleh Cubuk *et al.*, (2019) dengan metode *AutoAugment*, yang menggunakan *reinforcement learning* untuk mencari kebijakan augmentasi optimal dari data. Penelitian ini menerapkan pendekatan pada dataset CIFAR-10 dan ImageNet, *AutoAugment* terbukti secara signifikan meningkatkan akurasi model klasifikasi. Penelitian berikutnya untuk menyederhanakan proses pencarian, *RandAugment* dikembangkan oleh tim yang sama pada tahun 2020. *RandAugment* mengurangi ruang pencarian dengan memperbaiki skema probabilitas, menjadikannya solusi praktis untuk augmentasi otomatis.

Zhang *et al.*, (2018) memperkenalkan teknik *Mixup*, yang menghasilkan citra baru dari kombinasi linear antara dua citra dan labelnya. Teknik ini terbukti efektif dalam meningkatkan generalisasi model dan mengurangi *overfitting*. Pengembangan lanjutannya, *PuzzleMix* oleh Kim *et al.*, (2020), menggabungkan dua citra berdasarkan *saliency map* dan statistik lokal. *PuzzleMix* memberikan hasil yang lebih akurat dibandingkan *Mixup* standar, khususnya pada data kompleks.

Metode augmentasi berbasis pemodelan juga berkembang pesat. (Tanaka & Aranha, 2019) menggunakan *Generative Adversarial Networks (GAN)* untuk menghasilkan citra sintetis yang menyerupai data asli, membantu mengatasi masalah ketidakseimbangan kelas. (Xu *et al.*, 2022), disisi lain menerapkan pendekatan translasi gaya yang konsisten untuk pengenalan penyakit tanaman, membuktikan bahwa augmentasi berbasis gaya dapat meningkatkan performa klasifikasi secara signifikan.

Pendekatan lain yang terbukti efektif adalah *Cutout* yang diperkenalkan oleh DeVries *et al.*, (2017), di mana sebagian area citra dihapus secara acak selama pelatihan. Hal ini memaksa model untuk mempelajari fitur global yang lebih kuat. Teknik ini kemudian dikembangkan menjadi *CutMix* oleh (Yun *et al.*, 2012), yang memotong dan menggabungkan dua citra secara spasial serta menggabungkan labelnya, menghasilkan model dengan kemampuan pelokalan fitur yang baik.

Chen *et al.*, (2024) mengembangkan GridMask, untuk meningkatkan ketangguhan model terhadap oklusi yang menggunakan pola *masker grid* multiskala pada citra selama pelatihan. Metode ini terbukti meningkatkan ketahanan model terhadap variasi bentuk dan posisi objek. Penelitian yang lain dalam bidang deteksi objek, Bochkovskiy *et al.*, (2020) menggabungkan berbagai teknik augmentasi seperti *Mosaic*, *Scaling*, dan *Color Space transformation* dalam *framework* YOLOv4, yang menghasilkan model deteksi dengan kecepatan dan akurasi optimal.

Penelitian yang akan dilakukan adalah mengevaluasi kinerja *deep learning* untuk identifikasi spesies kupu-kupu Sumatera berbasis augmentasi data. Penggunaan istilah dalam penelitian ini, untuk “kelas” (*class*) dalam *machine learning* merujuk pada spesies biologis individu sehingga istilah “kelas” digunakan sebagai kategori target umum yang sesuai dengan tingkat taksonomi “spesies,” bukan “*Class*” pada tingkatan taksonomi yang lebih tinggi.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah:

1. Efektivitas penggunaan algoritma *deep learning* dalam mengidentifikasi spesies kupu-kupu Sumatera berdasarkan citra digital.
2. Perbedaan kinerja antara berbagai arsitektur *deep learning* dalam melakukan identifikasi spesies kupu-kupu Sumatera.
3. Pengaruh teknik augmentasi data terhadap akurasi model *deep learning* dalam identifikasi kupu-kupu Sumatera.
4. Model dan kombinasi teknik yang memberikan hasil paling optimal ditinjau dari aspek akurasi, efisiensi komputasi, dan konsumsi memori untuk identifikasi otomatis spesies kupu-kupu Sumatera berbasis *deep learning* dan augmentasi data.

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah mengevaluasi kinerja arsitektur-arsitektur *deep learning* dengan augmentasi data pada dataset kupu-kupu Sumatera.

1. Menganalisis efektivitas penerapan algoritma *deep learning* dalam mengidentifikasi spesies kupu-kupu Sumatera menggunakan citra digital sebagai data input.
2. Membandingkan kinerja berbagai arsitektur *deep learning* dalam identifikasi spesies kupu-kupu Sumatera.
3. Mengevaluasi pengaruh penerapan teknik-teknik augmentasi data terhadap akurasi model *deep learning* dalam identifikasi spesies kupu-kupu Sumatera.
4. Mengidentifikasi kombinasi terbaik antara arsitektur *deep learning* dan teknik augmentasi yang menghasilkan kinerja optimal dalam hal akurasi klasifikasi, efisiensi komputasi, dan penggunaan memori dalam sistem identifikasi otomatis spesies kupu-kupu Sumatera.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Penelitian ini difokuskan pada dataset kupu-kupu yang berasal dari Pulau Sumatera di Indonesia.
2. Model yang digunakan adalah *deep learning* dan teknik augmentasi data.
3. Dataset yang digunakan merupakan citra kupu-kupu endemik Sumatera yang berasal dari Taman Kupu-Kupu Gita Persada, Bandar Lampung. Ada delapan jenis kupu-kupu Sumatera yang sudah diambil citranya yaitu Kupu Raja Limau (*Papilio memnon*), Kupu Jojo (*Papilio nephelus*), Kupu Pantat Merah (*Pachliopta aristolochiae*), Kupu Batik Cap (*Cethosia penthesilea*), Kupu Raja Helena (*Troides helena*), Kupu Harimau Hijau Kuning (*Papilio peranthus*), Kupu Jarak (*Graphium doson*), Kupu Hijau Biru (*Graphium sarpedon*).
4. Alasan jumlah spesies kupu-kupu Sumatera hanya 8 spesies adalah memfokuskan pada delapan spesies memungkinkan peneliti untuk memastikan kualitas data yang lebih tinggi. Dataset ini memiliki kombinasi karakteristik yang unik karena mencakup spesies dengan warna kontras sekaligus spesies yang memiliki kemiripan morfologi tinggi. *Papilio memnon* memiliki sifat polimorfik dengan variasi warna hitam,

putih, dan biru. *Papilio nephelus* memiliki pita putih yang mencolok pada sayap hitam. *Papilio peranthus* memiliki kilauan hijau metalik dan biru neon. *Troides helena* memiliki kombinasi warna hitam dan kuning keemasan yang dominan. *Pachliopta aristolochiae* memiliki sayap hitam dengan abdomen merah cerah. *Cethosia penthesilea* memiliki warna jingga terang dengan pola menyerupai motif batik. *Graphium doson* memiliki pita kebiruan semi-transparan yang tersusun dari bercak-bercak memanjang. *Graphium sarpedon* memiliki pita biru toska tebal yang melintang di tengah sayap. Data yang lebih sedikit tetapi lebih terkontrol dan terannotasi dengan baik dapat menghasilkan model yang lebih akurat dan lebih dapat diandalkan. Ini juga mengurangi kemungkinan adanya bias atau kesalahan dalam data yang dapat mempengaruhi hasil penelitian. Pembatasan kelas juga memungkinkan model *deep learning* dapat lebih fokus dalam belajar fitur-fitur unik dari setiap kelas yang tersedia sehingga meningkatkan akurasi model dalam mengidentifikasi spesies kupu-kupu yang termasuk dalam dataset.

1.5 Manfaat Penelitian

Hasil penelitian ini dapat memberikan manfaat sebagai berikut :

1. Memberikan kontribusi ilmiah dalam penerapan algoritma *deep learning* untuk identifikasi spesies kupu-kupu Sumatera. Penelitian ini akan menambah pengetahuan akademik terkait efektivitas penggunaan *deep learning* dalam mengolah citra digital untuk tujuan identifikasi spesies, khususnya dalam bidang biodiversitas dan konservasi.
2. Menyediakan referensi perbandingan kinerja berbagai arsitektur *deep learning*. Hasil penelitian dapat menjadi acuan bagi peneliti dan praktisi dalam memilih arsitektur *deep learning* yang paling sesuai untuk identifikasi citra dengan karakteristik mirip kupu-kupu Sumatera, sehingga dapat meningkatkan akurasi dan efisiensi sistem.
3. Menghasilkan evaluasi empiris terhadap pengaruh teknik augmentasi data pada model *deep learning*. Penelitian ini akan memberikan bukti kuantitatif tentang efektivitas teknik augmentasi tertentu dalam

meningkatkan performa model, yang dapat digunakan sebagai pedoman dalam perancangan sistem identifikasi berbasis citra pada bidang lain.

4. Menemukan kombinasi optimal antara arsitektur *deep learning* dan teknik augmentasi untuk sistem identifikasi otomatis. Manfaat ini mencakup penerapan praktis dalam pembuatan sistem identifikasi kupu-kupu yang akurat, efisien secara komputasi, dan hemat penggunaan memori, sehingga sistem tersebut dapat diimplementasikan pada perangkat dengan sumber daya terbatas atau di lapangan.

1.6 Keterbaruan (*Novelty*)

Penelitian akan memberikan temuan baru berupa :

1. Penelitian ini merupakan studi pertama yang secara khusus menerapkan dan menganalisis efektivitas algoritma *deep learning* untuk identifikasi spesies kupu-kupu Sumatera berdasarkan citra digital dari lapangan (bukan spesimen laboratorium), yang selama ini masih minim eksplorasi di ranah biodiversitas lokal Indonesia.
2. Keterbaruan penelitian ini juga terletak pada perbandingan menyeluruh terhadap beberapa arsitektur *deep learning* yang berbasis CNN dan berbasis *Vision Tranformer* (ViT) dalam konteks spesifik identifikasi kupu-kupu Sumatera yang memiliki kompleksitas visual tinggi dan kondisi latar alami yang tidak terkontrol.
3. Penelitian ini memberikan kontribusi baru dengan mengevaluasi secara eksperimental pengaruh berbagai teknik-teknik augmentasi data pada kinerja *deep learning* terhadap akurasi klasifikasi pada dataset kecil dan tidak seimbang, yang umum dijumpai dalam penelitian konservasi spesies kupu-kupu Sumatera.
4. Penelitian ini mengimplementasikan aplikasi mobile sebagai kombinasi optimal antara arsitektur *deep learning* yang menggunakan teknik augmentasi yang menghasilkan akurasi tinggi, tetapi juga mempertimbangkan efisiensi komputasi dan penggunaan memori yang efisien.

TINJAUAN PUSTAKA

Bab ini menjelaskan berbagai penelitian yang telah dilakukan yang terkait dengan Identifikasi kupu-kupu Sumatera menggunakan *deep learning* dan augmentasi data pada *deep learning* untuk identifikasi kupu-kupu Sumatera.

2.1 Identifikasi kupu-kupu menggunakan *deep learning*

Penelitian tentang identifikasi spesies kupu-kupu menggunakan *machine learning* sudah dilakukan dengan dataset kupu-kupu yang ada di beberapa negara. *Machine learning* yang digunakan dari yang tradisional sampai *deep learning*. Tabel 1 memberikan informasi tentang penelitian-penelitian yang mengidentifikasi spesies kupu-kupu menggunakan *machine learning*.

Tabel 1. Penelitian-penelitian identifikasi kupu-kupu

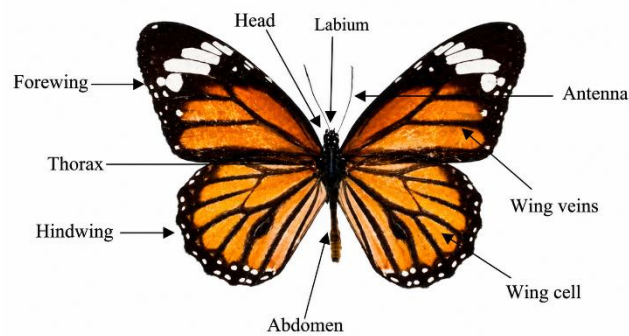
No.	Judul Penelitian	Peneliti	Dataset	Objek Penelitian	Hasil Penelitian
1	<i>Automatic identification of butterfly species using deep learning techniques</i>	(Liang <i>et al.</i> , 2020)	Citra kupu-kupu	Identifikasi dan klasifikasi kupu-kupu	Deteksi akurasi 98,35%, mAP 0,798 dalam klasifikasi, dengan menggunakan YOLO v3 dan teknik augmentasi data
2	<i>Deep learning techniques for automatic butterfly segmentation in ecological images</i>	(Tang <i>et al.</i> , 2020)	Citra kupu-kupu ekologi	Segmentasi kupu-kupu dalam citra ekologi	Penggunaan teknik <i>deep learning</i> menghasilkan segmentasi otomatis dengan akurasi yang tinggi pada citra ekologi

No.	Judul Penelitian	Peneliti	Dataset	Objek Penelitian	Hasil Penelitian
3	<i>Butterfly detection and classification based on integrated YOLO algorithm</i>	(Liang <i>et al.</i> , 2020)	1048 citra kupu-kupu	Deteksi dan klasifikasi kupu-kupu	Menggunakan YOLO terintegrasi dengan akurasi deteksi 98,35% dan mAP klasifikasi 0,798, serta evaluasi IoU dan penggunaan teknik label penuh dan label non-penuh
4	<i>Fine-grained visual categorization of butterfly specimens at subspecies level</i>	(Lin <i>et al.</i> , 2020)	Citra kupu-kupu subspecies	Kategorisasi visual kupu-kupu hingga level subspecies	Menggunakan CNN dengan skip-connections untuk klasifikasi rinci spesimen kupu-kupu hingga level subspecies dengan akurasi tinggi
5	<i>Rapid Fine-Grained Classification of butterflies based on FCM-KM and Mask R-CNN fusion</i>	(Tan <i>et al.</i> , 2020)	1188 citra kupu-kupu	Klasifikasi rinci kupu-kupu	Model gabungan FCM-KM dan <i>Mask R-CNN</i> dengan akurasi 83,62%, unggul dalam latar belakang yang kompleks
6	<i>Butterfly species identification using convolutional neural network (CNN)</i>	(Arzar <i>et al.</i> , 2019)	832 citra kupu-kupu	Identifikasi spesies kupu-kupu	Model CNN dengan akurasi 90%, mengusulkan pengembangan aplikasi berbasis <i>web</i> atau <i>mobile</i> untuk identifikasi spesies kupu-kupu secara real-time
7	<i>Automatic identification of butterfly species based on HoMSC and GLCMoIB</i>	(Li & Xiong, 2018)	Dataset citra kupu-kupu	Identifikasi otomatis spesies kupu-kupu	Menggunakan HoMSC dan GLCMoIB untuk identifikasi spesies dengan akurasi tinggi
8	<i>Arrhythmia classification based on improved monarch butterfly optimization algorithm</i>	(Nainwal <i>et al.</i> , 2022)	Citra kupu-kupu monarch	Klasifikasi aritmia menggunakan kupu-kupu <i>monark</i>	Menggunakan algoritma optimasi kupu-kupu <i>monark</i> yang telah diperbaiki untuk klasifikasi aritmia dengan akurasi tinggi
9	<i>Classification of butterfly species using traditional machine learning and deep learning methods</i>	(Zhu & Spachos, 2019)	<i>Leeds Butterfly Dataset</i>	Klasifikasi spesies kupu-kupu menggunakan metode kombinasi	PCA dengan SVM menghasilkan akurasi 52,8%, CNN 4-Conv model dengan data augmented mengatasi <i>overfitting</i> , VGG19 model hasil lebih baik dengan data augmented

No.	Judul Penelitian	Peneliti	Dataset	Objek Penelitian	Hasil Penelitian
9	<i>Classification of butterfly species using traditional machine learning and deep learning methods</i>	(Zhu & Spachos, 2019)	<i>Leeds Butterfly Dataset</i>	Klasifikasi spesies kupu-kupu menggunakan metode kombinasi	PCA dengan SVM menghasilkan akurasi 52,8%, CNN 4-Conv model dengan <i>data augmented</i> mengatasi <i>overfitting</i> , VGG19 model hasil lebih baik dengan <i>data augmented</i>
10	<i>Deep learning techniques for butterfly species classification and detection</i>	(Theivaprakasham, 2021b)	34,024 citra kupu-kupu	Klasifikasi dan deteksi spesies kupu-kupu	Model <i>deep learning</i> seperti ResNet-152, AlexNet, DenseNet-121, VGG-16 digunakan untuk klasifikasi dengan akurasi top-1 94,44%, top-3 98,46%, dan top-5 99,09%
11	<i>Extreme Learning Machine techniques for butterfly classification</i>	(Faruk <i>et al.</i> , 2015)	Dataset citra kupu-kupu	Klasifikasi spesies kupu-kupu menggunakan ELM	ELM dengan TEM mencapai akurasi 99,47%, SVM 98,16%, kNN 99,26%
12	<i>Using deep learning for butterfly species identification</i>	(Zhao <i>et al.</i> , 2019)	Dataset kecil citra kupu-kupu	Identifikasi spesies kupu-kupu dengan <i>deep learning</i>	Faster RCNN untuk identifikasi spesies kupu-kupu dengan akurasi 70,4%, membutuhkan dataset yang lebih besar dan <i>preprocessed images</i> untuk hasil yang lebih baik
13	<i>Identification of butterfly species using Content-Based Image Retrieval (CBIR)</i>	(Wang <i>et al.</i> , 2012a)	1104 citra kupu-kupu	Identifikasi kupu-kupu berdasarkan <i>family</i>	Metode CBIR dengan akurasi 84% untuk <i>family Papilionidae</i> , menggunakan ekstraksi fitur otomatis dan algoritma pencocokan template
14	<i>Butterfly species identification using Branch Length Similarity (BLS) entropy profile</i>	(Kang <i>et al.</i> , 2012)	50 citra kupu-kupu	Identifikasi spesies kupu-kupu berdasarkan bentuk sayap	Akurasi tinggi untuk spesies yang sama, metode BLS <i>entropy profile</i> berhasil diterapkan untuk identifikasi spesies kupu-kupu
15	<i>Automated framework for identifying butterflies and grasshoppers</i>	(Hassan <i>et al.</i> , 2014)	300 citra serangga (kupu-kupu dan belalang)	Identifikasi serangga berbasis warna dan bentuk	Akurasi 98% untuk belalang dan 84% untuk kupu-kupu, menggunakan SIFT dan SVM untuk

No.	Judul Penelitian	Peneliti	Dataset	Objek Penelitian	Hasil Penelitian
					deteksi fitur dan klasifikasi
16	<i>Image processing and machine learning techniques for butterfly identification</i>	(Kayci & Kaya, 2014)	190 citra kupu-kupu	Identifikasi spesies kupu-kupu menggunakan GLCM dan MLR	Akurasi 96,3% menggunakan GLCM untuk ekstraksi fitur tekstur dan MLR untuk klasifikasi
17	<i>Computer vision-based technique to recognize different butterfly species</i>	(Yilmaz et al., 2015)	190 citra kupu-kupu	Identifikasi spesies kupu-kupu berdasarkan fitur tekstur	Akurasi 89,47% menggunakan GLCM dan <i>Rough Set</i> untuk klasifikasi
18	<i>Automatic identification of butterfly species based on HoMSC and GLCMoIB</i>	(Li & Xiong, 2018)	750 citra kupu-kupu	Identifikasi spesies kupu-kupu berbasis tekstur dan bentuk	Akurasi 98% menggunakan HoMSC untuk ekstraksi fitur bentuk dan GLCMoIB untuk ekstraksi fitur tekstur
19	<i>Classification of butterfly species based on color and texture features</i>	(Kartika et al., 2017)	890 citra kupu-kupu	Klasifikasi spesies kupu-kupu menggunakan LBP dan HSV	Akurasi 72% menggunakan kombinasi fitur warna dan tekstur
20	<i>Detection of butterfly species using LBP and region props algorithm</i>	(Kartika et al., 2018)	890 citra kupu-kupu	Deteksi spesies kupu-kupu berbasis tekstur dan bentuk	Akurasi 66% menggunakan kombinasi fitur tekstur (LBP) dan bentuk (<i>region props</i>)
21	<i>Automatic butterfly species identification based on GLCM and weighted kNN</i>	(Xue et al., 2019)	150 citra kupu-kupu	Identifikasi otomatis spesies kupu-kupu berbasis tekstur	Akurasi 98% menggunakan GLCM untuk ekstraksi fitur tekstur dan <i>weighted kNN</i> untuk klasifikasi
22	<i>Identification of butterfly species based on GLCM and k-nearest neighbor (kNN)</i>	(Andrian et al., 2020)	600 citra kupu-kupu	Identifikasi spesies kupu-kupu berbasis tekstur	Akurasi 91,1% menggunakan GLCM untuk ekstraksi fitur tekstur dan <i>kNN</i> untuk klasifikasi
23	<i>Automatic butterfly detection and classification using YOLO algorithm</i>	(Liang et al., 2020)	1048 citra kupu-kupu	Deteksi dan klasifikasi kupu-kupu menggunakan YOLO	Akurasi deteksi 98,35% dan mAP klasifikasi 0,798, menggunakan teknik augmentasi data untuk meningkatkan performa
24	<i>Automatic identification of butterfly species using Faster R-CNN</i>	(Zhao et al., 2019)	5,695 citra kupu-kupu	Identifikasi spesies kupu-kupu menggunakan <i>Faster R-CNN</i>	mAP 70,4% menggunakan <i>Faster R-CNN</i> untuk ekstraksi fitur dan klasifikasi

Kupu-kupu memiliki beragam bentuk, pola, warna, dan tekstur sayap yang bervariasi. Ciri-ciri morfologi sayap kupu-kupu menjadi bagian penting dalam mengidentifikasi spesies. Fitur utama yang digunakan dalam identifikasi meliputi bentuk sayap, tekstur, dan warna. Proses identifikasi kupu-kupu secara tradisional dilakukan secara manual oleh ahli taksonomi, yang memerlukan pengetahuan profesional mendalam dan bisa menjadi tugas yang melelahkan dan rentan kesalahan. Aktifitas mengklasifikasikan kupu-kupu menggunakan kunci taksonomi dan DNA *sequencing*, diperlukan pemrosesan manual oleh ahli yang terlatih.



Gambar 1. Morfologi Kupu-kupu (Perveen & Khan, 2017)

Gambar 1 menunjukkan bagian utama morfologi kupu-kupu, termasuk kepala (*head*), *labium*, antena (*antenna*), urat sayap (*wing veins*), sel sayap (*wing cell*), perut (*abdomen*), sayap belakang (*hindwing*), sayap depan (*forewing*), dan dada (*thorax*) (Perveen & Khan, 2017). Ciri-ciri ini digunakan sebagai karakter taksonomi dalam identifikasi spesies kupu-kupu. Karakteristik bentuk, pola, ukuran, dan struktur morfologi pada setiap bagian tubuh digunakan sebagai ciri taksonomi dalam proses identifikasi dan klasifikasi spesies kupu-kupu Sumatera. Identifikasi spesies yang tampak serupa, pemeriksaan fitur struktural luar organ genital, terutama pada individu jantan, sangat penting. Proses ini sangat kompleks, memakan waktu, dan rentan terhadap kesalahan. Penelitian molekuler telah ditambahkan ke fitur identifikasi ini dalam beberapa tahun terakhir (Hebert & Gregory, 2005). Meskipun tidak selalu memberikan hasil yang akurat, penelitian saat ini, meskipun belum konklusif, mendukung penggunaan fitur morfologi dalam identifikasi kupu-kupu.

Identifikasi kupu-kupu berbasis morfologi adalah salah satu tugas yang paling menantang dalam penelitian taksonomi dan ekologi. Kemajuan pesat teknologi visi komputer (*computer vision*) serta pengenalan pola, identifikasi otomatis berbagai serangga mendapatkan banyak perhatian dalam beberapa tahun terakhir. (Kaya *et al.*, 2013) membuktikan bahwa spesies kupu-kupu dapat diidentifikasi secara akurat menggunakan teknik visi komputer. Informasi tentang tekstur dan warna dari citra kupu-kupu adalah karakteristik yang efektif untuk mengidentifikasi spesies kupu-kupu menggunakan teknik visi komputer.

Berbagai algoritma ekstraksi fitur digunakan untuk memperoleh fitur tekstur, misalnya *Grey Level Co-occurrence Matrices* (GLCM), *Laws' energy measures*, *Gabor filters*, dan *local binary patterns* (LBP). Fitur warna seperti *histograms of multiscale curvature* digunakan untuk mengekstrak fitur warna. Deskriptor berbasis bentuk seperti *Scale-Invariant Feature Transform* (SIFT) diterapkan, dan jaringan saraf dalam digunakan untuk klasifikasi citra yang mendetail.

Wang *et al.*, (2012) memperkenalkan metode untuk mengidentifikasi kupu-kupu di tingkat *family* menggunakan *Content-Based Image Retrieval* (CBIR). Metode ini mencapai akurasi 84% untuk *family Papilionidae* dengan menggunakan ekstraksi fitur otomatis dan algoritma pencocokan *template*. Kang *et al.*, (2012) menggunakan profil entropi *Branch Length Similarity* (BLS) untuk mengidentifikasi spesies kupu-kupu berdasarkan bentuk sayap. Metode ini terbukti berhasil secara statistik dan dapat diterapkan pada identifikasi rasial dan pengenalan serangga. Hassan *et al.* (2014) mempersembahkan *framework* otomatis untuk mengidentifikasi kupu-kupu dan belalang berdasarkan fitur warna dan bentuk. Sistem yang diusulkan mencapai akurasi 98% untuk belalang dan 84% untuk kupu-kupu.

Kayci dan Kaya (2014) menggunakan teknik pengolahan citra dan pembelajaran mesin dengan *Gray Level Co-occurrence Matrix* (GLCM) dan algoritma *Multinomial Logistic Regression* (MLR) untuk mengidentifikasi spesies kupu-kupu. Metode GLCM+MLR mencapai akurasi tertinggi 96,3% di antara teknik yang dibandingkan. Yılmaz *et al.*, (2015) menggunakan kombinasi fitur tekstur GLCM dan *Rough Set* (RS) untuk mengenali spesies kupu-kupu. Metode ini mencapai akurasi rata-rata tertinggi 89,47% untuk jarak 1 dan sudut 45 derajat.

Proses identifikasi berbasis morfologi ini telah menunjukkan keberhasilan yang signifikan, meskipun masih menghadapi tantangan karena variasi warna, bentuk, dan tekstur dalam *family* kupu-kupu. Metode *machine learning* tradisional digunakan untuk menyelesaikan masalah klasifikasi, klustering, regresi, dan reduksi dimensi. Algoritma yang terpisah digunakan untuk tujuan ekstraksi fitur untuk memperoleh fitur unik secara manual, dan algoritma klasifikasi yang terpisah digunakan untuk mengklasifikasikan kelas yang berbeda berdasarkan fitur input. Beberapa contoh algoritma *machine learning* tradisional adalah *Logistic Regression*, *Linear Regression*, *Naive Bayes*, dan *K-means clustering*.

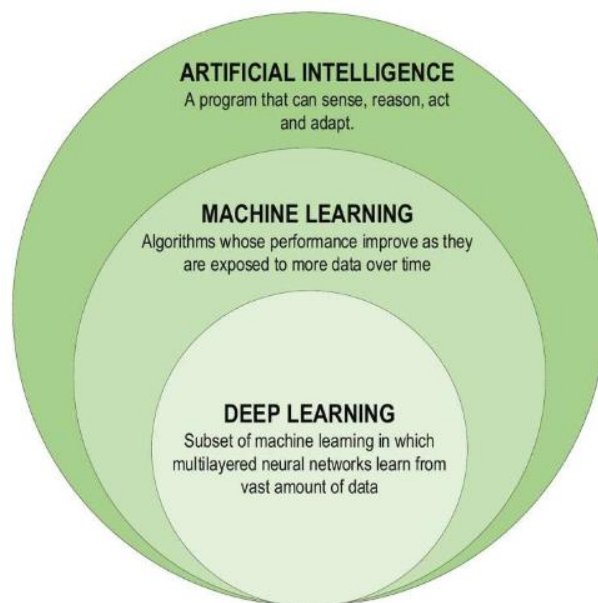
Wang *et al.*, (2012) memperkenalkan metode untuk mengidentifikasi kupu-kupu pada tingkat famili menggunakan *Content-Based Image Retrieval* (CBIR). Kang *et al.*, (2012) memperkenalkan pendekatan untuk mengidentifikasi spesies kupu-kupu menggunakan profil entropi *Branch Length Similarity* (BLS) dari *Unit Branching Network* (UBN) dengan partisi dan bobot. Mereka menggunakan bentuk sayap kupu-kupu sebagai fitur untuk mengidentifikasi spesies kupu-kupu. Metode yang diusulkan berhasil secara statistik dan dapat diterapkan dalam identifikasi rasial dan pengenalan serangga. Hassan *et al.*, (2014) menyajikan *framework* otomatis yang dapat mengidentifikasi kupu-kupu dan belalang berdasarkan fitur warna dan bentuk.

Kartika *et al.*, (2018) menyajikan teknik untuk mengklasifikasikan spesies kupu-kupu berdasarkan metode kuantisasi warna pada *HSV Color Space* untuk mengekstraksi fitur warna dan *Local Binary Pattern* (LBP) untuk mengekstraksi fitur tekstur. Kartika *et al.*, (2018) mengembangkan metode deteksi spesies kupu-kupu dari citra kupu-kupu menggunakan *Local Binary Pattern* (LBP) dan algoritma *region props*. Xue *et al.*, (2019) memperkenalkan proses identifikasi spesies kupu-kupu otomatis yang didirikan pada *Gray-Level Co-occurrence Matrix* (GLCM) dan pengklasifikasi kNN berbobot tanpa menggunakan seluruh citra kupu-kupu, yang meningkatkan kemampuan proses ekstraksi fitur. Andrian *et al.*, (2020) melakukan studi untuk mengidentifikasi enam spesies kupu-kupu yang berbeda berdasarkan fitur yang diekstraksi dari *Gray-Level Co-occurrence Matrix* (GLCM) dan mengklasifikasikannya menggunakan *k-nearest neighbor* (kNN).

Penelitian yang menggunakan metode *machine learning* tradisional untuk mengekstraksi fitur tekstur spesies kupu-kupu, sebagian besar peneliti menggunakan *Gray-level Co-occurrence Matrix* (GLCM). Beberapa peneliti juga menggunakan *Local Binary Pattern* (LBP) untuk mengekstraksi informasi tekstur. SVM, KNN, dan MLR digunakan sebagai pengklasifikasi oleh para peneliti. Akurasi pengklasifikasi KNN lebih tinggi daripada pengklasifikasi SVM yang dapat dilihat dari karya Li dan Xiong (2018) serta Xue *et al.*, (2019).

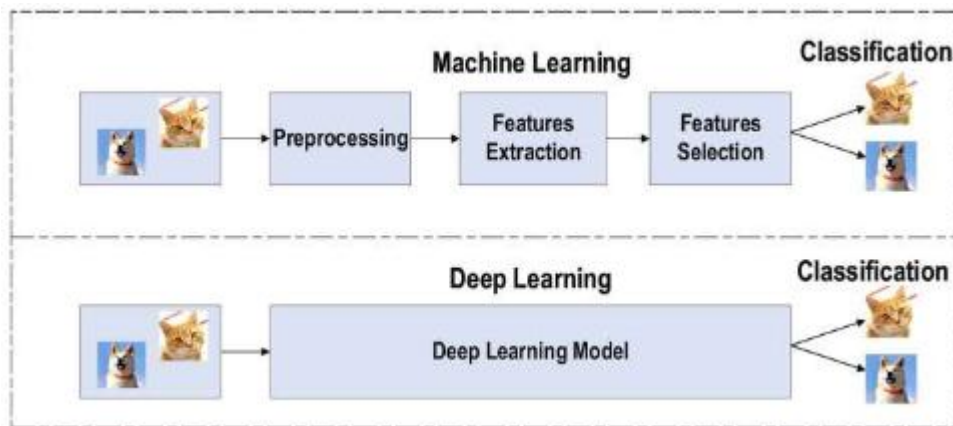
2.1.1 Deep Learning

Deep learning (DL) adalah cabang dari *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan (*deep neural networks*) untuk mengekstraksi fitur dan pola dari data. Teknik ini sangat efektif dalam memproses data dalam jumlah besar dan kompleks, seperti gambar, suara, dan teks. *Deep learning* telah membawa revolusi dalam bidang visi komputer, pemrosesan bahasa alami, dan banyak lagi, berkat kemampuannya untuk belajar representasi data yang lebih abstrak dan komprehensif.



Gambar 2. *Deep learning Family* (Alzubaidi *et al.*, 2021)

Deep learning adalah sebuah subset dari *machine learning* (Gambar 2), terinspirasi oleh pola pemrosesan informasi yang ditemukan di otak manusia. *Deep learning* tidak memerlukan aturan yang dirancang oleh manusia untuk beroperasi, sebaliknya, *deep learning* menggunakan sejumlah besar data untuk memetakan input yang diberikan ke label tertentu. *Deep learning* dirancang menggunakan banyak lapisan algoritma (jaringan saraf tiruan atau *Artificial Neural Networks* (ANNs) dimana masing-masing lapisan memberikan interpretasi yang berbeda terhadap data yang telah diberikan kepada mereka. *Deep learning* memungkinkan pembelajaran dan klasifikasi dilakukan dalam satu langkah, lihat Gambar 3. (Alzubaidi *et al.*, 2021).

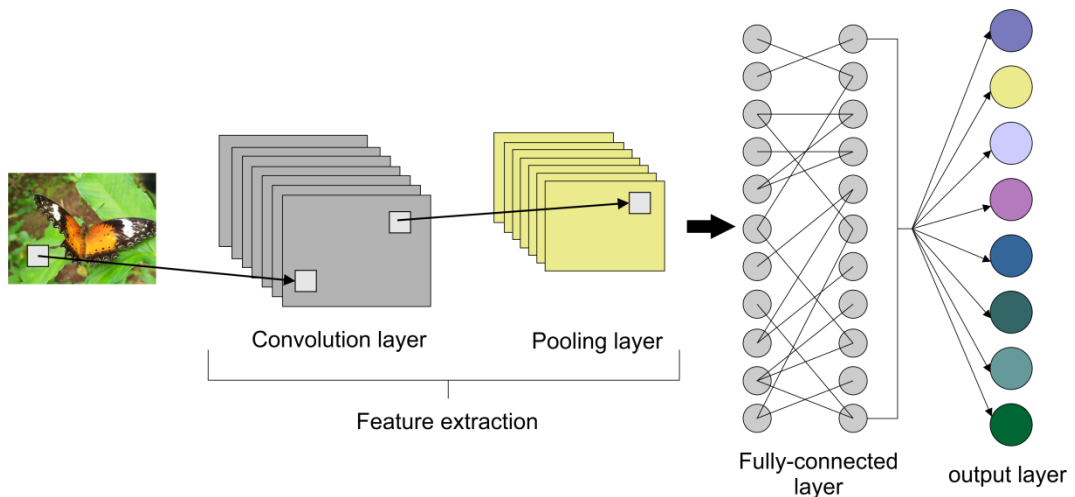


Gambar 3. Perbedaan *deep learning* dan *machine learning* tradisional (Alzubaidi *et al.*, 2021)

Convolutional Neural Network (CNN) merupakan tipe jaringan DL yang paling populer dan banyak digunakan dalam berbagai aplikasi, terutama yang melibatkan data citra digital. CNN dirancang dengan inspirasi dari struktur visual korteks otak manusia, yang memungkinkan ekstraksi fitur penting secara otomatis tanpa perlu intervensi manusia yang signifikan. CNN secara efektif menggunakan parameter yang lebih sedikit dibandingkan jaringan saraf tradisional dengan memanfaatkan konsep "parameter sharing" melalui operasi konvolusi, serta memiliki kemampuan tinggi dalam mendeteksi pola visual secara lokal maupun global pada citra.

Berbagai arsitektur CNN telah dikembangkan seperti AlexNet, ResNet, DenseNet, hingga arsitektur terbaru seperti *EfficientNet* dan *Vision Transformer* (ViT). Arsitektur-arsitektur ini memiliki berbagai perbaikan, termasuk dalam aspek struktur jaringan, optimisasi parameter, dan mekanisme regularisasi, yang bertujuan untuk meningkatkan kinerja klasifikasi, deteksi, maupun segmentasi citra secara signifikan. Kinerja DL juga memiliki sejumlah tantangan yang signifikan seperti tingginya kebutuhan data latih, ketidakseimbangan data (*imbalanced data*), permasalahan interpretabilitas, serta risiko *overfitting* dan *underfitting*. Solusi untuk mengatasi tantangan tersebut seperti augmentasi data (peningkatan variasi data melalui transformasi seperti rotasi, translasi, *flipping*, distorsi warna, dan pencahayaan), *transfer learning* (memanfaatkan *model pre-trained* untuk tugas baru), serta mekanisme regularisasi seperti *Dropout* dan *Batch Normalization* telah diusulkan dan banyak digunakan (Alzubaidi *et al.*, 2021a).

Convolutional Neural Network (CNN) merupakan salah satu algoritma yang digunakan pada *deep learning*. CNN dikembangkan sekitar tahun 1988 oleh Yann LeCun dan pertama kali digunakan untuk mengenali karakter seperti *zip codes* (kode pos) dan digit. Algoritma ini merupakan pengembangan lebih lanjut dari *Multilayer Perceptron* (MLP) yang memiliki banyak lapisan di dalamnya, untuk bisa memroses nilai-nilai *pixel* pada citra. MLP awalnya hanya bisa digunakan untuk mengenali karakter dengan kinerja yang masih minim. CNN hadir dengan perkembangan lebih lanjut dari MLP, tidak hanya mengenali karakter, CNN juga dapat mengenali citra dengan kinerja yang lebih baik dibandingkan MLP. Kinerja CNN dapat terlihat dengan adanya penambahan lapisan-lapisan tersembunyi untuk membantu identifikasi sebuah citra, serta susunan dari *neural network* (jaringan saraf) yang saling terhubung untuk menyatukan ciri sebuah citra (Alzubaidi *et al.*, 2021). Lapisan-lapisan awal pada CNN biasanya mendeteksi ciri dasar dari sebuah objek, seperti garis-garis vertikal, horisontal, kemudian output dari lapisan pertama akan dijadikan masukan di lapisan selanjutnya untuk mengekstraksi ciri yang lebih kompleks. Lapisan yang semakin dalam akan mendeteksi ciri objek yang semakin sulit juga, seperti bentuk dan warna yang semakin detail. Lapisan-lapisan dan *neural network* dapat dibidang sangat membantu dalam keseluruhan proses CNN. Lapisan di dalam CNN dapat dilihat pada Gambar 4.



Gambar 4. Lapisan *Convolutional Neural Network* (CNN).

2.1.1.1 *Feature extraction layer*

Feature extraction layer (lapisan ekstraksi fitur atau ciri) ialah *layer* atau lapisan di dalam CNN yang bertugas melakukan ekstraksi ciri dan menyimpannya menjadi nilai-nilai *pixel* dalam sebuah *array*. Citra masukan tersusun atas nilai *pixel* berbentuk *array* multidimensional, karena ukuran *array* tersebut terlalu besar untuk diproses, sehingga diperlukan filter untuk mengambil beberapa ciri utama dari citra tersebut. Ekstraksi fitur menyediakan *layer* atau lapisan-lapisan termasuk di dalamnya beberapa filter yang akan melalui proses perhitungan dengan perkalian matriks antara citra masukan dan filternya. Proses perkalian akan diproses lagi untuk lebih menyederhanakan nilai ciri yang akan diproses dalam klasifikasi. Inti dari ekstraksi fitur selain mengambil ciri dari citra, tetapi juga menyederhanakan bentuk dan nilai *pixel* yang dimiliki oleh citra masukan tanpa menghilangkan ciri yang penting dari objek (Almryad & Kutucu, 2020). Lapisan yang dilewati dalam *feature extraction* diantaranya *convolution layer* dan *pooling layer*.

2.1.1.2 *Convolution Layer*

Convolution layer atau lapisan konvolusi merupakan proses ketika menyederhanakan nilai pada citra masukan, sekaligus menyimpan ciri citra melalui nilai *pixel* yang dihasilkan dari penyederhanaan. Penyederhanaan

dilakukan melalui perkalian antara matriks masukan dan filter. Citra masukan akan memiliki nilai *pixel* berbentuk *array* multidimensional yang harus disederhanakan, baru kemudian dilakukan proses penyederhanaan nilai tersebut tanpa menghilangkan informasi utama ciri objeknya menggunakan sebuah filter. Hasil dari lapisan konvolusi ini bisa juga disebut sebagai *feature map*.

Pooling layer atau lapisan pooling berfungsi untuk mengurangi ukuran dari kumpulan ciri hasil konvolusi atau mengurangi ukuran dari citra masukan (Borza *et al.*, 2018). Pengurangan ukuran tersebut ditujukan agar proses komputasi lebih ringan untuk dijalankan saat memproses data (Alzubaidi *et al.*, 2021). Proses pooling biasanya mengambil nilai terbesar (*max pooling*) atau nilai rata-rata (*average pooling*) *pixel* dari *feature map* yang dihasilkan *convolution layer* (Irfansyah *et al.*, 2021). Nilai *pixel* yang diambil akan merepresentasikan ciri utama dari sebuah objek, sehingga tidak mengurangi informasi ciri objek meskipun ukuran kumpulan ciri citra terus berkurang. *Pooling layer* juga ditujukan agar ciri yang akan diproses untuk klasifikasi, merupakan ciri utama dan tidak menyisakan ciri lain yang tidak berkaitan dengan objek yang dimaksudkan.

2.1.1.3 Fully Connected Layer

Fully connected layer ialah lapisan dalam CNN yang digunakan untuk tahapan klasifikasi objek citra. *Fully connected layer* juga merupakan tahapan terakhir dalam CNN hingga bisa didapatkan hasil klasifikasi terhadap kelas objek. Masukan yang bisa diterima *fully connected layer* haruslah berbentuk vektor atau satu dimensi, sehingga perlu dilakukan *flatten* atau perataan bentuk *feature map* sebelum masuk ke tahap klasifikasi (Almryad & Kutucu, 2020). *Fully connected layer* terdiri atas *weights* (bobot) dan bias, serta terdapat neuron yang nantinya akan saling terhubung dengan menggunakan nilai bobot dan bias. *Neuron* yang terhubung sama seperti susunan puzzle, setiap ciri akan saling terhubung hingga bisa didapatkan hasil klasifikasi kelas objek.

2.1.1.4 Rectified Linear Unit (ReLU) Activation

ReLU *activation* atau aktivasi ReLU digunakan untuk memperkenalkan dan mengajarkan fungsi non-linear ke dalam jaringan konvolusi. ReLU biasanya digunakan dalam *convolution layer* ataupun *pooling layer*. Fungsi non-linear ialah kelebihan dari CNN, sehingga bisa memisahkan ciri maupun kelas yang membutuhkan lebih dari sekedar garis lurus (linear). ReLU akan mengembalikan nilai negatif yang dimiliki konvolusi menjadi 0 dan mengembalikan nilai positif menjadi nilai itu sendiri. Proses tersebut yang menjadi ciri utama dari aktivasi ReLU, serta membuat performa ReLU lebih baik diantara aktivasi lainnya. Fungsi yang digunakan dalam ReLU yaitu $f(x) = \max(0, x)$ (Ilahiyah & Nilogiri, 2018).

2.1.1.5 Softmax Activation

Softmax activation atau aktivasi *softmax* ialah fungsi matematis yang digunakan untuk mengubah angka vektor menjadi sebuah probabilitas vektor. *Softmax* biasanya digunakan saat proses klasifikasi dan penentuan kelas objek, setiap node neuron atau ciri objek yang terhubung dan telah dijumlahkan bobot-bobotnya, akan memiliki nilai probabilitas yang menentukan kelas objek tersebut. *Softmax* akan mengubah output berupa jumlah bobot menjadi sebuah nilai probabilitas. Output dari *softmax* diartikan sebagai probabilitas atau kemungkinan kelas dari sebuah objek. *Softmax* sesuai digunakan untuk *multiclass classification* atau klasifikasi banyak kelas yang biasanya lebih dari dua kelas.

2.1.2 Arsitektur *deep learning* yang digunakan pada penelitian

Penelitian ini menggunakan tujuh arsitektur *deep learning* dipilih untuk mengevaluasi kinerjanya dalam identifikasi spesies kupu-kupu Sumatera berbasis citra, yaitu DenseNet201, InceptionV3, MobileNet, ResNet50, Xception, EfficientNetB0, dan *Vision Transformer* (ViT). Arsitektur tersebut merepresentasikan berbagai keluarga model yang umum digunakan dengan karakteristik desain yang berbeda, mencakup koneksi residual pada ResNet50, konektivitas padat pada DenseNet201, konvolusi *depthwise separable* pada MobileNet dan Xception, *compound scaling* pada EfficientNetB0, serta

mekanisme berbasis perhatian pada ViT. Pemilihan ini bertujuan menyediakan perbandingan terkontrol antara model berbasis *Convolutional Neural Network* (CNN) dan model berbasis transformer dalam pengaturan eksperimen yang konsisten, bukan untuk menetapkan satu arsitektur sebagai yang paling optimal secara universal.

Convolutional Neural Network memproses data visual melalui serangkaian lapisan yang secara otomatis mempelajari fitur secara hierarkis, sebagaimana ditunjukkan pada Gambar 3. Citra masukan, khususnya foto kupu-kupu, memasukkan data piksel mentah ke dalam lapisan konvolusi yang mendeteksi pola dasar seperti tepi, tekstur, dan bentuk. Lapisan ReLU menambahkan non-linearitas untuk menangkap hubungan yang lebih kompleks. Lapisan *pooling* mengurangi dimensi peta fitur, merangkum informasi penting, meningkatkan ketahanan terhadap variasi, serta mengurangi kebutuhan komputasi. Fitur tingkat tinggi tersebut kemudian diratakan dan diteruskan ke lapisan *fully connected* untuk proses klasifikasi. Lapisan keluaran menghasilkan distribusi probabilitas antar kelas yang mencerminkan tingkat kepercayaan prediksi. Arsitektur ini banyak digunakan dalam tugas pengenalan citra (Alzubaidi *et al.*, 2021).

DenseNet201 meningkatkan penggunaan kembali fitur melalui koneksi padat, sehingga memperlancar aliran informasi dan gradien serta menghasilkan model yang lebih ringkas dengan kinerja yang lebih baik (Yasmin *et al.*, 2023). InceptionV3 memanfaatkan konvolusi terfaktorisasi dan *auxiliary classifier* untuk meningkatkan efisiensi komputasi serta mengurangi overfitting, sehingga sesuai untuk berbagai tugas pengenalan citra (Spiesman *et al.*, 2021). MobileNet dirancang untuk aplikasi visi pada perangkat bergerak dan sistem tertanam dengan memanfaatkan konvolusi *depthwise separable* guna menekan biaya komputasi dan ukuran model, tanpa mengorbankan akurasi secara signifikan (Adityawan *et al.*, 2023). ResNet50 mengatasi masalah *vanishing gradient* pada jaringan dalam melalui koneksi residual, sehingga memungkinkan pelatihan model yang lebih dalam dengan akurasi yang lebih tinggi (Karim *et al.*, 2024). Xception menggunakan konvolusi *depthwise separable* dan menunjukkan kinerja yang kuat, termasuk pencapaian akurasi validasi tinggi pada tugas klasifikasi citra (Hasan *et al.*, 2024). EfficientNetB0 dikenal dengan metode *compound scaling*, yang

memungkinkan pencapaian kinerja tinggi dengan jumlah parameter yang lebih sedikit melalui penskalaan yang konsisten pada kedalaman, lebar, dan resolusi jaringan. *Vision Transformer* (ViT) sebagai arsitektur yang lebih baru memproses citra dengan membaginya menjadi *patch* berukuran tetap, memperlakukannya sebagai urutan, serta memanfaatkan mekanisme *self-attention* untuk menangkap ketergantungan jarak jauh, berbeda dengan pendekatan lokal pada CNN (Pucci *et al.*, 2025). Arsitektur lain yang umum digunakan, seperti VGGNet atau varian transformer skala besar yang lebih baru, tidak disertakan dalam penelitian ini untuk menjaga ruang lingkup eksperimen tetap terkendali dan memastikan konsistensi protokol pelatihan pada seluruh model dalam kondisi dataset yang sama. Pendekatan ini memungkinkan perbandingan yang lebih terfokus dalam kerangka eksperimen terkontrol, terutama dengan mempertimbangkan keterbatasan ukuran dataset.

2.1.3 *Hyperparameter*

Hyperparameter merupakan jenis konfigurasi atau pengaturan terhadap model sebelum melakukan proses pelatihan data atau training. Nilai *hyperparameter* tidak bisa dimunculkan secara otomatis oleh data dalam model, tetapi didefinisikan secara manual oleh praktisi, itulah yang membedakannya dengan parameter (Afaq *et al.*, 2020) Parameter akan muncul selama proses training dan tidak bisa ditentukan secara manual nilainya, seperti nilai bobot pada *neural network*. *Hyperparameter* yang telah diatur sebelum proses training, akan menentukan kemunculan nilai-nilai parameter yang digunakan selama proses training agar bisa melakukan prediksi data dengan baik. Nilai parameter hanya muncul melalui data-data yang diberikan termasuk di dalamnya *hyperparameter*, sehingga *hyperparameter* juga menjadi penentu performa dan keberhasilan model dalam memprediksi. *Hyperparameter* yang paling umum dan sering digunakan dalam model CNN yaitu, *learning-rate*, *batch-size*, *optimizer*, dan *epoch*. *Learning-rate* digunakan untuk menghitung nilai koreksi terhadap bobot pada saat proses training berjalan (Rochmawati *et al.*, 2021). *Hyperparameter* ini mengacu kepada tingkat kecepatan algoritma atau ukuran langkah yang perlu diambil oleh *gradient descent* untuk mencapai titik optimal model.

Batch-size merupakan pengambilan sejumlah sample dari data training yang akan dibentuk menjadi kelompok atau batch. Tiap *batch* diambil untuk digunakan dalam melakukan proses satu iterasi yaitu tahap pembacaan data secara *forward* (ke depan) dan *backward* (ke belakang) (Rochmawati *et al.*, 2021). *Batch-size* mempengaruhi kecepatan model dalam belajar dan kestabilan model tersebut. *Optimizer Adam* termasuk optimizer yang efektif dan dapat melatih data dalam waktu yang cukup singkat. Optimizer ini merupakan pengembangan dari algoritma *Stochastic Gradient Descent* (SGD) klasik yang telah mengalami pembaharuan bobot jaringannya (Rochmawati *et al.*, 2021). *Epoch* merupakan nilai yang menentukan jumlah proses training dilakukan. *Epoch* terhitung ketika model telah berhasil membaca keseluruhan data training (Afaq *et al.*, 2020). Satu *epoch* dihitung ketika semua data training tiap *batch* selesai dipelajari oleh model.

2.1.4 Confusion Matrix

Confusion matrix biasanya digunakan untuk menunjukkan ringkasan performa dan hasil prediksi dari klasifikasi objek. Perhitungan *confusion matrix* dapat memberikan gambaran seberapa akurat hasil klasifikasi objek dan *error* yang dihasilkan oleh model. Jumlah prediksi benar dan salah akan dihitung dalam *confusion matrix* untuk kemudian menghasilkan prediksi keseluruhan maupun tiap kelasnya. *Confusion matrix* terdiri dari *accuracy*, *precision*, *recall*, dan *f1-score*. *Confusion matrix* memiliki ukuran penilaian melalui empat pengukur kinerja klasifikasi, *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) (Maxwell *et al.*, 2021). Gambaran umum metrik tersebut dapat dilihat pada Tabel 2.

Tabel 2. Pengukur kinerja klasifikasi

		Hasil Prediksi	
		<i>Positive</i>	<i>Negative</i>
Data Sebenarnya	<i>Positive</i>	TP	FN
	<i>Negative</i>	FP	TN

Keterangan:

True Positive : model dengan benar memprediksi sebuah kelas (positive) sebagai kelas yang sebenarnya (*positive*).

True Negative : model dengan benar memprediksi sebuah kelas (negative) sebagai kelas lain (*negative*).

False Positive : model salah memprediksi kelas lain (negative) sebagai kelas yang sebenarnya (*positive*).

False Negative : model salah memprediksi kelas yang sebenarnya (positive) sebagai kelas lain (*negative*).

Accuracy ialah penilaian yang dapat menggambarkan performa model dalam klasifikasi semua kelasnya. Penghitungan dilakukan dengan menjumlahkan metrik prediksi benar dan membaginya dengan total keseluruhan prediksi.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots\dots\dots (Maxwell et al., 2021).$$

Precision digunakan untuk menghitung hasil prediksi, berupa prediksi yang benar terhadap sebuah kelas yang sebenarnya dan prediksi yang keliru atau salah sasaran terhadap sebuah kelas yang sebenarnya. *Precision* mengukur akurasi model dalam mengklasifikasi sebuah kelas sebagai kelas yang sebenarnya, maupun kelas lain yang terklasifikasi sebagai kelas yang sebenarnya. Model akan memiliki *precision* yang besar ketika hasil prediksi benar (TP) semakin besar dan hasil prediksi salah (FP) semakin kecil.

$$Precision = \frac{TP}{TP + FP} \dots\dots\dots (Maxwell et al., 2021)$$

Recall digunakan untuk mengukur kemampuan model mendeteksi kelas yang sebenarnya. Hasil deteksi yang dilihat berupa kelas sebenarnya yang terprediksi sebagai kelas lain maupun kelas yang benar terprediksi sebagai kelas sebenarnya. *Recall* hanya mementingkan tingkat keberhasilan kelas sebenarnya terprediksi, baik sebagai kelas tersebut maupun sebagai kelas lain.

$$Recall = \frac{TP}{TP + FN} \dots\dots\dots (Maxwell et al., 2021)$$

F1-Score digunakan untuk mengukur mean dari *precision* dan *recall*. Indikasi yang ditampilkan oleh *F1-Score* yaitu model klasifikasi memiliki nilai *precision* dan *recall* yang baik. *F1-Score* biasanya digunakan ketika dataset yang digunakan dalam pelatihan model tidak seimbang jumlahnya. Perhitungan ini dapat sebagai nilai pembandingan antara algoritma yang digunakan dalam melatih data.

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \dots\dots\dots (Maxwell et al., 2021)$$

2.2 Augmentasi data pada *deep learning* untuk identifikasi kupu-kupu

Penelitian tentang teknik augmentasi citra pada *deep learning* sudah dilakukan dengan dataset citra. Tabel 3 memberikan informasi tentang penelitian-penelitian teknik augmentasi citra pada *deep learning*.

Tabel 3. Penelitian teknik augmentasi citra pada *deep learning*

No.	Judul Penelitian	Peneliti	Dataset	Objek Penelitian	Hasil Penelitian
1	<i>LayerMix: Enhanced Data Augmentation for Robust Deep Learning</i>	(Ahmad et al., 2026)	CIFAR-10, CIFAR-100, ImageNet-200, dan ImageNet-1K	Pengembangan teknik augmentasi data bernama <i>LayerMix</i> yang dirancang untuk meningkatkan ketahanan (robustness) model deep learning terhadap berbagai gangguan data, seperti korupsi gambar dan pergeseran domain	Efisiensi Ketahanan: Menurunkan mean corruption error (mCE) sebesar 14,26% pada ImageNet-200 dan hingga 19,7% pada CIFAR-100 dibandingkan dengan baseline. Akurasi & Rendisi: Meningkatkan akurasi data bersih sebesar 0,5% dan ketahanan rendisi sebesar 8,13% di atas baseline. Keamanan ML: Terbukti lebih unggul dan konsisten dalam berbagai tugas keamanan pembelajaran mesin dibandingkan teknik augmentasi standar lainnya.
2	<i>Image augmentation techniques for deep learning</i>	(M. Xu et al., 2023)	Beragam dataset citra	Teknik augmentasi citra	Mengategorikan teknik augmentasi citra menjadi tiga: <i>model-free</i> , <i>model-based</i> , dan <i>optimizing policy-based</i> . Memberikan panduan memilih dan merancang metode augmentasi yang sesuai.

No.	Judul Penelitian	Peneliti	Dataset	Objek Penelitian	Hasil Penelitian
3	<i>Image augmentation with geometric transformations</i>	(Krizhevsky <i>et al.</i> , 2012)	ImageNet	Klasifikasi citra menggunakan CNN	Teknik augmentasi seperti <i>translate, flip, dan intensity changing</i> digunakan untuk meningkatkan performa AlexNet pada dataset ImageNet.
4	<i>Image augmentation with geometric transformations</i>	(Krizhevsky <i>et al.</i> , 2012)	ImageNet	Klasifikasi citra menggunakan CNN	Teknik augmentasi seperti <i>translate, flip, dan intensity changing</i> digunakan untuk meningkatkan performa AlexNet pada dataset ImageNet.
5	<i>AutoAugment: Learning augmentation policies from data</i>	(Cubuk <i>et al.</i> , 2019)	CIFAR-10, ImageNet	<i>Optimizing policy-based</i> augmentasi citra	<i>AutoAugment</i> menggunakan <i>reinforcement learning</i> untuk menemukan kebijakan augmentasi optimal, meningkatkan akurasi model pada berbagai dataset.
6	<i>Mixup: Beyond Empirical Risk Minimization</i>	(Zhang <i>et al.</i> , 2018)	CIFAR-10, ImageNet	Klasifikasi citra	<i>Teknik Mixup</i> menggabungkan dua citra dan labelnya untuk membuat citra baru, meningkatkan akurasi dan generalisasi model pada berbagai dataset.
7	<i>Generative adversarial networks for data augmentation in deep learning</i>	Tanaka & Aranha (2019)	Beragam dataset citra	Augmentasi berbasis model	Menggunakan GAN untuk menghasilkan citra baru yang mirip dengan data asli, mengatasi masalah ketidakseimbangan kelas dan meningkatkan performa model.
8	<i>Cutout: Simple yet effective regularization method for convolutional neural networks</i>	(DeVries & Taylor, 2017)	CIFAR-10, ImageNet	Klasifikasi citra	<i>Cutout</i> secara acak menghapus bagian dari citra selama pelatihan untuk memaksa model belajar fitur yang lebih umum dan menghindari <i>overfitting</i> .
9	<i>GridMask Data Augmentation</i>	(Chen <i>et al.</i> , 2024)	CIFAR-10, ImageNet	Klasifikasi citra	<i>GridMask</i> menerapkan masker grid multiskala ke citra untuk meniru oklusi, meningkatkan model yang tangguh terhadap variasi data.
10	<i>Style-consistent image translation for data augmentation in plant disease recognition</i>	(M. Xu <i>et al.</i> , 2022)	Plant disease images	Pengenalan penyakit tanaman	Teknik translasi citra yang konsisten dengan gaya untuk augmentasi data, meningkatkan akurasi pengenalan penyakit tanaman.
11	<i>PuzzleMix: Exploiting Saliency and Local Statistics for Optimal Mixup</i>	(Kim <i>et al.</i> , 2020)	CIFAR-10, ImageNet	Klasifikasi citra	<i>PuzzleMix</i> mengoptimalkan masker untuk menggabungkan dua citra menggunakan informasi

No.	Judul Penelitian	Peneliti	Dataset	Objek Penelitian	Hasil Penelitian
12	<i>YOLOv4: Optimal speed and accuracy of object detection</i>	(Bochkovskiy et al., 2020)	COCO	Deteksi objek	<i>saliency</i> dan <i>Local Statistics</i> , meningkatkan akurasi model pada dataset beragam. YOLOv4 menggunakan berbagai teknik augmentasi seperti <i>Mosaic</i> , <i>Scale</i> , dan <i>Color space</i> untuk meningkatkan akurasi dan kecepatan deteksi objek.
13	<i>RandAugment: Practical automated data augmentation with a reduced search space</i>	(Cubuk et al., 2020)	CIFAR-10, ImageNet	Optimizing policy-based augmentasi citra	<i>RandAugment</i> mengurangi ruang pencarian <i>AutoAugment</i> dengan menyesuaikan probabilitas, membuat augmentasi otomatis lebih praktis dan efisien.
14	<i>CutMix: Regularization strategy to train strong classifiers with localizable features</i>	(Yun et al., 2019)	CIFAR-10, ImageNet	Klasifikasi citra	<i>CutMix</i> menggabungkan dua citra secara spasial dan menggabungkan labelnya, membantu model belajar fitur lokal yang dapat dilokalisasi dan meningkatkan akurasi.

Augmentasi citra telah menjadi teknik penting dalam meningkatkan performa model *deep learning*, terutama ketika dataset yang tersedia terbatas atau tidak seimbang. Penelitian mengenai *LayerMix* memperkenalkan teknik augmentasi data yang dirancang khusus untuk meningkatkan ketahanan (*robustness*) model *deep learning* terhadap gangguan seperti korupsi gambar dan pergeseran domain. Teknik ini terbukti secara konsisten melampaui metode state-of-the-art lainnya dengan menurunkan mean corruption error (mCE) sebesar 14,26% pada ImageNet-200 dan mengurangi mCE hingga 19,7% pada CIFAR-100, sembari tetap memberikan peningkatan akurasi pada data bersih sebesar 0,5% (Ahmad et al., 2026). Hal ini memperkuat peran augmentasi citra sebagai teknik krusial dalam meningkatkan performa model, terutama saat menghadapi dataset yang terbatas atau tidak seimbang. Hasil penelitian berupa survey komprehensif oleh (M. Xu et al., 2023), teknik augmentasi citra diklasifikasikan ke dalam tiga kategori utama: *model-free*, *model-based*, dan *optimizing policy-based*. Teknik *model-free* mencakup transformasi geometris, pemrosesan citra warna, dan transformasi intensitas. Contohnya adalah teknik *flip*, *crop*, dan *translate* yang digunakan oleh

(Krizhevsky *et al.*, 2012) dalam arsitektur AlexNet, serta teknik *crop* dan *flip* yang digunakan oleh (He *et al.*, 2016) dalam arsitektur ResNet.

Teknik *model-based* menggunakan model generatif untuk menghasilkan citra baru. Contoh signifikan dari teknik ini adalah penggunaan *Generative Adversarial Networks* (GANs) untuk menyeimbangkan kelas yang tidak seimbang, seperti yang dilakukan oleh Mariani *et al.*, (2018) dengan ImbCGAN, yang menggunakan GAN untuk menghasilkan citra dari kelas minoritas. Teknik lain yang serupa adalah DAGAN yang dikembangkan oleh (antoniou *et al.*, 2018), di mana citra digunakan sebagai kondisi untuk generator dan *discriminator*, memungkinkan pembuatan citra baru berdasarkan label yang diberikan.

Penelitian oleh DeVries *et al.*, (2017) memperkenalkan teknik *Cutout*, di mana sebagian area dari citra ditutupi dengan masker berukuran tetap untuk setiap citra. Teknik ini dirancang untuk meniru *occlusion* dan meningkatkan kemampuan generalisasi model dengan memaksa model belajar dari bagian citra yang tidak lengkap. Teknik lain seperti *Random Erasing* juga mengubah area persegi panjang acak dari citra dengan nilai piksel acak untuk tujuan yang sama.

Penelitian oleh Yun *et al.*, (2019) memperkenalkan *CutMix*, sebuah teknik yang menggabungkan dua citra secara spasial dan menggabungkan labelnya secara linier. Teknik ini menunjukkan peningkatan kinerja signifikan dalam klasifikasi citra dengan memberikan lebih banyak variasi dan kompleksitas pada data pelatihan. Penelitian tentang teknik *AugMix* yang dikembangkan oleh Hendrycks *et al.*, (2020) menggabungkan beberapa augmentasi dasar dan hasilnya digabungkan dengan citra asli untuk meningkatkan ketahanan dan ketidakpastian model terhadap perubahan data.

Implementasi teknik augmentasi ini telah terbukti efektif dalam berbagai aplikasi, termasuk klasifikasi citra dan deteksi objek, contohnya, teknik *Mixup* yang dikembangkan oleh Zhang *et al.*, (2017) menggabungkan dua citra dan labelnya secara linier, menghasilkan peningkatan akurasi validasi di ImageNet-2012 sebesar 1,5% pada ResNet-50 dan 1,2% pada ResNet-101. Teknik ini juga menunjukkan bahwa citra yang dihasilkan lebih masuk akal bagi mesin dibandingkan manusia, menekankan pentingnya augmentasi dalam ruang distribusi sekitar citra asli.

Teknik augmentasi seperti *GridMask* yang dikembangkan oleh Chen *et al.*, (2020) menggunakan masker *grid* multiskala untuk meniru *occlusion* nyata, menghasilkan keseimbangan yang lebih baik antara penghapusan dan pelestarian area penting dari objek dalam citra. Teknik ini, bersama dengan lainnya, menunjukkan bagaimana augmentasi citra dapat memperbaiki masalah umum seperti variasi citra, ketidakseimbangan kelas, dan domain *shift* dalam tugas visi komputer. Penelitian dan pengembangan teknik augmentasi citra yang terus berlanjut memainkan peran penting dalam meningkatkan kinerja model *deep learning*, memberikan alat yang lebih kuat dan adaptif untuk berbagai aplikasi praktis di dunia nyata.

METODOLOGI PENELITIAN

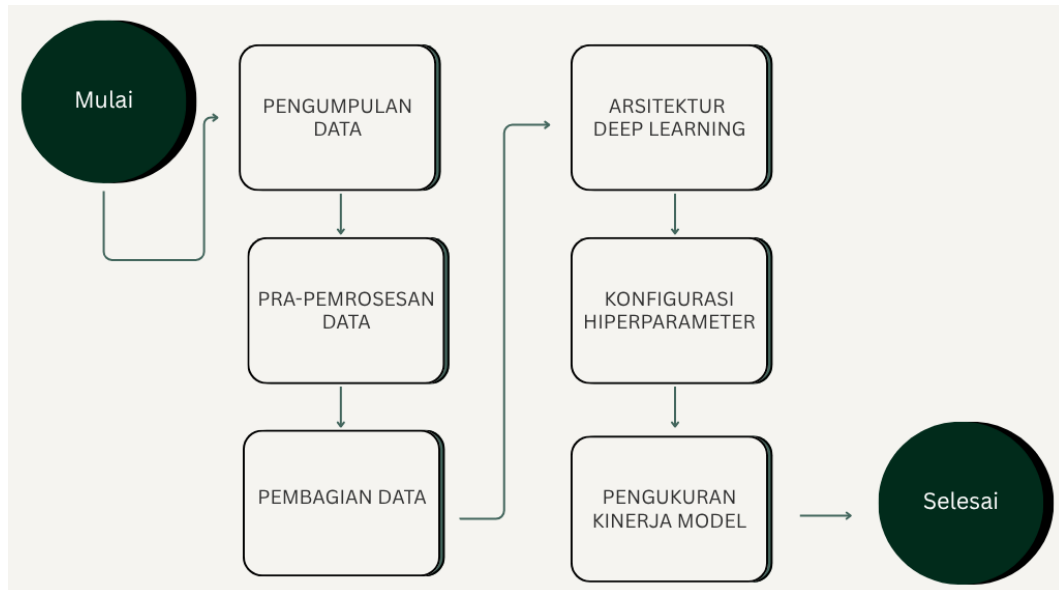
Bab III ini membahas secara rinci pendekatan, tahapan, dan strategi yang digunakan dalam pelaksanaan penelitian ini. Metodologi penelitian digunakan pada dua penelitian, pertama penelitian tentang Identifikasi kupu-kupu Sumatera menggunakan *deep learning* dan augmentasi data pada *deep learning* untuk identifikasi kupu-kupu Sumatera.

3.1 Identifikasi kupu-kupu Sumatera menggunakan *deep learning*

Metodologi penelitian ini mencakup enam tahapan utama yang secara sistematis memandu proses pengembangan, pelatihan, serta evaluasi berbagai arsitektur *deep learning* untuk identifikasi spesies kupu-kupu Sumatera. Tahapan penelitian yang disajikan pada Gambar 5 menjadi landasan bagi rangkaian langkah tersebut, yang meliputi pengumpulan data, pra-pemrosesan data, pembagian data, arsitektur *deep learning*, konfigurasi *hyperparameter*, dan pengukuran kinerja model. Penjelasan mendalam mengenai setiap tahapan tersedia pada subbagian berikut guna menjamin aspek transparansi, reproduksibilitas, serta ketajaman metodologis.

3.1.1 Pengumpulan Data

Proses pengumpulan data melibatkan 800 citra JPEG beresolusi tinggi yang diambil menggunakan kamera digital di bawah pencahayaan alami di Taman Kupu-Kupu Gita Persada. Pendokumentasian delapan spesies Sumatera tersebut dilakukan dari berbagai sudut pandang guna menangkap variasi bentuk sayap, postur, serta orientasi secara alami. Strategi ini memungkinkan model mencapai



Gambar 5. Tahapan Penelitian Identifikasi Kupu-kupu Sumatera dengan *deep learning*.

kemampuan generalisasi yang tangguh tanpa bergantung pada manipulasi buatan. Protokol kontrol kualitas diterapkan dengan mengecualikan citra yang memiliki distorsi parah, terutama keaburan akibat gerakan (*motion blur*) atau pencahayaan berlebih (*overexposure*). Citra dengan oklusi parsial dan latar belakang yang kompleks tetap dipertahankan guna mensimulasikan tantangan nyata di lapangan. Pakar dari taman kupu-kupu melakukan validasi identitas setiap spesies untuk memastikan akurasi taksonomi serta meminimalkan kesalahan pelabelan (*label noise*).

3.1.2 Pra-pemrosesan Data

Tahap pra-pemrosesan difokuskan pada persiapan dataset citra untuk pelatihan model dengan menggunakan alur kerja yang konsisten dan dapat direproduksi. Dataset terlebih dahulu dibersihkan dengan menghapus citra duplikat, rusak, dan berkualitas rendah untuk memastikan konsistensi data. Seluruh citra diubah ukurannya menjadi 224×224 piksel menggunakan interpolasi nearest neighbor agar sesuai dengan kebutuhan masukan model yang dievaluasi. Nilai piksel diubah

skalanya dari rentang 0–255 menjadi 0–1 dan dinormalisasi menggunakan nilai mean dan standar deviasi ImageNet. Seluruh citra dipertahankan dalam format RGB tiga kanal. Dataset dibagi menjadi data pelatihan (60%), validasi (20%), dan pengujian (20%) menggunakan ImageDataGenerator dengan prapemrosesan ImageNet. One-hot encoding diterapkan untuk merepresentasikan label kelas, sedangkan data pelatihan diacak untuk meningkatkan kemampuan generalisasi model. Penelitian ini tidak menerapkan penghapusan latar belakang maupun pemotongan citra sehingga model dapat mempelajari fitur dari konteks citra asli. Alur prapemrosesan mencakup pemuatan citra, pengubahan ukuran, normalisasi, pengodean label, dan pembangkitan batch untuk pelatihan serta evaluasi. Alur kerja prapemrosesan yang terstandarisasi ini memastikan konsistensi dan reproduktibilitas pada seluruh eksperimen.

3.1.3 Pembagian Data

Tahap ketiga meliputi pembagian kumpulan data yang telah diproses sebelumnya ke dalam tiga subset: 60% data latih, 20% validasi, dan 20% pengujian. Pembagian ini dirancang untuk menyediakan data pelatihan yang memadai bagi optimasi model, set validasi untuk penyesuaian *hyperparameter* dan *early stopping*, serta set pengujian independen untuk penilaian performa yang objektif. Keseimbangan kelas tetap dipertahankan selama proses ini guna mencegah terjadinya kebocoran data antar subset.

Proses penghapusan data duplikat dilakukan sebelum partisi dengan menghasilkan *hash* MD5 untuk setiap citra guna memastikan tidak ada citra identik yang muncul di berbagai subset. Pendekatan berbasis *hash* ini secara spesifik menghapus salinan citra yang sama persis agar tidak terdistribusi ke dalam beberapa subset. Penggunaan MD5 memiliki keterbatasan karena hanya mampu mengidentifikasi file yang identik secara bit dan tidak dapat mendeteksi citra yang hampir identik dari spesimen yang sama akibat variasi sudut, pencahayaan, atau latar belakang (Morra dan Lamberti, 2019). Evaluasi komparatif terbaru menunjukkan bahwa metode *hashing* sangat efisien secara komputasi untuk pencocokan persis, namun kinerjanya kurang optimal pada duplikat yang mendekati identik atau di bawah transformasi geometris (Mahmud *et al.*, 2026). Strategi pemisahan tingkat

spesimen diimplementasikan untuk mengatasi batasan tersebut dan mencegah potensi kebocoran identitas. Seluruh citra yang diambil dari individu kupu-kupu yang sama dialokasikan secara eksklusif ke dalam satu subset saja: baik itu pelatihan, validasi, atau pengujian (Wahlteiz & Wahlteiz, 2024). Pendekatan gabungan ini menjamin penghapusan duplikat persis sekaligus mencegah kontaminasi silang antar subset data oleh citra spesimen yang hampir identik guna mendukung klaim generalisasi yang lebih andal.

Strategi partisi bertingkat (*stratified*) per kelas diterapkan untuk menjaga representasi proporsional dari kedelapan spesies kupu-kupu di seluruh subset guna memitigasi bias terhadap kelas dominan. Citra diacak di dalam setiap kelas sebelum dialokasikan dengan rasio 60:20:20 ke dalam set pelatihan, validasi, dan pengujian untuk memastikan keseimbangan kelas. Kumpulan data pelatihan mencakup 480 citra, set validasi 160 citra, dan set pengujian 160 citra dari total 800 citra yang tersedia. Rincian pembagian kumpulan data tersebut disajikan secara lengkap dalam Tabel 4. Ukuran sampel total sebanyak 800 citra diakui secara eksplisit sebagai sebuah batasan metodologis. Volume data yang terbatas dalam konteks pembelajaran mendalam (*deep learning*) dapat menghambat kemampuan model dalam mempelajari fitur berdimensi tinggi serta meningkatkan risiko *overfitting*. Augmentasi data sengaja tidak diterapkan demi menjaga standar integritas data yang tinggi meskipun volume sampel terbatas. Langkah ini diambil agar performa model mencerminkan kemampuannya dalam mengenali variasi morfologi dunia nyata, bukan sekadar artefak sintesis, sehingga inflasi metrik dapat dihindari.

Tabel 4. Pembagian dataset untuk pelatihan, validasi, dan pengujian

Subset	Prosentase	Jumlah citra	Per Kelas
Pelatihan	60%	480	60
Validasi	20%	160	20
Pengujian	20%	160	20

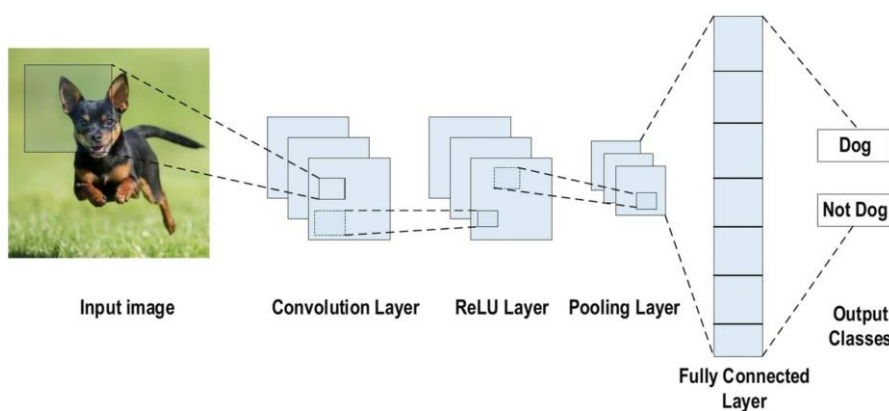
Model dievaluasi berdasarkan kapasitasnya untuk melakukan generalisasi pada spesimen yang benar-benar baru daripada sekadar menghafal fitur spesifik

individu. Kebocoran data dicegah secara ketat melalui penghapusan duplikat sebelum partisi dan alokasi bertingkat. Keputusan strategis untuk tidak menerapkan augmentasi data bertujuan memprioritaskan keaslian data dibandingkan volume sintetis. Penggunaan citra asli secara eksklusif memastikan akurasi yang dilaporkan mencerminkan kemampuan model menangani variasi morfologi alami, seperti postur sayap dan pencahayaan, tanpa gangguan atau bias dari transformasi geometris. Set pelatihan menyediakan data untuk penyempurnaan bobot berbasis *backpropagation* yang penting bagi identifikasi spesies. Set validasi berfungsi memantau dinamika pelatihan dan mendukung mekanisme kontrol seperti *early stopping*. Evaluasi akhir studi ini mengandalkan set pengujian independen sehingga tidak menggunakan *k-fold cross validation*.

3.1.4 Arsitektur *Deep Learning*

Tahap keempat melibatkan pemilihan tujuh arsitektur pembelajaran mendalam mutakhir guna mengevaluasi kinerjanya dalam identifikasi spesies kupu-kupu: DenseNet201, InceptionV3, MobileNet, ResNet50, Xception, EfficientNetB0, dan ViT. Arsitektur-arsitektur tersebut merepresentasikan beragam strategi desain yang memungkinkan perbandingan komprehensif antara model berbasis konvolusi dan model berbasis transformer.

Convolutional Neural Network memproses data visual melalui lapisan-lapisan yang mempelajari fitur hierarkis secara otomatis diperlihatkan pada gambar 6. Citra masukan, khususnya foto kupu-kupu, menyalurkan data piksel mentah ke



Gambar 6. Contoh arsitektur CNN untuk klasifikasi citra (Alzubaidi *et al.*, 2021).

lapisan konvolusi untuk mendeteksi pola dasar seperti tepi, tekstur, dan bentuk. Lapisan ReLU menambahkan sifat non-linearitas guna menangkap hubungan fitur yang kompleks. Lapisan *pooling* berfungsi mengurangi dimensi peta fitur, merangkum informasi utama, meningkatkan ketahanan terhadap variasi, serta meminimalkan beban komputasi. Fitur-fitur tingkat tinggi tersebut kemudian diratakan (*flattened*) dan diteruskan ke lapisan *fully connected* untuk proses klasifikasi. Lapisan keluaran menghasilkan distribusi probabilitas di seluruh kelas yang mencerminkan tingkat kepercayaan prediksi model. Arsitektur ini secara umum menunjukkan keunggulan yang signifikan dalam tugas-tugas pengenalan citra (Alzubaidi *et al.*, 2021).

DenseNet201 mendorong penggunaan kembali fitur melalui koneksi padat yang memfasilitasi aliran informasi dan gradien untuk menghasilkan model yang lebih ringkas serta meningkatkan performa (Yasmin *et al.*, 2023). InceptionV3 memanfaatkan konvolusi terfaktorisasi dan pengklasifikasi tambahan guna meningkatkan efisiensi komputasi serta memitigasi *overfitting*, sehingga sangat cocok untuk berbagai tugas pengenalan spesies (Spiesman *et al.*, 2021). MobileNet dirancang khusus untuk aplikasi visi pada perangkat seluler dengan memanfaatkan konvolusi *depthwise separable* guna memangkas biaya komputasi tanpa mengorbankan akurasi secara signifikan (Adityawan *et al.*, 2023). ResNet50 mengatasi masalah *vanishing gradient* pada jaringan saraf yang dalam melalui koneksi residual yang memungkinkan pelatihan model dengan tingkat kedalaman dan akurasi yang lebih tinggi (Karim *et al.*, 2024). Xception dicirikan oleh penggunaan konvolusi *depthwise separable* yang telah terbukti memberikan kinerja kuat dengan akurasi validasi yang unggul dibandingkan model serupa lainnya (Hasan *et al.*, 2024). EfficientNetB0 dikenal melalui metode penskalaan gabungan yang menggunakan lebih sedikit parameter untuk mencapai performa luar biasa dengan menskalakan kedalaman, lebar, dan resolusi jaringan secara konsisten. ViT sebagai arsitektur yang lebih modern memproses citra dengan membaginya ke dalam *patch* berukuran tetap, memperlakukannya sebagai urutan, dan menggunakan mekanisme *self-attention* untuk menangkap dependensi jarak jauh. Karakteristik ini membedakannya dari operasi lokal pada CNN yang cenderung fokus pada fitur-fitur spasial terdekat (Pucci *et al.*, 2025).

3.1.5 Konfigurasi *Hyperparameter*

Strategi pembekuan (*freezing*) yang terarah diterapkan secara khusus pada arsitektur *convolutional neural network* seperti InceptionV3, Xception, dan DenseNet201. Seluruh basis konvolusi pra-latih yang mengandung bobot teroptimasi dari kumpulan data ImageNet dipertahankan dalam kondisi beku selama fase pelatihan utama. Pemilihan pendekatan ini bertujuan memanfaatkan kemampuan ekstraksi fitur yang telah mapan guna memastikan fitur spasial tingkat rendah, seperti margin sayap yang rumit dan tekstur sisik yang krusial bagi identifikasi kupu-kupu, tetap terjaga. Komponen yang dapat dilatih hanya terbatas pada bagian kepala klasifikasi baru yang mengintegrasikan lapisan *Global Average Pooling* serta lapisan *output dense*. Pembatasan pembaruan bobot pada lapisan tingkat atas tersebut memungkinkan model beradaptasi dengan tugas klasifikasi delapan spesies tanpa merusak pengetahuan dasar yang diperoleh selama pra-pelatihan.

Vision Transformer memerlukan konfigurasi yang lebih konservatif mengingat sensitivitas inheren arsitektur tersebut terhadap ukuran kumpulan data yang terbatas. Lapisan *patch embedding* dan blok transformer awal dibekukan secara ketat demi menjaga stabilitas struktural serta mencegah fenomena *catastrophic forgetting* pada dataset berjumlah 800 citra. Fokus pelatihan dialihkan sepenuhnya pada blok transformer akhir dan bagian *Multi-Layer Perceptron*. Strategi pelepasan beku (*unfreezing*) secara selektif ini memungkinkan model menyempurnakan mekanisme *self-attention* untuk mengenali nuansa spesifik morfologi serangga sekaligus menekan risiko *overfitting* yang tinggi pada lapisan dalam arsitektur transformer.

Tahap kelima dari penelitian ini melibatkan *hyperparameter tuning* secara intensif untuk menyempurnakan dinamika pelatihan di seluruh arsitektur yang dievaluasi. Upaya tersebut memperluas optimasi awal dengan teknik yang memperkuat stabilitas konvergensi guna menetapkan protokol pelatihan seragam bagi penilaian yang metodologis. Rincian *hyperparameter* yang digunakan secara presisi telah dicantumkan dalam Tabel 5 untuk menjamin transparansi serta replikabilitas studi. Pengoptimal AdamW dipilih untuk meningkatkan kemampuan

generalisasi dengan memisahkan peluruhan bobot (*weight decay*) dari pembaruan gradien. Penerapan metode ini menjaga efektivitas regularisasi, mencegah *overfitting* pada data pelatihan, serta memfasilitasi konvergensi yang lebih stabil. Laju pembelajaran dikalibrasi secara spesifik untuk setiap arsitektur model berdasarkan hasil empiris dari uji coba eksperimental mandiri, bukan sekadar mengadopsi dari literatur terdahulu.

Tabel 5. Konfigurasi *hyperparameter* pelatihan yang digunakan

<i>Hyperparameter</i>	Konfigurasi
<i>Optimizer</i>	AdamW
<i>Learning Rate</i>	0,00001 (mayoritas) / 0,00003 (DenseNet201) / 0,0001 (ViT)
<i>LR Scheduler</i>	ReduceLROnPlateau (<i>patience</i> 3)
<i>Early Stopping</i>	Yes (<i>patience</i> 5)
<i>Batch Size</i>	32
<i>Maximum Epochs</i>	20
<i>Loss Function</i>	<i>Categorical Cross-Entropy</i>
<i>Dropout Rate</i>	0,3 (mayoritas) / 0,5 (InceptionV3 dan Xception) / 0,2 (ViT)
<i>Random Seed</i>	42
<i>Framework</i>	<i>TensorFlow / PyTorch</i> (khusus pada ViT)
<i>Hardware</i>	NVIDIA Tesla T4 GPU

Model-model utama menggunakan nilai dasar 0,00001, sementara DenseNet201 menggunakan 0,00003 dan ViT menggunakan 0,0001 setelah melalui rangkaian pengujian iteratif. Karakteristik hierarki fitur DenseNet201 yang kompleks serta operasi *self-attention* pada ViT mendasari penyesuaian nilai tersebut agar ekstraksi fitur spesifik kupu-kupu berjalan efektif. Efisiensi pelatihan terhadap kompleksitas data diatur melalui dua protokol adaptif. Penjadwal *ReduceLROnPlateau* akan menurunkan laju pembelajaran jika kerugian validasi mengalami stagnasi selama tiga *epoch*. Mekanisme *early stopping* dengan toleransi lima *epoch* akan menghentikan proses pelatihan saat performa validasi mencapai titik maksimal guna mencegah penurunan kemampuan generalisasi. Masalah kelebihan parameterisasi pada kepala klasifikasi dimitigasi dengan penerapan *dropout*. Rasio sebesar 0,3 digunakan secara standar pada MobileNet, ResNet50, dan EfficientNet-B0; nilai 0,5 diterapkan untuk menangani peta fitur yang luas pada InceptionV3 dan Xception; serta nilai 0,2 digunakan pada ViT guna menjaga ketepatan atensi.

Prinsip replikabilitas menjadi fondasi bagi validitas metodologis dalam studi ini. Keragaman spesies kupu-kupu di seluruh subset dijamin melalui pembagian data pelatihan, validasi, dan pengujian dengan rasio 60:20:20 secara bertingkat. Faktor ketidakteraturan stokastik dieliminasi dengan menetapkan benih acak (*random seed*) bernilai 42 untuk memastikan konsistensi hasil. Eksperimen yang dijalankan pada Google Colab memanfaatkan GPU NVIDIA Tesla T4 dan CPU Intel Xeon 2,20 GHz guna mendukung evaluasi lintas *framework* yang adil antara TensorFlow dan PyTorch. Protokol pra-pemrosesan data serta *hyperparameter* inti distandarisasi secara menyeluruh untuk menjamin perbandingan yang akurat antara kedua lingkungan tersebut. Logika optimasi yang identik, termasuk penggunaan AdamW, fungsi kerugian *categorical cross-entropy*, dan ukuran *batch* sinkron sebesar 32, diterapkan secara konsisten meski menggunakan pustaka yang berbeda. Prosedur normalisasi data dan perubahan ukuran citra dilakukan secara seragam pada kedua *framework* agar sinyal masukan tetap konsisten. Penggunaan konfigurasi seragam serta perangkat keras yang sama memastikan bahwa variasi performa yang diamati merupakan representasi murni dari karakteristik arsitektural model, bukan disebabkan oleh ketidaksesuaian tingkat *framework* atau gangguan lingkungan.

3.1.6 Pengukuran Kinerja Model

Fase penilaian akhir mengevaluasi model yang telah dilatih menggunakan empat pengukuran evaluasi kinerja : Akurasi, Presisi, *Recall*, dan *F1-Score* (Ali *et al.*, 2022). Sifat multikelas dari tugas klasifikasi spesies kupu-kupu ini mendasari perhitungan metrik menggunakan rata-rata makro (*macro-averaging*) dan rata-rata mikro (*micro-averaging*). Rata-rata makro menghitung performa secara merata di semua kelas, sementara rata-rata mikro memberikan bobot pada performa berdasarkan jumlah sampel tiap kelas untuk menangani potensi ketidakseimbangan data. Matriks konfusi dihasilkan untuk setiap model guna memvisualisasikan kinerja per kelas serta menyoroti kesalahan klasifikasi antara spesies yang memiliki kemiripan morfologi. Analisis mendalam ini mengungkapkan kekuatan dan kelemahan model, seperti *recall* yang lebih tinggi pada spesies dominan dibandingkan tantangan presisi pada spesies langka,

sehingga memastikan metrik yang digunakan mampu menangkap nuansa ekologis di luar skor agregat. Persamaan

$$\begin{aligned}
P_{\{macro\}} &= \frac{1}{C} \sum_{\{i=1\}}^C \frac{TP_i}{TP_i + FP_i} & R_{\{macro\}} &= \frac{1}{C} \sum_{\{i=1\}}^C \frac{TP_i}{TP_i + FN_i} & F1_{\{macro\}} &= \frac{1}{C} \sum_{\{i=1\}}^C \frac{2TP_i}{2TP_i + FP_i + FN_i} \\
&(a) & &(b) & &(c) \\
P_{\{micro\}} &= \frac{\sum_{i=1}^C TP_i}{\sum_{i=1}^C (TP_i + FP_i)} & R_{\{micro\}} &= \frac{\sum_{i=1}^C TP_i}{\sum_{i=1}^C (TP_i + FN_i)} & F1_{\{micro\}} &= \frac{2 \sum_{i=1}^C TP_i}{2 \sum_{i=1}^C TP_i + \sum_{i=1}^C FP_i + \sum_{i=1}^C FN_i} \\
&(d) & &(e) & &(f) \\
\bar{x} &= \frac{1}{n} \sum_{i=1}^n x_i & & & & s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \\
&(g) & & & & (h)
\end{aligned}$$

Gambar 7. Metrik Evaluasi : (a) *Macro-Precision*, (b) *Macro-Recall*, (c) *Macro-F1-Score*, (d) *Micro-Precision*, (e) *Micro-Recall*, (f) *Micro-F1-Score*, (g) *Mean*, (h) *Standard deviation*.

untuk menghitung metrik-metrik tersebut disajikan dalam Gambar 7 dengan label (a) hingga (h).

Seluruh hasil kinerja dilaporkan sebagai rata-rata aritmatika (\bar{x}) dan standar deviasi sampel (s) dari beberapa kali pengujian eksperimental demi menjaga integritas ilmiah dan memastikan replikabilitas penelitian (Ohamouddou *et al.*, 2025). Definisi matematis untuk metrik-metrik tersebut, (k) dan (l) disajikan pada Gambar 12. Nilai rata-rata menetapkan kecenderungan sentral dari efektivitas model selama jumlah uji coba tertentu (Bouhaddour *et al.*, 2023), sementara standar deviasi berfungsi sebagai ukuran dispersi data (Sayed *et al.*, 2024). Simbol dalam persamaan tersebut mewakili jumlah uji coba eksperimental, menunjukkan skor kinerja pada pengujian ke- i , dan mewakili rata-rata yang dihitung dari seluruh pengujian.

Metrik komputasi seperti waktu pelatihan, kecepatan inferensi, ukuran model, dan jumlah total parameter melengkapi kerangka evaluasi ini selain metrik klasifikasi standar. Penambahan metrik tersebut memungkinkan penilaian holistik mengenai kesesuaian setiap arsitektur untuk implementasi nyata dalam pemantauan ekologi

kupu-kupu pada perangkat dengan sumber daya terbatas (Pucci *et al.*, 2025). Evaluasi multifaset ini mengungkap kekuatan prediktif sekaligus pertimbangan praktis, seperti keseimbangan antara akurasi tinggi dengan kebutuhan perangkat keras yang layak pada platform seperti Google Colab.

3.2 Augmentasi data pada *deep learning* untuk identifikasi kupu-kupu Sumatera

Penelitian ini menerapkan teknik augmentasi data untuk meningkatkan jumlah dan keragaman citra kupu-kupu Sumatera pada proses pelatihan model *deep learning*. Penelitian ini menggunakan augmentasi geometris dasar berupa *flipping* dan *rotation* untuk menghasilkan variasi orientasi citra kupu-kupu. Teknik *flipping* melakukan pembalikan arah horizontal maupun vertikal pada citra asli, sedangkan teknik *rotation* melakukan perubahan sudut tertentu pada citra kupu-kupu. Penelitian ini juga menggunakan teknik augmentasi tingkat lanjut berupa *CutMix*, *Mixup*, *AugMix*, dan *LayerMix*. Teknik *CutMix* menggabungkan dua citra berbeda melalui proses pemotongan sebagian area citra dan penggantian area tersebut menggunakan bagian dari citra lain. Teknik *Mixup* membentuk citra baru melalui interpolasi dua citra beserta labelnya. Teknik *AugMix* menghasilkan citra baru melalui kombinasi beberapa transformasi augmentasi secara acak dan terkontrol. Teknik *LayerMix* melakukan pencampuran representasi fitur pada beberapa lapisan citra selama proses pembelajaran model.

3.2.1 Pembagian data untuk dataset yang menggunakan teknik augmentasi data

Dataset penelitian terdiri atas delapan kelas yang merupakan spesies kupu-kupu Sumatera, yaitu *Papilio memnon*, *Papilio nephelus*, *Pachliopta aristolochiae*, *Cethosia penthesilea*, *Troides helena*, *Papilio peranthus*, *Graphium doson*, dan *Graphium sarpedon*. Setiap spesies memiliki 100 citra asli (*original image*), karena ada 8 spesies kupu-kupu maka citra data aslinya menjadi 800. Tabel 6 menunjukkan distribusi teknik augmentasi data pada setiap spesies kupu-kupu Sumatera berbasis 6 teknik augmentasi yang digunakan pada penelitian ini. Setiap spesies memperoleh 100 data baru dari tiap teknik augmentasi data sehingga

jumlah data baru untuk 8 spesies kupu-kupu menjadi 800, sehingga jumlah citra asli dan jumlah citra hasil tiap teknik augmentasi data adalah 1600.

Tabel 6. Distribusi teknik augmentasi data pada setiap spesies

Spesies Kupu-Kupu Sumatera	<i>Flipping</i>	<i>Rotation</i>	<i>CutMix</i>	<i>Mixup</i>	<i>AugMix</i>	<i>LayerMix</i>
<i>Papilio memnon</i>	100	100	100	100	100	100
<i>Papilio nephelus</i>	100	100	100	100	100	100
<i>Pachliopta aristolochiae</i>	100	100	100	100	100	100
<i>Cethosia penthesilea</i>	100	100	100	100	100	100
<i>Troides helena</i>	100	100	100	100	100	100
<i>Papilio peranthus</i>	100	100	100	100	100	100
<i>Graphium doson</i>	100	100	100	100	100	100
<i>Graphium sarpedon</i>	100	100	100	100	100	100
Total	800	800	800	800	800	800

Proses augmentasi hanya diterapkan pada data pelatihan (training data) dan tidak diterapkan pada data validasi maupun data pengujian. Pendekatan tersebut bertujuan untuk menjaga objektivitas evaluasi model *deep learning* pada proses validasi dan pengujian. Pembagian dataset hasil augmentasi menggunakan rasio 60% untuk data pelatihan, 20% untuk data validasi, dan 20% untuk data pengujian. Pembagian dataset tersebut menghasilkan distribusi data yang seimbang pada setiap tahap eksperimen sehingga proses pelatihan dan evaluasi model dapat dilakukan secara optimal. Tabel 7 menunjukkan pembagian dataset hasil augmentasi untuk pelatihan, validasi, dan pengujian pada penelitian ini.

Tabel 7. Pembagian data pada dataset hasil augmentasi

Subset	Presentase	Jumlah Citra
Pelatihan	60%	1.280
Validasi	20%	160
Pengujian	20%	160
Total	100%	1.600

Distribusi augmentasi yang seimbang pada setiap spesies bertujuan untuk mengurangi ketimpangan distribusi data (*class imbalance*) selama proses pelatihan model *deep learning*. Pendekatan tersebut membantu model mempelajari karakteristik morfologi kupu-kupu secara lebih merata pada seluruh spesies.

Variasi data hasil augmentasi juga diharapkan membantu model mengenali pola visual kupu-kupu pada berbagai kondisi orientasi, pencahayaan, tekstur, dan kompleksitas visual.

3.2.2 Teknik augmentasi data *flipping* pada dataset kupu-kupu Sumatera

Teknik *flipping* atau *flip* melakukan transformasi geometris melalui pembalikan arah citra secara horizontal dan vertikal terhadap citra asli kupu-kupu. Teknik ini termasuk augmentasi geometris dasar yang umum digunakan pada klasifikasi citra karena transformasi tersebut tetap mempertahankan identitas objek utama. Penelitian sebelumnya menjelaskan bahwa *flipping* membantu model *deep learning* mengenali objek pada berbagai orientasi dan posisi pengamatan.

Penerapan *flipping* pada dataset kupu-kupu Sumatera diharapkan membantu model mempelajari pola warna, bentuk sayap, tekstur, dan struktur morfologi dari arah yang berbeda. Teknik ini juga membantu mengurangi positional bias pada dataset pelatihan sehingga model tidak hanya bergantung pada satu orientasi objek tertentu. Variasi orientasi hasil *flipping* membantu model mengenali kupu-kupu pada kondisi alami ketika posisi objek berubah akibat arah terbang maupun sudut pengambilan gambar. Gambar 8 menunjukkan contoh citra kupu-kupu original sebelum dan sesudah dilakukan augmentasi *flip horizontal* sedangkan gambar 9 menunjukkan contoh citra kupu-kupu original sebelum dan sesudah dilakukan augmentasi *flip vertikal*.



Gambar 8. Teknik augmentasi data *flip* horizontal pada citra original.



Gambar 9. Teknik augmentasi data *flip* vertikal pada citra original.

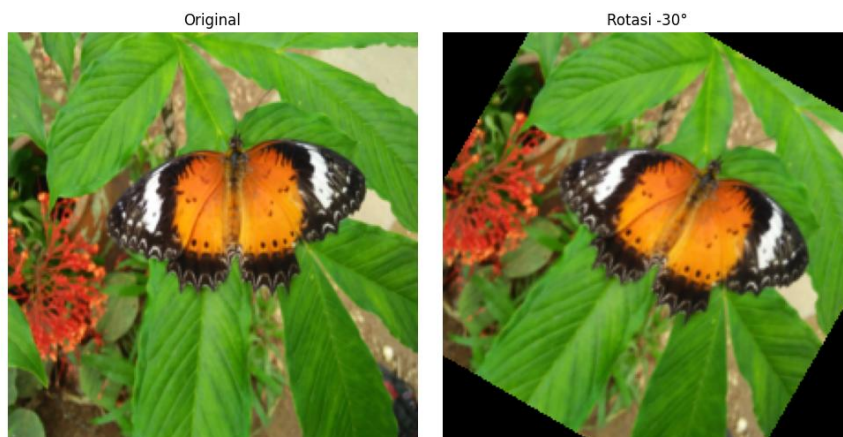
3.2.3 Teknik augmentasi data *rotation* (rotasi) pada dataset kupu-kupu Sumatera

Teknik rotasi melakukan transformasi citra melalui proses rotasi objek pada sudut tertentu terhadap citra asli kupu-kupu. Teknik ini menghasilkan variasi orientasi citra tanpa mengubah karakteristik utama objek. Penelitian tentang augmentasi data menjelaskan bahwa rotasi diharapkan membantu model *deep learning* mempelajari invariansi sudut terhadap objek citra sehingga meningkatkan kemampuan generalisasi model.

Penerapan rotasi pada dataset kupu-kupu Sumatera dilakukan untuk mensimulasikan kondisi nyata pengambilan citra pada berbagai arah kamera dan posisi kupu-kupu. Variasi sudut rotasi membantu model mengenali bentuk sayap, pola venasi, dan tekstur tubuh pada berbagai orientasi. Teknik ini juga membantu model mengurangi bias terhadap posisi objek tertentu pada dataset pelatihan. Gambar 10 dan 11 menunjukkan contoh citra kupu-kupu sebelum dan sesudah dilakukan augmentasi rotasi 30° dan -30° .



Gambar 10. Teknik augmentasi data rotasi 30^0 pada citra original.



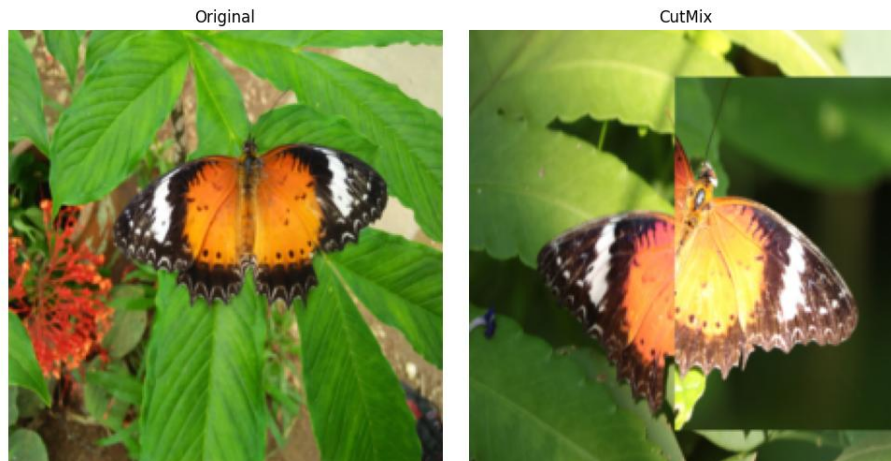
Gambar 11. Teknik augmentasi data rotasi -30^0 pada citra original.

3.2.4 Teknik augmentasi data *CutMix* pada dataset kupu-kupu Sumatera

Teknik *CutMix* melakukan proses pencampuran dua citra melalui pemotongan sebagian area citra pertama dan penggantian area tersebut menggunakan bagian tertentu dari citra kedua. Teknik ini menghasilkan citra sintetis baru yang mengandung kombinasi fitur visual dari dua citra berbeda. Penelitian sebelumnya menjelaskan bahwa teknik pencampuran citra mampu meningkatkan robustness model deep learning terhadap variasi distribusi data visual.

Penerapan *CutMix* pada dataset kupu-kupu Sumatera membantu model mempelajari fitur lokal dan fitur global secara bersamaan. Teknik ini membantu

model mengenali pola warna, bentuk sayap, dan tekstur tubuh antarspesies kupu-kupu yang memiliki kemiripan visual. Teknik *CutMix* juga membantu meningkatkan regularisasi model sehingga model tidak hanya fokus pada satu bagian objek tertentu. Gambar 12 menunjukkan contoh hasil augmentasi *CutMix* pada citra kupu-kupu Sumatera.



Gambar 12. Teknik augmentasi data *CutMix* pada citra original.

3.2.5 Teknik augmentasi data *Mixup* pada dataset kupu-kupu Sumatera

Teknik *Mixup* melakukan interpolasi dua citra beserta label kelasnya untuk menghasilkan citra sintesis baru. Teknik ini membentuk distribusi data yang lebih kontinu pada ruang fitur sehingga model dapat mempelajari batas klasifikasi yang lebih stabil. Penelitian sebelumnya menjelaskan bahwa *Mixup* mampu meningkatkan generalisasi model dan membantu mengurangi overfitting pada proses pelatihan deep learning.

Penerapan *Mixup* pada dataset kupu-kupu Sumatera menghasilkan kombinasi visual antara dua spesies kupu-kupu dengan proporsi tertentu. Teknik tersebut membantu model memahami hubungan antar fitur visual dari berbagai spesies kupu-kupu. Teknik *Mixup* juga membantu model membangun *decision boundary* yang lebih robust terhadap perubahan distribusi data dan noise citra. Gambar 13 menunjukkan contoh hasil augmentasi *Mixup* pada citra kupu-kupu Sumatera.

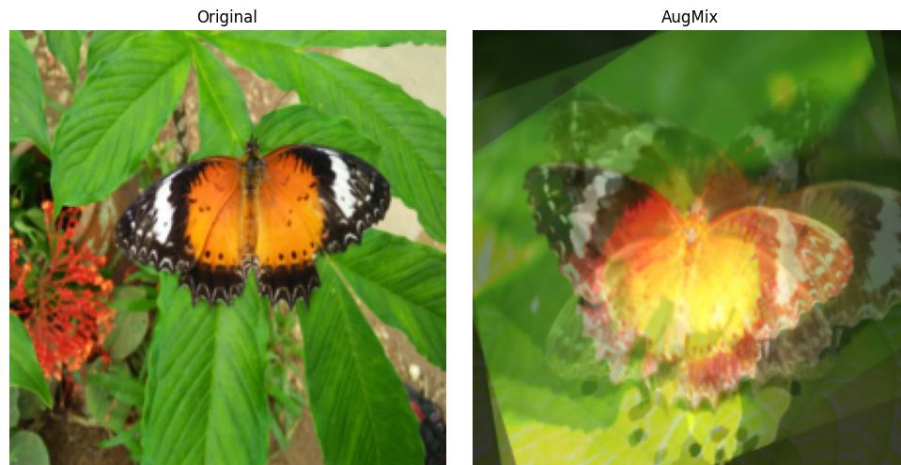


Gambar 13. Teknik augmentasi data *Mixup* pada citra original.

3.2.6 Teknik augmentasi data *AugMix* pada dataset kupu-kupu Sumatera

Teknik *AugMix* melakukan kombinasi beberapa transformasi augmentasi secara acak dan terkontrol pada satu citra untuk menghasilkan variasi visual yang lebih kompleks. Teknik ini menggabungkan berbagai transformasi geometris dan fotometrik seperti rotasi, perubahan kontras, perubahan warna, translasi, dan manipulasi tekstur pada satu citra. Penelitian sebelumnya menjelaskan bahwa *AugMix* meningkatkan robustness model terhadap noise, blur, distorsi visual, dan perubahan distribusi data citra.

Penerapan *AugMix* pada dataset kupu-kupu Sumatera membantu model deep learning mengenali spesies kupu-kupu pada berbagai kondisi lingkungan dan pencahayaan. Teknik ini juga membantu meningkatkan konsistensi prediksi model terhadap perubahan kualitas citra pada data pengujian. Kombinasi beberapa transformasi pada *AugMix* menghasilkan variasi data yang lebih kompleks dibandingkan augmentasi tunggal. Gambar 14 menunjukkan contoh hasil augmentasi *AugMix* pada citra kupu-kupu Sumatera.



Gambar 14. Teknik augmentasi data *AugMix* pada citra original.

3.2.7 Teknik augmentasi data *LayerMix* pada dataset kupu-kupu Sumatera

Teknik *LayerMix* merupakan metode augmentasi modern yang menggunakan pendekatan structured mixing pipeline berbasis kompleksitas visual untuk menghasilkan citra sintesis baru. Penelitian *LayerMix* menjelaskan bahwa metode ini meningkatkan robustness model deep learning terhadap out-of-distribution samples, natural corruption, dan adversarial perturbation.

LayerMix melakukan pencampuran fitur visual secara bertingkat pada beberapa representasi citra sehingga model dapat mempelajari hubungan fitur yang lebih kompleks dibandingkan augmentasi konvensional. Teknik ini menggunakan kombinasi beberapa metode blending seperti arithmetic mean, geometric mean, pixel mixing, dan element mixing untuk menghasilkan variasi citra yang lebih stabil dan terstruktur.

Penerapan *LayerMix* pada dataset kupu-kupu Sumatera membantu model mempelajari pola sayap, tekstur tubuh, venasi sayap, dan distribusi warna secara lebih mendalam. Teknik ini menghasilkan variasi visual dengan kompleksitas tinggi sehingga model deep learning dapat mengenali spesies kupu-kupu pada berbagai kondisi lingkungan nyata.

Gambar 15 menunjukkan contoh hasil augmentasi *LayerMix* pada citra kupu-kupu Sumatera.

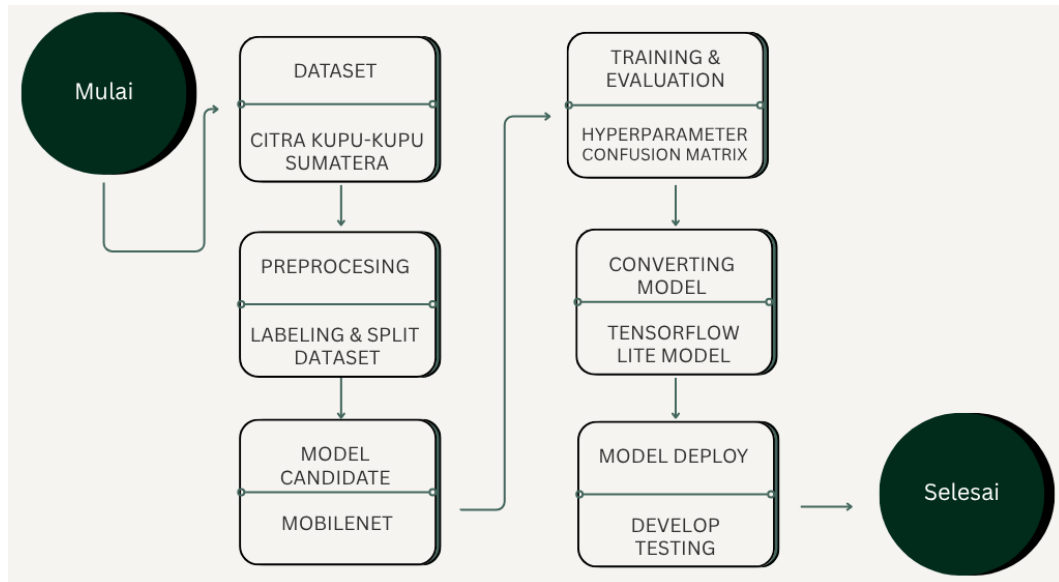


Gambar 15. Teknik augmentasi data *LayerMix* (*arithmetic*, *geometric*, *pixel* dan *element*) pada citra original.

3.3 Implementasi aplikasi mobile untuk identifikasi kupu-kupu

Flowchart penelitian menunjukkan tahapan penelitian identifikasi spesies kupu-kupu Sumatera menggunakan metode deep learning berbasis citra digital (Gambar 16). Proses penelitian dimulai dari pengumpulan dataset hingga implementasi

model pada aplikasi mobile menggunakan *TensorFlow Lite*. Setiap tahapan memiliki fungsi penting dalam menghasilkan sistem identifikasi otomatis yang akurat, efisien, dan dapat digunakan pada perangkat *smartphone*.



Gambar 16. *Flowchart* pengembangan aplikasi *mobile* berbasis *deep learning*.

1. Mulai

Tahapan penelitian dimulai dari proses inialisasi penelitian yang meliputi penentuan tujuan penelitian, pengumpulan kebutuhan sistem, dan penyusunan alur eksperimen identifikasi spesies kupu-kupu Sumatera berbasis deep learning.

2. Dataset Citra Kupu-Kupu Sumatera

Tahapan berikutnya melakukan pengumpulan dataset citra kupu-kupu Sumatera. Dataset penelitian berasal dari citra kupu-kupu lapangan yang diperoleh dari Taman Kupu-Kupu Gita Persada, Bandar Lampung. Dataset terdiri atas delapan spesies kupu-kupu Sumatera dengan variasi posisi, pencahayaan, latar belakang, dan orientasi objek.

Dataset digunakan sebagai sumber data utama pada proses pelatihan dan evaluasi model deep learning. Tahapan ini bertujuan untuk memperoleh data citra yang representatif terhadap kondisi alami kupu-kupu Sumatera.

3. *Preprocessing*, Labeling, dan Split Dataset

Tahapan preprocessing melakukan proses persiapan data sebelum digunakan pada pelatihan model deep learning. Proses *preprocessing* meliputi normalisasi citra, perubahan ukuran gambar (*image resizing*), dan pembersihan data yang tidak sesuai. Tahapan labeling melakukan pemberian label kelas pada setiap citra kupu-kupu sesuai spesies masing-masing. Label tersebut digunakan sebagai target klasifikasi pada proses pelatihan deep learning. Tahapan split dataset membagi dataset menjadi data pelatihan, data validasi, dan data pengujian. Pembagian dataset dilakukan menggunakan rasio tertentu untuk menjaga keseimbangan proses pelatihan dan evaluasi model.

4. *Model Candidate* – MobileNet

Tahapan model candidate menentukan arsitektur deep learning yang digunakan pada penelitian. Flowchart menunjukkan penggunaan MobileNet sebagai kandidat model utama karena memiliki ukuran model yang ringan dan efisien untuk implementasi pada perangkat mobile.

MobileNet dipilih karena mampu menghasilkan performa klasifikasi yang baik dengan kebutuhan komputasi dan penggunaan memori yang lebih rendah dibandingkan beberapa arsitektur deep learning lainnya. Karakteristik tersebut sangat sesuai untuk implementasi sistem identifikasi kupu-kupu pada smartphone.

5. *Training dan Evaluation*

Tahapan training melakukan proses pelatihan model deep learning menggunakan dataset pelatihan yang telah dipersiapkan sebelumnya. Model mempelajari karakteristik visual kupu-kupu seperti pola sayap, warna, tekstur, dan bentuk morfologi untuk membedakan setiap spesies.

Tahapan *evaluation* melakukan evaluasi performa model menggunakan data validasi dan data pengujian. Evaluasi dilakukan untuk mengukur kemampuan generalisasi model terhadap data baru yang tidak digunakan selama proses pelatihan. Tahapan ini juga melibatkan pengaturan hyperparameter seperti *learning rate*, *batch size*, jumlah *epoch*, *optimizer*, dan fungsi aktivasi untuk memperoleh performa model yang optimal.

6. *Hyperparameter dan Confusion Matrix*

Tahapan hyperparameter bertujuan untuk menentukan konfigurasi terbaik pada proses pelatihan model deep learning. Pengaturan hyperparameter mempengaruhi stabilitas pelatihan, kecepatan konvergensi, dan akurasi model.

Tahapan confusion matrix digunakan untuk menganalisis performa klasifikasi model pada setiap kelas spesies kupu-kupu Sumatera. Confusion matrix membantu mengidentifikasi kelas yang memiliki tingkat kesalahan klasifikasi tinggi maupun rendah.

7. *Converting Model – TensorFlow Lite Model*

Tahapan converting model melakukan konversi model deep learning hasil pelatihan ke format *TensorFlow Lite* (.tflite). *TensorFlow Lite* merupakan framework yang dirancang untuk implementasi model *deep learning* pada perangkat *mobile* dan *embedded system*. Proses konversi bertujuan untuk mengurangi ukuran model dan meningkatkan efisiensi inferensi pada smartphone tanpa mengurangi performa klasifikasi secara signifikan.

8. *Model Deploy dan Develop Testing*

Tahapan model deploy melakukan implementasi model *TensorFlow Lite* pada aplikasi mobile identifikasi kupu-kupu Sumatera. Model diintegrasikan ke dalam sistem aplikasi sehingga pengguna dapat melakukan identifikasi spesies kupu-kupu secara otomatis menggunakan kamera smartphone maupun gambar dari galeri. Tahapan *develop testing* melakukan pengujian aplikasi *mobile* untuk memastikan seluruh fitur sistem berjalan dengan baik. Pengujian meliputi proses identifikasi real-time, kompatibilitas perangkat, kecepatan inferensi, penggunaan memori, dan kestabilan aplikasi pada perangkat *mobile*.

9. Selesai

Tahapan selesai menunjukkan bahwa seluruh proses penelitian, pelatihan model, evaluasi performa, konversi model, dan implementasi aplikasi *mobile* telah berhasil dilakukan. Sistem identifikasi spesies kupu-kupu Sumatera berbasis deep learning siap digunakan sebagai alat bantu identifikasi otomatis pada perangkat *smartphone*.

SIMPULAN DAN SARAN

5.1 Simpulan

Penelitian ini berhasil mengevaluasi kinerja berbagai arsitektur *deep learning* dalam identifikasi spesies kupu-kupu Sumatera menggunakan citra digital berbasis augmentasi data. Hasil penelitian menunjukkan bahwa penerapan algoritma *deep learning* mampu mengidentifikasi spesies kupu-kupu Sumatera secara efektif berdasarkan karakteristik visual seperti pola sayap, tekstur, warna, dan bentuk morfologi kupu-kupu pada citra lapangan.

Penelitian ini juga berhasil membandingkan performa beberapa arsitektur *deep learning* berbasis *Convolutional Neural Network* (CNN) dan *Vision Transformer* (ViT) pada proses klasifikasi spesies kupu-kupu Sumatera. Perbandingan tersebut menunjukkan adanya perbedaan kemampuan setiap arsitektur dalam mempelajari representasi fitur visual pada dataset kupu-kupu yang memiliki kompleksitas tinggi dan latar belakang alami yang tidak terkontrol.

Penerapan teknik augmentasi data memberikan pengaruh signifikan terhadap peningkatan performa model *deep learning* pada proses identifikasi spesies kupu-kupu Sumatera. Teknik augmentasi mampu meningkatkan keragaman data pelatihan sehingga model menjadi lebih *robust* terhadap variasi orientasi, pencahayaan, tekstur, dan distribusi visual citra. Hasil evaluasi menunjukkan bahwa kombinasi teknik augmentasi tertentu mampu meningkatkan akurasi klasifikasi dibandingkan penggunaan data asli tanpa augmentasi.

Penelitian ini juga berhasil mengidentifikasi kombinasi terbaik antara arsitektur *deep learning* dan teknik augmentasi data yang menghasilkan performa optimal

berdasarkan aspek akurasi klasifikasi, efisiensi komputasi, dan penggunaan memori. Kombinasi tersebut memberikan kontribusi penting dalam pengembangan sistem identifikasi otomatis spesies kupu-kupu Sumatera berbasis perangkat bergerak (*mobile application*) yang efisien dan adaptif terhadap kondisi lapangan.

5.2 Saran

Pengembangan penelitian selanjutnya perlu diarahkan pada peningkatan generalisasi model melalui penggunaan dataset yang lebih besar dan beragam, baik dari segi lokasi, kondisi lingkungan, maupun variasi spesies. Penerapan teknik augmentasi data yang terkontrol perlu dievaluasi untuk meningkatkan ketahanan model terhadap variasi visual. Validasi eksternal menggunakan data lapangan menjadi langkah penting untuk memastikan kinerja model dalam kondisi nyata.

Evaluasi implementasi pada perangkat bergerak atau sistem berbasis *edge computing* juga diperlukan untuk menilai kelayakan praktis dalam aplikasi monitoring keanekaragaman hayati. Penguatan aspek biologis, termasuk analisis keterkaitan antara kesalahan identifikasi dengan karakter morfologi dan variasi taksonomi, perlu dilakukan untuk meningkatkan kontribusi penelitian terhadap bidang biodiversitas. Pendekatan ini diharapkan dapat mendukung pengembangan sistem identifikasi spesies yang lebih akurat, adaptif, dan aplikatif dalam konteks konservasi dan dokumentasi keanekaragaman hayati.

DAFTAR PUSTAKA

- Adityawan, H. T., Farroq, O., Santosa, S., Islam, H. M. M., Sarker, M. K., & Setiadi, D. R. I. M. (2023). Butterflies Recognition using Enhanced Transfer Learning and Data Augmentation. *Journal of Computing Theories and Applications*, 1(2), 115–128. <https://doi.org/10.33633/jcta.v1i2.9443>
- Afaq, S., & Rao, S. (2020). Significance Of Epochs On Training A Neural Network. *International Journal of Scientific & Technology Research*, 9, 485–488. www.ijstr.org
- Ahmad, H. M., Morle, D., & Rahimi, A. (2026). LayerMix: Enhanced Data Augmentation for Robust Deep Learning. *Pattern Recognition*, 172. <https://doi.org/10.1016/j.patcog.2025.112332>
- Ali, K., Shaikh, Z. A., Khan, A. A., & Laghari, A. A. (2022). Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer. In *Neuroscience Informatics* (Vol. 2, Number 4). Elsevier Masson s.r.l. <https://doi.org/10.1016/j.neuri.2021.100034>
- Almryad, A. S., & Kutucu, H. (2020). Automatic identification for field butterflies by convolutional neural networks. *Engineering Science and*

Technology, an International Journal, 23(1), 189–195.
<https://doi.org/10.1016/j.jestch.2020.01.006>

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), 1–74.
<https://doi.org/10.1186/s40537-021-00444-8>

Andrian, R., Maharani, D., Muhammad, M. A., & Junaidi, A. (2020). Butterfly identification using gray level co-occurrence matrix (GlcM) extraction feature and k-nearest neighbor (knn) classification. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, 6(1), 11–21.
<https://doi.org/10.26594/register.v6i1.1602>

Arzar, N. N., Sabri, N., Mohd Johari, N. F., Amilah Shari, A., Mohd Noordin, M. R., & Ibrahim, S. (2019). Butterfly Species Identification Using Convolutional Neural Network (CNN). *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, 221–224. <https://doi.org/10.1109/I2CACIS.2019.8825031>

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. <http://arxiv.org/abs/2004.10934>

Borza, D., Itu, R., & Danescu, R. (2018). Micro Expression Detection and Recognition from High Speed Cameras using Convolutional Neural Networks. *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 201–208. <https://doi.org/10.5220/0006548802010208>

Bouhaddour, S., Saadi, C., Bouabdallaoui, I., Sbihi, M., & Guerouate, F. (2023). A Novel Hybrid Approach for Daily Tourism Arrival

Forecasting: The PROPHET-Bayesian Gaussian Process-Forward Neural Network Model. *Ingenierie Des Systemes d'Information*, 28(4), 833–842. <https://doi.org/10.18280/isi.280404>

Chen, P., Liu, S., Zhao, H., Wang, X., & Jia, J. (2024). GridMask Data Augmentation. 1–9. <http://arxiv.org/abs/2001.04086>

Cho, J., Lee, K., Shin, E., Choy, G., & Do, S. (n.d.). *Medical Image Deep Learning With Hospital Pacs Dataset*.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). AutoAugment: Learning Augmentation Strategies From Data. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 113–123. <https://doi.org/10.1109/CVPR.2019.00020>

DeVries, T., & Taylor, G. W. (2017). *Improved Regularization of Convolutional Neural Networks with Cutout*. <http://arxiv.org/abs/1708.04552>

Faruk, O., Kaya, Y., Kayci, L., Tekin, R., Faruk Ertuğrul, Ö., Kaya, Y., & Kayci, L. (2015). A Vision System for Classifying Butterfly Species by using Law's Texture Energy Measures. In *Avestia Publishing International Journal on Computer Vision* (Vol. 1). <https://www.researchgate.net/publication/270903382>

Hasan, M. R., Rahman, M. M., Shahriar, F., Khan, M. S. I., Mohi Uddin, K. M., & Hasan, M. M. (2024). Smart farming: Leveraging IoT and deep learning for sustainable tomato cultivation and pest management. *Crop Design*, 3(4). <https://doi.org/10.1016/j.crope.2024.100079>

Hassan, S. N. A., Rahman, N. S. A., Htike, Z. Z., & Win, S. Le. (2014). Advances in Automatic Insect Classification. *Electrical and Electronics*

- Engineering: An International Journal*, 3(2), 51–63.
<https://doi.org/10.14810/elelij.2014.3204>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Identity Mappings in Deep Residual Networks*. <http://arxiv.org/abs/1603.05027>
- Hebert, P. D. N., & Gregory, T. R. (2005). The Promise of DNA Barcoding for Taxonomy. *Systematic Biology*, 54(5), 852–859.
<https://doi.org/10.1080/10635150500354886>
- Ilahiyah, S., & Nilogiri, A. (2018). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *Jurnal Sistem & Teknologi Informasi Indonesia*, 3, 49–56.
- Irfansyah, D., Mustikasari, M., & Suroso, A. (2021). Arsitektur Convolutional Neural Network (CNN) Alexnet Untuk Klasifikasi Hama Pada Citra Daun Tanaman Kopi. *Jurnal Informatika: Jurnal Pengembangan IT*, 6(2), 87–92. <https://doi.org/10.30591/jpit.v6i2.2802>
- Kang, S.-H., Jeon, W., & Lee, S.-H. (2012). Butterfly species identification by branch length similarity entropy. *Journal of Asia-Pacific Entomology*, 15(3), 437–441. <https://doi.org/10.1016/j.aspen.2012.05.005>
- Karim, A. A. J., Mahmud, M. Z., & Khan, R. (2024). Advanced vision transformers and open-set learning for robust mosquito classification: A novel approach to entomological studies. *PLOS Computational Biology*, 20(12), e1012654-. <https://doi.org/10.1371/journal.pcbi.1012654>
- Kartika, D. S. Y., Herumurti, D., & Yuniarti, A. (2017). Butterfly Image Classification Using Color Quantization Method on HSV Color Space

and Local Binary Pattern. In *Postgraduate Program Institut Teknologi Sepuluh Nopember*.

Kartika, D. S. Y., Herumurti, D., & Yuniarti, A. (2018). Local Binary Pattern Method And Feature Shape Extraction For Detecting Butterfly Image. *International Journal of GEOMATE*, 15(50). <https://doi.org/10.21660/2018.50.IJCST21>

Kaya, Y., Kayci, L., & Tekin, R. (2013). A Computer Vision System for the Automatic Identification of Butterfly Species via Gabor-Filter-Based Texture Features and Extreme Learning Machine: GF+ELM. *TEM Journal*, 13–20. <https://doi.org/10.18421/TEM21-02>

Kayci, L., & Kaya, Y. (2014). A vision system for automatic identification of butterfly species using a grey-level co-occurrence matrix and multinomial logistic regression. *Zoology in the Middle East*, 60(1), 57–64. <https://doi.org/10.1080/09397140.2014.892340>

Kim, J.-H., Choo, W., & Song, H. O. (2020). Puzzle Mix: Exploiting Saliency and Local Statistics for Optimal Mixup. *International Conference on Machine Learning*. <http://arxiv.org/abs/2009.06962>

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. <http://code.google.com/p/cuda-convnet/>

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>

Li, F., & Xiong, Y. (2018). Automatic identification of butterfly species based on HoMSC and GLCMoIB. *The Visual Computer*, 34(11), 1525–1533. <https://doi.org/10.1007/s00371-017-1426-1>

- Liang, B., Wu, S., Xu, K., & Hao, J. (2020). *Butterfly Detection and Classification Based on Integrated YOLO Algorithm* (pp. 500–512). https://doi.org/10.1007/978-981-15-3308-2_55
- Lin, Z., Jia, J., Gao, W., & Huang, F. (2020). Fine-grained visual categorization of butterfly specimens at sub-species level via a convolutional neural network with skip-connections. *Neurocomputing*, *384*, 295–313. <https://doi.org/10.1016/j.neucom.2019.11.033>
- Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*, 11–26. <https://doi.org/10.1016/j.neucom.2016.12.038>
- Mahmud, M. F., Nusrat, Z., & Pan, W. D. (2026). Comparative Evaluation of Perceptual Hashing and Deep Embedding Methods for Robust and Efficient Image Deduplication. *Electronics*, *15*(7), 1493. <https://doi.org/10.3390/electronics15071493>
- Maxwell, A. E., Warner, T. A., & Guillén, L. A. (2021). Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—part 1: Literature review. In *Remote Sensing* (Vol. 13, Number 13). MDPI AG. <https://doi.org/10.3390/rs13132450>
- Morra, L., & Lamberti, F. (2019). Benchmarking unsupervised near-duplicate image detection. *Expert Systems with Applications*, *135*, 313–326. <https://doi.org/https://doi.org/10.1016/j.eswa.2019.05.002>
- Nainwal, A., Kumar, Y., & Jha, B. (2022). Arrhythmia classification based on improved monarch butterfly optimization algorithm. *Journal of King*

Saud University - Computer and Information Sciences, 34(8), 5100–5109. <https://doi.org/10.1016/j.jksuci.2022.01.002>

Ohamouddou, M., Ohamouddou, S., El Afia, A., & Lasri, R. (2025). ATMS-KD: Adaptive temperature and mixed sample knowledge distillation for a lightweight residual CNN in agricultural embedded systems. *Smart Agricultural Technology*, 12, 101617. <https://doi.org/https://doi.org/10.1016/j.atech.2025.101617>

Perveen, F. K., & Khan, A. (2017). Introductory Chapter: Lepidoptera. In F. K. Perveen (Ed.), *Lepidoptera* (1). IntechOpen. <https://doi.org/10.5772/intechopen.70452>

Pucci, R., Kalkman, V. J., & Stowell, D. (2025). Performance of Computer Vision Algorithms for Fine-Grained Classification Using Crowdsourced Insect Images. *IET Computer Vision*, 19(1), e70006. <https://doi.org/https://doi.org/10.1049/cvi2.70006>

Rochmawati, N., Hidayati, H. B., Yamasari, Y., Tjahyaningtijas, H. P. A., Yustanti, W., & Prihanto, A. (2021). Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam. *Journal of Information Engineering and Educational Technology*, 5(2), 44–48. <https://doi.org/10.26740/jieet.v5n2.p44-48>

Sayed, G. I., Alshater, H., & Hassanien, A. E. (2024). Predicting the potential toxicity of the metal oxide nanoparticles using machine learning algorithms. *Soft Computing*, 28(17–18), 10235–10261. <https://doi.org/10.1007/s00500-024-09774-0>

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60. <https://doi.org/10.1186/s40537-019-0197-0>

- Soekardi, H., Larasati, A., Djausal, A., & Martinus. (2016). *Kupu-Kupu Lampung Taman Kupu-Kupu Gita Persada*. Yayasan Sahabat Alam.
- Spiesman, B. J., Gratton, C., Hatfield, R. G., Hsu, W. H., Jepsen, S., McCornack, B., Patel, K., & Wang, G. (2021). Assessing the potential for deep learning and computer vision to identify bumble bee species from images. *Scientific Reports*, *11*(1). <https://doi.org/10.1038/s41598-021-87210-1>
- Tan, A., Zhou, G., & He, M. (2020). Rapid Fine-Grained Classification of Butterflies Based on FCM-KM and Mask R-CNN Fusion. *IEEE Access*, *8*, 124722–124733. <https://doi.org/10.1109/ACCESS.2020.3007745>
- Tanaka, F. H. K. dos S., & Aranha, C. (2019). *Data Augmentation Using GANs*. <http://arxiv.org/abs/1904.09135>
- Tang, H., Wang, B., & Chen, X. (2020). Deep learning techniques for automatic butterfly segmentation in ecological images. *Computers and Electronics in Agriculture*, *178*, 105739. <https://doi.org/10.1016/j.compag.2020.105739>
- Theivaprakasham, H. (2021). Identification of Indian butterflies using Deep Convolutional Neural Network. *Journal of Asia-Pacific Entomology*, *24*(1), 329–340. <https://doi.org/10.1016/j.aspen.2020.11.015>
- Wahlteinez, O., & Wahlteinez, S. J. (2024). An open-source general purpose machine learning framework for individual animal re-identification using few-shot learning. *Methods in Ecology and Evolution*, *15*(2), 373–387. <https://doi.org/https://doi.org/10.1111/2041-210X.14278>

- Wang, J., Ji, L., Liang, A., & Yuan, D. (2012). The identification of butterfly families using content-based image retrieval. *Biosystems Engineering*, *111*(1), 24–32. <https://doi.org/10.1016/j.biosystemseng.2011.10.003>
- Xu, M., Yoon, S., Fuentes, A., & Park, D. S. (2023). A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. *Pattern Recognition*, *137*, 109347. <https://doi.org/10.1016/j.patcog.2023.109347>
- Xu, W., Li, X., Deng, Y., Lam, W., & Bing, L. (n.d.). *PeerDA: Data Augmentation via Modeling Peer Relation for Span Identification Tasks* (Vol. 1). Long Papers.
- Xue, A., Li, F., & Xiong, Y. (2019). Automatic Identification of Butterfly Species Based on Gray-Level Co-occurrence Matrix Features of Image Block. *Journal of Shanghai Jiaotong University (Science)*, *24*(2), 220–225. <https://doi.org/10.1007/s12204-018-2013-y>
- Yasmin, R., Das, A., Rozario, L. J., & Islam, M. E. (2023). Butterfly detection and classification techniques: A review. In *Intelligent Systems with Applications* (Vol. 18). Elsevier B.V. <https://doi.org/10.1016/j.iswa.2023.200214>
- Yilmaz, K., Kayci, L., Ertugrul, O. F., Tekin, R., Yılmaz, K., Lokman, K., Faruk, E. Ö., & Ramazan, T. (2015). Identification of Butterfly Species with Rough Set Approach Based on Textural Features. In *Avestia Publishing International Journal of Computer Vision* (Vol. 1). <http://creativecommons.org/licenses/by/3.0>
- Yun, S., Han, D., Chun, S., Oh, S. J., Yoo, Y., & Choe, J. (2019). CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6022–6031. <https://doi.org/10.1109/ICCV.2019.00612>

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond Empirical Risk Minimization. <http://arxiv.org/abs/1710.09412>

Zhao, R., Li, C., Ye, S., & Fang, X. (2019a). Butterfly Recognition Based on Faster R-CNN. *Journal of Physics: Conference Series*, 1176, 032048. <https://doi.org/10.1088/1742-6596/1176/3/032048>

Zhu, L., & Spachos, P. (2019). Butterfly Classification with Machine Learning Methodologies for an Android Application. 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 1–5. <https://doi.org/10.1109/GlobalSIP45357.2019.8969441>