KLASIFIKASI CITRA HEWAN KHAS SUKU DAYAK DENGAN MENGGUNAKAN CNN

TUGAS AKHIR

Disusun Untuk Memenuhi Syarat Kelulusan Program Strata I
pada Sekolah Tinggi Manajemen Informatika dan Komputer
(STMIK) Palangkaraya

OLEH

JEKI
C2155201001
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2025

PERSETUJUAN

KLASIFIKASI CITRA HEWAN KHAS SUKU DAYAK DENGAN MENGGUNAKAN CNN

Tugas Akhir Ini Telah Disetujui dan Disahkan

Pembimbing I,	Pembimbing II,
Dr. Maura Widyaningsih, S.Kom., M.Cs.	Lili Rusdiana, M.Kom.
NIK. 197502232009001	NIK. 198707282011007

Mengetahui

Ketua STMIK Palangkaraya,

Suparno, M.Kom

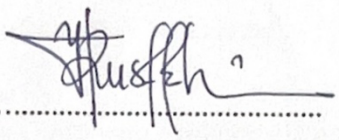
NIK. 196901041995105

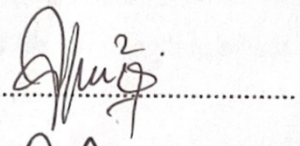
PENGESAHAN

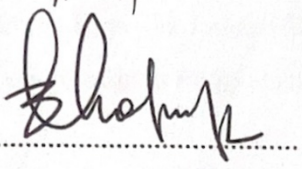
KLASIFIKASI CITRA HEWAN KHAS SUKU DAYAK DENGAN MENGGUNAKAN CNN

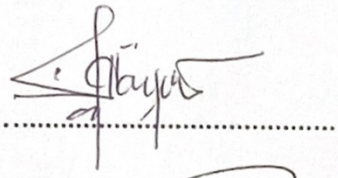
Tugas Akhir ini Telah Diujikan, Dinilai, dan Disahkan
Oleh Tim Penguji pada Tanggal 21 Januari 2025

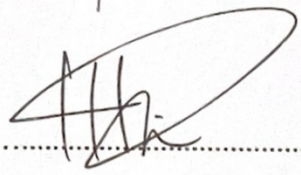
Tim Penguji Tugas Akhir :

1. Sulistyowati, S.Kom., M.Cs.
   Ketua ..... 

2. Elok Faiqotul Himmah, S.Si., M.Sc.
   Sekretaris ..... 

3. Hotmian Sitohang, M.Kom.
   Anggota ..... 

4. Dr. Maura Widyaningsih, S.Kom., M.Cs.
   Anggota ..... 

5. Lili Rusdiana, M.Kom.
   Anggota ..... 

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama Mahasiswa : **JEKI**
N I M : **C2155201001**

menyatakan bahwa Tugas Akhir dengan judul :

KLASIFIKASI CITRA HEWAN KHAS SUKU DAYAK
DENGAN MENGGUNAKAN CNN

adalah hasil karya saya dan bukan merupakan duplikasi sebagian atau seluruhnya dari karya orang lain, kecuali bagian yang sumber informasi dicantumkan.

Pernyataan ini dibuat dengan sebenar-benarnya secara sadar dan bertanggungjawab dan saya bersedia menerima sanksi pembatalan Tugas Akhir apabila terbukti melakukan duplikasi terhadap Tugas Akhir atau karya ilmiah lain yang sudah ada.

Palangka Raya, 18 Januari 2025

Yang Membuat Pernyataan,



JEKI

MOTTO DAN PERSEMBAHAN

"Jangan pernah menyerah, setiap langkah kecil yang diambil dengan ketekunan akan membawa kita lebih pada tujuan yang besar"

Dengan rasa syukur dan berbahagia, karya ini saya persembahkan kepada:

1. Kedua orang tua penulis yang tercinta dan terkasih, yang senantiasa selalu memberikan semangat, dukungan, doa dan cinta tanpa batas kepada penulis.
2. Ibu Dr. Maura Widyaningsih, S.Kom., M.Cs., dan Ibu Lili Rusdiana, M.Kom., selaku dosen pembimbing yang selalu sabar dalam memberikan bimbingan, saran, dan motivasi selama pengerjaan tugas akhir ini.
3. Teman-teman kelas TI-A Angkatan 2021, yang selalu memberikan semangat dan kebersamaan yang berarti sepanjang masa perkuliahan.

INTISARI

Jeki, C2155201001, 2025. Klasifikasi Citra Hewan Khas Suku Dayak Dengan Menggunakan CNN, Pembimbing I Dr. Maura Widyaningsih, S.Kom., M.Cs, Pembimbing II Lili Rusdiana, M.Kom.

Fauna khas Dayak, seperti Orangutan, Burung Tingang, Bekantan, Lutung Merah, dan Owa Kalimantan, memiliki peranan penting dalam menjaga keseimbangan ekosistem dan melestarikan budaya masyarakat Kalimantan Tengah. Namun, rendahnya tingkat kesadaran generasi muda terhadap keberadaan fauna tersebut menjadi hambatan utama dalam upaya pelestariannya.

Penelitian ini bertujuan untuk mengembangkan teknologi berbasis *Convolutional Neural Network* yang mampu mendukung pelestarian fauna khas Dayak melalui sistem klasifikasi otomatis yang lebih efisien dan akurat.

Metode yang digunakan adalah pendekatan *transfer learning* dengan arsitektur CNN *InceptionV3*, disertai pengumpulan *dataset* dari berbagai sumber. *Dataset* ini diproses melalui tahap *resizing*, normalisasi, dan augmentasi untuk meningkatkan kualitas dan keberagaman data.

Hasil penelitian menunjukkan bahwa model yang dikembangkan mampu mengklasifikasikan fauna khas Dayak dengan tingkat akurasi hingga 99,5% dan *F1-Score* sebesar 1,00. Teknologi ini diharapkan dapat meningkatkan kesadaran masyarakat akan pentingnya menjaga kekayaan alam serta warisan budaya Dayak.

Kata kunci : *convolutional neural network*, klasifikasi fauna dayak, *transfer learning*, *inceptionv3*.

ABSTRACT

Jeki, C2155201001, 2025. *Classification of Traditional Dayak Animal Images Using CNN*, *Advisor* I Dr. Maura Widyaningsih, S.Kom., M.Cs., *Advisor* II Lili Rusdiana, M.Kom.

Dayak's distinctive fauna, such as the Orangutan, Hornbill (Burung Tingang), Proboscis Monkey (Bekantan), Red Langur (Lutung Merah), and Bornean Gibbon (Owa Kalimantan), play a crucial role in maintaining ecosystem balance and preserving the cultural heritage of the Central Kalimantan community. However, the low awareness among younger generations about these species poses a significant challenge to conservation efforts.

This research aims to develop a Convolutional Neural Network based technology to support the conservation of Dayak's distinctive fauna through a more efficient and accurate automatic classification system.

The method employed involves a transfer learning approach using the CNN InceptionV3 architecture, along with dataset collection from various sources. The dataset was processed through resizing, normalization, and augmentation stages to enhance its quality and diversity.

The results of this study indicate that the developed model can classify Dayak's distinctive fauna with an accuracy of up to 99.5% and an F1-Score of 1,00. This technology is expected to raise public awareness of the importance of preserving natural wealth and the cultural heritage of the Dayak community.

Keywords: convolutional neural network, dayak fauna classification, transfer learning, inceptionv3.

KATA PENGANTAR

Puji dan syukur atas berkat dan juga pernyertaan Tuhan yang luar biasa, sehingga penulis dapat menyelesaikan Tugas Akhir ini yang berjudul "KLASIFIKASI CITRA HEWAN KHAS SUKU DAYAK DENGAN MENGGUNAKAN CNN". Tugas Akhir ini disusun untuk memenuhi syarat dalam memperoleh gelar Sarjana Komputer di Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya.

Pada kesempatan ini penulis juga menyampaikan terimakasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bimbingan, motivasi, semangat, dalam menyelesaikan penulisan Tugas Akhir ini, yaitu kepada :

1. Dr. Maura Widyaningsih, S.Kom., M.Cs, selaku dosen Pembimbing I yang telah meluangkan waktunya untuk memberikan bimbingan dalam penyelesaian Tugas Akhir ini.
2. Lili Rusdiana, M.Kom., selaku dosen Pembimbing II dalam penulisan Tugas Akhir ini, yang juga banyak memberikan saran dan masukan dalam penulisan Tugas Akhir ini.

Penulis mengharapkan masukan, saran dan kritik yang bersifat membangun untuk menyempurnakan Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi penulis dan pembaca terutama untuk untuk menambah ilmu pengetahuan.

Palangka Raya, 18 Januari 2025

Penulis

DAFTAR ISI

PERSETUJUAN ..... **ii**
PENGESAHAN ..... **iii**
PERNYATAAN KEASLIAN ..... **iv**
MOTTO DAN PERSEMBAHAN ..... **v**
INTISARI ..... **vi**
ABSTRACT ..... **vii**
KATA PENGANTAR ..... **viii**
DAFTAR ISI ..... **ix**
DAFTAR TABEL ..... **x**
DAFTAR GAMBAR ..... **xi**
DAFTAR LAMPIRAN ..... **xii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang Masalah 1
1.2 Perumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan dan Manfaat 4
1.5 Sistematika Penulisan 5
BAB II TINJAUAN PUSTAKA **7**
2.1 Kajian Teori 7
2.2 Penelitian Yang Relevan 27
BAB III METODOLOGI PENELITIAN **28**
3.1 Tinjauan Umum 28
3.2 Jenis penelitian 28
3.3 Desain Penelitian 29
3.4 Instrumen Penelitian 37
3.5 Analisis Kebutuhan 37
3.5.1 Analisis Data 38
3.5.2 Analisis Proses 41
BAB IV HASIL DAN PEMBAHASAN **45**
4.1 Hasil 45
4.2 Pembahasan 69
BAB V KESIMPULAN DAN SARAN **73**
5.1 Kesimpulan 73
5.2 Saran 74
DAFTAR PUSTAKA

DAFTAR TABEL

Tabel 2. 1 Metriks yang umum digunakan..... 24
Tabel 2. 2 *Library* & *Framework Python* 25
Tabel 2. 3 Penelitian yang Relevan..... 27
Tabel 3. 1 Perangkat Keras 38
Tabel 3. 2 Perangkat Lunak 38
Tabel 3. 3 Distribusi Jumlah Data..... 39
Tabel 4. 1 *Hyperparameter* yang di-*tunning*..... 46
Tabel 4. 2 Augmentasi data yang diterapkan..... 46
Tabel 4. 3 Akurasi dari semua model..... 47
Tabel 4. 4 Pengujian Model 67

DAFTAR GAMBAR

Gambar 2. 1 Orangutan 8
Gambar 2. 2 Burung Tingang 9
Gambar 2. 3 Bekantan 9
Gambar 2. 4 Lutung Merah 10
Gambar 2. 5 Owa Kalimantan 11
Gambar 2. 6 Arsitektur CNN 13
Gambar 2. 7 Arsitektur *InceptionV3* 16
Gambar 2. 8 *Confusion Matrix* 23
Gambar 3. 1 Desain Penelitian 29
Gambar 3. 2 *Preprocessing* 31
Gambar 3. 3 Pembagian *Dataset* 33
Gambar 3. 4 Implementasi Model *InceptionV3* 34
Gambar 3. 5 Arsitektur CNN *InceptionV3* 42
Gambar 4. 1 Grafik Model Pertama 50
Gambar 4. 2 Grafik Model Ketiga 51
Gambar 4. 3 Grafik Model Keempat 52
Gambar 4. 4 Grafik Model Keempat 53
Gambar 4. 5 *Classification Report* Model Pertama 55
Gambar 4. 6 *Classification Report* Model Kedua 56
Gambar 4. 7 *Classification Report* Model Ketiga 57
Gambar 4. 8 *Clasisification Report* Model Keempat 58
Gambar 4. 9 *Confusion Matrix* Model Pertama 60
Gambar 4. 10 *Confusion Matrix* Model Kedua 61
Gambar 4. 11 *Confusion Matrix* Model Ketiga 63
Gambar 4. 12 *Confusion Matrix* Model Keempat 64

DAFTAR LAMPIRAN

Lampiran 1. Surat tugas pembimbing Tugas Akhir
Lampiran 2. Lembar konsultasi bimbingan Tugas Akhir
Lampiran 3. Surat tugas penguji sidang
Lampiran 4. Berita acara penilaian sidang TA
Lampiran 5. Sampel *Dataset*
Lampiran 6. *Listing* program

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Teknologi informasi berkembang pesat setiap harinya dan membawa dampak besar pada berbagai aspek kehidupan manusia, termasuk pendidikan dan pelestarian budaya (Zhao, et al., 2024). Di Indonesia, salah satu tantangan besar yang dihadapi adalah kurangnya kesadaran generasi muda terhadap fauna lokal, terutama di wilayah Kalimantan Tengah, yang merupakan rumah bagi berbagai spesies hewan khas, seperti burung tingang, owa kalimantan, orangutan, lutung merah, dan bekantan. Fauna ini memiliki nilai ekologis dan budaya yang penting, terutama bagi masyarakat Dayak. Namun, minimnya materi pembelajaran berbasis teknologi yang menarik dan interaktif menyebabkan pengetahuan tentang fauna ini mulai memudar di kalangan generasi muda.

Sebagai solusi, teknologi kecerdasan buatan (AI), khususnya *Convolutional Neural Network* (CNN), dapat digunakan untuk membantu dalam klasifikasi dan pengenalan citra satwa liar. CNN telah terbukti sangat efektif dalam mengenali objek pada gambar dan mampu menangani klasifikasi objek yang kompleks, seperti hewan (Alzubaidi, et al., 2021). Penelitian ini memanfaatkan CNN, dengan pendekatan *transfer learning*, untuk mengembangkan model klasifikasi citra hewan khas Dayak. Pendekatan ini memungkinkan model

yang dilatih pada *dataset* besar untuk digunakan pada *dataset* yang lebih kecil, sehingga meningkatkan akurasi dan kecepatan dalam proses pelatihan.

Metode yang digunakan meliputi beberapa tahapan kunci. Pertama, pengumpulan *dataset* citra hewan khas Dayak dari berbagai sumber. Tahapan selanjutnya adalah *preprocessing*, yang meliputi pemrosesan awal seperti *resizing*, normalisasi, dan augmentasi data untuk meningkatkan variasi data dan mengurangi bias model. Implementasi model CNN dengan strategi *transfer learning* kemudian dilakukan untuk membangun model klasifikasi. Setelah model selesai dilatih, tahap evaluasi dilakukan dengan mengukur kinerja model menggunakan metrik seperti akurasi**,** presisi**,** *recall***,** dan *F1-Score* untuk mendapatkan gambaran yang komprehensif tentang kemampuan model dalam melakukan klasifikasi.

Penelitian ini berfokus pada pengembangan model klasifikasi citra hewan Dayak yang efektif dengan memanfaatkan teknologi *Convolutional Neural Network* (CNN). Model ini akan dilatih untuk dapat mengenali karakteristik visual yang spesifik dari setiap spesies, seperti bentuk tubuh, pola warna, tekstur bulu, dan ciri-ciri morfologi unik lainnya untuk membedakan satu spesies dengan spesies lainnya. Misalnya, Burung Tingang dapat diidentifikasi dari bentuk paruh yang melengkung dan pola warna yang kontras pada bulunya. Melalui pelatihan terhadap *dataset* yang telah disiapkan, model CNN ini diharapkan dapat secara efektif mengidentifikasi dan mengklasifikasikan fauna lokal berdasarkan ciri-ciri fisik tersebut. Oleh karena itu, hasil penelitian ini ditargetkan tidak hanya memberikan kontribusi dalam pengembangan teknologi klasifikasi citra berbasis CNN, tetapi

juga dapat menjadi landasan dasar untuk penelitian lebih lanjut di bidang konservasi dan pelestarian fauna lokal Kalimantan Tengah yang mempunyai nilai budaya dan ekologi yang tinggi

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah ″Bagaimana cara memperbaiki model klasifikasi citra hewan khas Dayak menggunakan *Convolutional Neural Network* (CNN) yang mampu menyelesaikan permasalahan ketidakseimbangan data menjadi optimal pada *dataset* yang terbatas. Seperti keterbatasan dalam ukuran, perubahan posisi objek, dan jumlah kelas tidak sama″.

1.3 Batasan Masalah

Agar penelitian ini tetap fokus pada permasalahan yang ada dan tidak melebar dari topik pembahasan, beberapa batasan masalah yang ditetapkan adalah sebagai berikut:

a. Penelitian berfokus pada klasifikasi citra hewan khas Dayak Ngaju, seperti Kahiu, Burung Tingang, Bakara, Kalasi, dan Kalawet.
b. *Dataset* yang digunakan dalam penelitian ini berasal dari sumber terbuka umum (*public*) seperti *Kaggle*, *Image.CV* dan *RoboFlow*.
c. Arsitektur menggunakan *Convolutional Neural Network* (CNN) dengan menerapkan model *InceptionV3*.
d. Penyesuaian *hyperparameter* utama, seperti *learning rate*, jumlah *epoch*, dan *batch size*, untuk meningkatkan akurasi model CNN.

e. Evaluasi performa menggunakan *confusion matrix* untuk menentukan akurasi, presisi, *recall*, dan *F1-Score*.

1.4 Tujuan dan Manfaat

f. Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1) Memperbaiki model klasifikasi berbasis *Convolutional Neural Network* (CNN), guna memberikan solusi praktis dalam pengenalan hewan fauna khas suku Dayak.
2) Menguji dan mengevaluasi kinerja model klasifikasi yang dikembangkan dengan dataset gambar terbatas, menggunakan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-Score*, serta menganalisis efektivitas teknik *transfer learning* dalam meningkatkan akurasi klasifikasi hewan khas suku Dayak.
3) Memprediksi kelas hewan khas suku Dayak dengan akurat menggunakan input gambar yang diambil dari internet maupun dari jepretan kamera secara langsung.

g. Manfaat

1) Ilmu pengetahuan dan teknologi

Penelitian ini memperkaya pemahaman mengenai penerapan CNN dalam mengklasifikasikan citra fauna khas Dayak yang masih jarang terdokumentasi secara digital. Selain itu, hasil penelitian ini dapat menjadi acuan bagi peneliti lain yang ingin mengembangkan teknologi kecerdasan buatan dalam bidang pengenalan citra.

2) Pelestarian budaya

Dengan adanya klasifikasi citra fauna khas Dayak, penelitian ini berkontribusi dalam mendokumentasikan keanekaragaman hayati yang memiliki keterkaitan dengan budaya lokal, sehingga dapat mendukung upaya konservasi fauna khas daerah tersebut.

3) Pendidikan

Hasil klasifikasi ini dapat digunakan sebagai materi ajar tambahan dalam dunia pendidikan, khususnya untuk meningkatkan pemahaman mengenai spesies fauna khas Dayak serta pentingnya menjaga kelestarian satwa lokal.

4) Masyarakat umum

Penelitian ini menyajikan informasi yang lebih terstruktur dan akurat terkait fauna khas Dayak, sehingga dapat meningkatkan kesadaran masyarakat akan keunikan serta urgensi perlindungan terhadap satwa yang ada di Kalimantan Tengah.

1.5 Sistematika Penulisan

Adapun sistematika penulisan dalam penulisan tugas akhir ini terdiri dari bebrapa bab dan masing-masing bab membahas dan menguraikan pokok permasalahan yang berbeda, sebagai gambaran disini penulis menyertakan garis-garis besarnya yaitu :

BAB I PENDAHULUAN

Bab ini berisikan latar belakang masalah, rumusan masalah, Batasan masalah, tujuan dan manfaat, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisikan tentang tinjauan pustaka yang diambil dari penelitian yang relevan beserta susunan kajian teori yang disesuaikan dengan tema Tugas Akhir.

BAB III METODOLOGI PENELITIAN

Bab ini berisikan tentang tahapan yang dilakukan peneliti dalam mengumpulkan informasi atau data yang dibutuhkan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisikan tentang hasil dari penelitian dan pembahasan dari hasil yang didapat.

BAB V KESIMPULAN DAN SARAN

Bab ini berisikan tentang kesimpulan dan saran yang diberikan untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Teori

Pada bab ini, penulis telah menggali informasi yang ada sebelumnya tentang teori yang mendukung atau yang menjadi dasar di dalam penelitian ini. Teori-teori tersebut didapat dari berbagai sumber dan literatur.

2.1.1 Fauna Khas Dayak

Fauna khas Kalimantan Tengah seperti orangutan tergolong satwa yang terancam punah dan membutuhkan perhatian khusus untuk pelestariannya. Bagi masyarakat Dayak, satwa-satwa ini tidak hanya penting secara ekologis, tetapi juga memiliki nilai budaya dan spiritual. Upaya pelestarian fauna Dayak memiliki arti yang mendalam, baik dalam menjaga kelestarian lingkungan maupun menjaga warisan budaya suku Dayak yang seringkali memiliki kearifan lokal yang berkaitan dengan fauna di sekitarnya (Indrawan, et al,. 2007).

Upaya konservasi yang dilakukan dengan melibatkan masyarakat lokal, seperti masyarakat Dayak, sangat penting dalam melestarikan fauna ini. Dengan adanya keterlibatan masyarakat lokal, tidak hanya membantu dalam menjaga keberadaan spesies ini, namun juga menguatkan hubungan antara manusia dengan alam yang telah berlangsung secara turun-temurun oleh suku Dayak.

Berikut ini adalah beberapa hewan khas Dayak yang terancam punah dan harus dilestarikan:

a. Kahiu (Orangutan)



Gambar 2. 1 Orangutan
Sumber: KLHK, 2017

Orangutan Kalimantan berbeda dengan orangutan di pulau lain, seperti Sumatera. Orangutan Kalimantan lebih banyak menghabiskan hidupnya di tanah daripada di pohon. Orangutan Kalimantan juga biasanya memiliki warna bulu yang lebih gelap dari Orangutan Sumatera dan memiliki bantalan pipi yang melebar di wajah mereka (Bellboy, 2024).

Penurunan luas hutan di Kalimantan tentu berimbas pada kehidupan Orangutan Kalimantan secara langsung. Hal ini terbukti dengan berkurangnya populasi Orangutan Kalimantan setiap tahunnya.

b. Burung Tingang (Enggang)



Gambar 2. 2 Burung Tingang
Sumber: Media Budaya Indonesia, 2018

Burung Rangkong atau disebut juga enggang merupakan simbol Kalimantan yang dapat hidup hingga 35 tahun. Rangkong dewasa berukuran hingga 1,6 meter, untuk membedakan jenis kelamin rangkong atau enggang ini, ciri-ciri paruh yang besar dengan warna mata yang berbeda membedakan jantan dan betina. Di kalangan suku Dayak, burung ini juga dianggap sebagai simbol kepemimpinan (Bellboy, 2024).

Keberadaan burung ini juga telah dinyatakan sangat terancam akibat aktivitas pembalakan liar yang berkepanjangan di Kalimantan.

c. Bakara (Bekantan)



Gambar 2. 3 Bekantan
Sumber: Wijaya, 2023

Bekantan merupakan salah satu jenis primata yang memiliki karakteristik tubuh yang unik. Bekantan atau dikenal juga dengan nama latin *Nasalis larvatus* merupakan jenis monyet yang memiliki hidung lebih panjang dan besar dibandingkan dengan primata lainnya. Bekantan dewasa memiliki panjang sekitar 60 hingga 75 cm (Bellboy, 2024).

Habitat asli primata ini adalah hutan mangrove, tetapi jumlah lahan mangrove yang berkurang membuat hewan endemik ini terancam punah dan mencari tempat baru untuk tinggal.

d. Kalasi (Lutung Merah)



Gambar 2. 4 Lutung Merah
Sumber: Istiqomah, 2024

Hewan khas Kalimantan ini, yang termasuk dalam keluarga Kera, memiliki ekor panjang dan bulu berwarna kemerahan. Lutung Merah biasanya mengonsumsi dedaunan dan biji-bijian. Hidupnya dihabiskan dalam kelompok berjumlah tujuh hingga delapan ekor. Lutung Merah tampak sangat aktif saat mencari makan dan bergerak di siang hari (Bellboy, 2024).

Habitat asli lutung merah adalah di hutan primer dan sekunder. Penebangan hutan secara liar dan perluasan lahan secara besar-besaran di

Kalimantan. Hal ini tentu menyebabkan populasi lutung merah di Kalimantan terancam. Tidak jarang jika lutung merah masuk ke perkampungan penduduk untuk mencari makan.

e. Kalawet (Owa Kalimantan)



Gambar 2. 5 Owa Kalimantan
Sumber: gibbonesia, 2022

Meskipun masih termasuk dalam keluarga Kera, Owa Kalimantan adalah salah satu hewan khas Kalimantan yang dapat ditemukan di berbagai tempat di Pulau Kalimantan. Hewan ini biasanya disebut dengan beberapa sebutan, seperti Kalawet dan Lempiau, dan sebutan-sebutan ini biasanya berbeda tergantung pada tempat hewan tersebut tinggal (Bellboy, 2024).

Hewan ini juga terancam punah akibat dari perburuan dan penebangan hutan liar dan peluasan lahan membuat habitat tempat tinggal hewan ini semakin sedikit.

2.1.2 Citra

Citra merupakan bentuk informasi yang diperlukan manusia selain teks, suara dan video. Informasi yang terkandung dalam sebuah citra di interpresentasikan berbeda-beda oleh manusia satu dengan lainnya (Maulana, et al., 2022).

Dari definisi citra diatas penulis menyimpulkan bahwa citra merupakan representasi visual dari objek, adegan, atau fenomena yang digunakan untuk menyampaikan informasi dengan baik dan efektif. Citra dapat menangkap detail visual yang sulit dijelaskan dengan teks atau suara, dan cara seseorang melihat citra berbeda tergantung pada latar belakang, pengalaman, dan konteks di mana citra itu digunakan.

2.1.3 Klasifikasi Gambar

Klasifikasi gambar merupakan proses mengelompokkan objek berdasarkan kelas (*class*) tertentu. Sehingga dapat mengenali objek dengan mudah. Klasifikasi gambar adalah proses mengidentifikasi kelas informasi dari gambar raster *multiband* (Diantoro & Ardriansyah, 2019).

Dari penjelasan di atas penulis menyimpulkan bahwa klasifikasi gambar adalah langkah penting dalam analisis dokumen, dengan tujuan untuk mengatur dan menginterpretasikan informasi visual secara sistematis

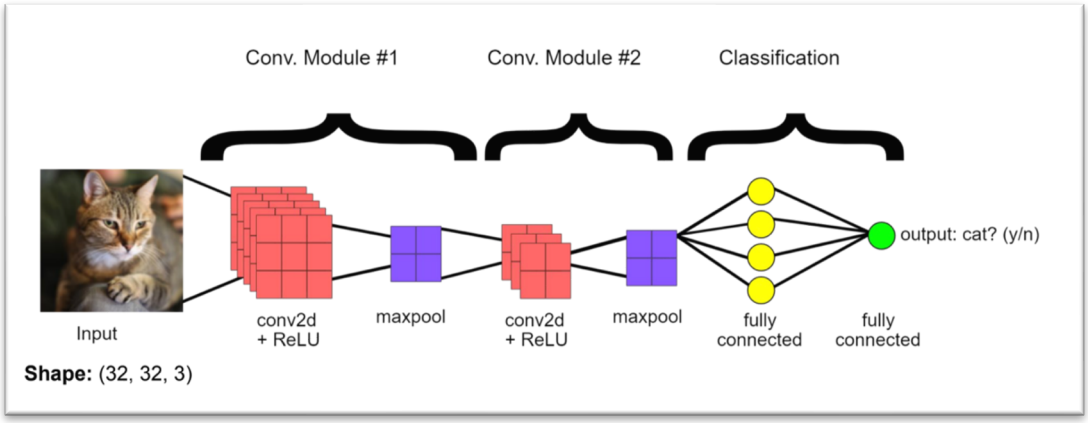
2.1.4 *Deep Learning*

Deep learning adalah sebuah kelas algoritma *machine learning* yang disebut *neural network*. *Neural network* adalah model matematika yang terinspirasi oleh struktur otak. *Deep learning* memungkinkan algoritma *neural network* bekerja dengan sangat baik dalam membangun model prediksi di

sekitar masalah yang kompleks seperti *computer vision* dan pemodelan bahasa. (Bisong, 2019).

Berdasarkan definisi diatas *deep learning* menggunakan lapisan-lapisan jaringan saraf untuk secara otomatis mengekstrak karakteristik dari data yang kompleks tanpa memerlukan campur tangan manusia untuk memilih karakteristik yang relevan. Setiap lapisan jaringan saraf mendefinisikan representasi yang lebih abstrak dari data mentah, sehingga memungkinkan model untuk menangkap pola yang sangat kompleks dan terperinci.

2.1.5 Convolutional Neural Network (CNN)



Gambar 2. 6 Arsitektur CNN
Sumber: Dumane, 2020

Convolution Neural Network merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra (Azhar, 2021).

Dari definisi di atas dapat disimpulkan bahwa CNN lebih efektif dibangun untuk mengenali pola dan fitur dalam gambar. CNN dapat

mengekstrak fitur lokal dari gambar, seperti tepi, sudut, dan tekstur, yang sangat penting untuk deteksi objek, dengan menggunakan proses konvolusi. Lapisan konvolusi, *pooling*, dan lapisan yang terhubung sepenuhnya adalah beberapa lapisan yang membentuk struktur CNN. Lapisan-lapisan ini digabungkan untuk meminimalkan kompleksitas data dan meningkatkan akurasi prediksi.

2.1.6 *Transfer Learning*

Transfer learning adalah bentuk pembelajaran mesin di mana model yang telah dilatih sebelumnya yang dilatih pada tugas tertentu digunakan kembali sebagai titik awal dan disesuaikan dengan tugas lain dalam kumpulan data yang berbeda (Ebbenoj, et al., 2022).

Transfer learning memungkinkan penggunaan waktu dan sumber daya yang sangat efisien dalam pelatihan model karena model yang telah dilatih sebelumnya dapat mengambil manfaat dari pengetahuan yang telah ada. Proses ini sangat membantu ketika jumlah data yang tersedia untuk tugas-tugas baru terbatas. Akibatnya, melatih model dari awal dapat menghasilkan hasil yang terlalu pas atau kurang ideal. Dengan menggunakan model yang sudah terlatih, pengguna dapat melakukan *fine tuning* atau penyesuaian parameter pada model untuk meningkatkan kinerja dalam konteks tugas yang berbeda. *Transfer learning* telah terbukti efektif dalam berbagai aplikasi, seperti klasifikasi citra, pembelajaran bahasa, dan analisis sentimen, di mana model yang telah dikembangkan sebelumnya dapat dengan cepat diadaptasi untuk mengatasi masalah baru dan spesifik. Pendekatan ini tidak hanya mempercepat proses pengembangan model, tetapi juga meningkatkan hasil dengan memanfaatkan pengetahuan yang sudah ada dalam model.

2.1.7 *InceptionV3*

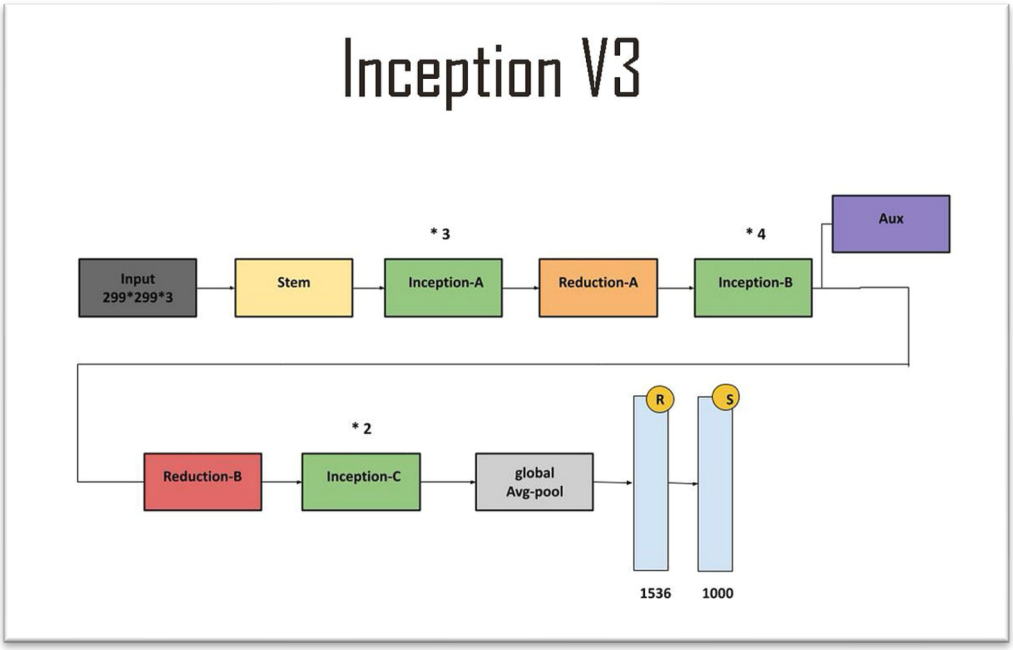
Metode *InceptionV3* adalah arsitektur *convolution neural network* yang dikembangkan oleh tim peneliti di *Google* pada tahun 2015, dan merupakan pengembangan dari model *InceptionV1* dan *InceptionV2*. Tujuan utama dari *InceptionV3* adalah untuk meningkatkan kecepatan komputasi dan kinerja model dalam pengenalan gambar (Widyadhana, et al., 2024). Beberapa fitur utama dari *InceptionV3* antara lain:

1) Modul Inception: Menggabungkan berbagai filter konvolusi secara paralel untuk mengekstrak fitur gambar dengan berbagai ukuran dan skala, menghasilkan representasi gambar yang lebih kaya.
2) Aksesori Baru: Menambahkan *AveragePooling* sebelum lapisan *softmax* untuk mengurangi *overfitting* dan meningkatkan akurasi pada *dataset* beragam.
3) Reduksi Dimensi: Menggunakan konvolusi 1x1 untuk mengurangi jumlah saluran pada lapisan lebih dalam, sehingga mengurangi kompleksitas dan meningkatkan kecepatan komputasi.
4) *Batch Normalization*: Mempercepat konvergensi pelatihan dan mengurangi ketergantungan pada inisialisasi parameter dengan teknik normalisasi *batch*.
5) Augmentasi Data: Memanfaatkan augmentasi data selama pelatihan untuk meningkatkan kemampuan generalisasi model dan mencegah *overfitting*.

6) *Pretraining*: Dilatih terlebih dahulu pada *dataset* besar seperti *ImageNet* untuk memperoleh representasi fitur yang baik, meningkatkan kinerja, dan mempercepat konvergensi pada tugas klasifikasi spesifik.

Metode *InceptionV3* telah terbukti sangat efektif dalam berbagai tugas pengenalan gambar, seperti klasifikasi gambar, deteksi objek, dan segmentasi gambar. Keunggulannya terletak pada kemampuannya untuk menghasilkan representasi fitur yang lebih kaya dengan menggunakan modul *Inception*, sambil tetap memperhatikan efisiensi komputasi.

Berikut adalah arsitektur dari model *InceptionV3* yang dilihat pada gambar 2.7.



Gambar 2. 7 Arsitektur *InceptionV3*
Sumber: Brital, 2021

a. Input Layer

Model dimulai dengan input layer, yang menerima gambar berukuran 299x299 piksel dengan 3 channel warna (RGB). Layer ini bertugas sebagai masukan

utama ke dalam jaringan, di mana gambar akan diproses untuk diekstraksi fitur-fiturnya melalui operasi berikutnya.

b. *Stem*

Bagian awal model, yaitu *Stem*, bertujuan mengekstraksi fitur dasar dari gambar masukan. *Stem* terdiri dari beberapa operasi konvolusi dan pooling yang secara bertahap mengurangi dimensi gambar dari 299x299 menjadi lebih kecil, sambil mempertahankan informasi penting seperti pola-pola dasar pada gambar, seperti tepi dan tekstur.

c. *Inception* A

Lapisan *Inception* A dirancang untuk menangkap informasi dari berbagai skala. Modul ini menggunakan kombinasi filter konvolusi berukuran 1x1, 3x3, dan *pooling* untuk mendeteksi pola-pola yang beragam. Kombinasi ini memungkinkan model mengekstraksi fitur pada level spasial yang berbeda, menghasilkan representasi fitur yang lebih kaya dan kompleks.

d. *Reduction* A

Layer *Reduction* A bertugas mengurangi dimensi spasial gambar secara signifikan dengan operasi *pooling* dan konvolusi. Proses ini mengurangi resolusi gambar, sehingga lebih efisien secara komputasi, tetapi tetap mempertahankan fitur-fitur penting yang telah diekstraksi sebelumnya.

e. *Inception* B

Pada lapisan *Inception* B, proses ekstraksi fitur dilanjutkan dengan struktur yang lebih kompleks dibandingkan *Inception* A. *Filter* konvolusi beragam tetap digunakan, seperti 1x1 dan 3x3, untuk menangkap pola-pola yang lebih

tinggi. Dimensi keluaran modul ini lebih kecil secara spasial tetapi memiliki kedalaman fitur yang lebih besar.

f. *Auxiliary Classifier*

Auxiliary classifier adalah cabang tambahan yang membantu pelatihan model. Dengan memproses keluaran jaringan di bagian tengah, cabang ini memberikan prediksi sementara untuk memperbaiki distribusi gradien pada lapisan awal. Namun, komponen ini hanya digunakan selama pelatihan, bukan pada tahap inferensi.

g. *Reduction* B

Lapisan *Reduction* B adalah langkah reduksi dimensi kedua. Di sini, resolusi spasial kembali dikurangi secara signifikan melalui operasi *pooling* dan konvolusi. Pada tahap ini, gambar hampir sepenuhnya direpresentasikan sebagai abstraksi fitur tingkat tinggi, dengan resolusi sangat kecil tetapi jumlah *channel* yang banyak.

h. *Inception* C

Modul *Inception* C menyempurnakan fitur-fitur kompleks yang telah diekstraksi. Modul ini menggunakan kombinasi filter konvolusi dan *pooling* untuk menangkap pola-pola yang lebih spesifik dari fitur sebelumnya. Dengan resolusi spasial yang sangat kecil, modul ini berfokus pada penguatan representasi fitur.

i. *Global Average Pooling*

Setelah semua fitur diproses, lapisan *Global Average Pooling* digunakan untuk merata-rata setiap *channel* fitur menjadi satu nilai. Proses ini mereduksi fitur

2D menjadi vektor 1D dengan panjang 1536, yang menjadi representasi utama gambar masukan.

j. *Fully Connected Layer*

Lapisan *fully connected* menerima vektor fitur hasil GAP dan menghubungkannya ke 1000 *node*, di mana setiap *node* mewakili satu kelas. Lapisan ini mengintegrasikan semua informasi fitur menjadi representasi akhir sebelum prediksi.

k. *Output Layer*

Lapisan terakhir adalah *softmax*, yang menghasilkan distribusi probabilitas untuk setiap kelas. *Output* berupa vektor dengan 1000 elemen, di mana elemen dengan probabilitas tertinggi akan menjadi prediksi akhir model.

2.1.8 *Preprocessing*

Preprocessing merupakan tahap untuk melakukan sebuah proses awal dalam pengolahan data. Pada tahap ini data yang akan diolah bertujuan untuk menghindarkan dari data yang mengganggu (*noise*) atau data yang tidak konsisten (Alghifari & Juardi, 2021).

Dalam proses ini, berbagai pendekatan digunakan untuk menyiapkan dan meningkatkan kualitas data, yang memungkinkan latihan model yang lebih efektif. Beberapa metode *preprocessing* yang sering digunakan adalah:

a. *Resizing*

Proses mengubah ukuran citra agar sesuai dengan dimensi yang diperlukan oleh model. Hal ini penting untuk memastikan bahwa semua

input citra memiliki ukuran yang konsisten, yang mempermudah pelatihan model.

b. Normalisasi

Teknik yang digunakan untuk mengubah nilai piksel citra agar berada dalam rentang tertentu, biasanya antara 0 dan 1. Proses ini membantu dalam meningkatkan konvergensi selama pelatihan model dengan memastikan bahwa setiap fitur memiliki skala yang sama.

c. Augmentasi

Teknik yang menciptakan variasi dalam *dataset* dengan menggunakan perubahan gambar seperti rotasi, membalik, memperbesar, dan perubahan kontras bertujuan untuk meningkatkan generalisasi model dengan mengeksploitasi variasi yang ada dan mengurangi risiko *overfitting*.

2.1.9 *Hyper Parameter*

Hyperparameter menentukan struktur dan proses pelatihan model. Penyetelan (*tuning*) *hyperparameter* bertujuan untuk menemukan kombinasi nilai yang dapat mengoptimalkan performa model dalam menangani masalah yang diberikan (Goodfellow, et al., 2016).

Hyperparameter adalah parameter yang ditetapkan sebelum proses pelatihan model dan memiliki dampak yang sangat signifikan pada kinerja model. Adapun beberapa contoh *hyperparameter* dalam model *deep learning* antara lain jumlah *epoch*, *batch size*, dan *learning rate*. Berbeda degan parameter model yang dipelajari secara otomatis, *hyperparameter* ini diatur secara manual oleh pembuat model.

Hyperparameter ini dapat dilakukan tunning dengan tujuan untuk meningkatkan kinerja dari model, berikut ini adalah *hyperparameter* yang sangat berpengaruh pada hasil model.

a) *Learning Rate*

Jika *learning rate* yang digunakan terlalu tinggi, model akan sulit untuk mencapai konvergensi, sementara jika *learning rate* yang digunakan terlalu rendah akan membuat pelatihan membutuhkan waktu yang lebih lama. Dengan melakukan tunning yang tepat, *learning rate* juga dapat mempercepat pelatihan dan mencapai hasil yang lebih baik.

b) *Batch Size*

Batch yang lebih kecil dapat menghasilkan generalisasi yang lebih baik tetapi lambat, sementara batch yang lebih besar dapat mempercepat pelatihan tetapi membutuhkan lebih banyak memori. Mengoptimalkan akurasi dan efisiensi dengan mengatur ukuran batch.

c) Jumlah *Epoch*

Mengatur *epoch* dengan benar dapat membantu mencegah *overfitting* dan *underfitting*. Dengan memastikan bahwa model berhenti dilatih pada saat yang optimal dengan *tuning* yang tepat, dapat meningkatkan generalisasi terhadap data baru.

2.1.10 Kompilasi Model

Kompilasi model merupakan proses mempersiapkan model *neural network* untuk pelatihan atau inferensi dengan menentukan arsitekturnya, *optimizer*, *loss function*, dan juga metrik evaluasi yang akan digunakan selama pelatihan (Yu, et al., 2023).

Hal ini memastikan bahwa model dapat secara efisien menghitung lintasan maju dan mundur, dan mengoptimalkan parameter berdasarkan konfigurasi yang dipilih. Proses ini biasanya melibatkan pendefinisian bagaimana model belajar dan meningkatkan akurasinya dari waktu ke waktu.

Menurut *Keras documentation* dan literatur *deep learning*, kompilasi model melibatkan tiga elemen kunci:

1) *Optimizer*

   Algoritma yang digunakan untuk memperbarui bobot model berdasarkan *gradient* dari *loss function*.

2) *Loss Function*

   Fungsi yang digunakan untuk menghitung seberapa jauh prediksi model dari hasil yang diharapkan.

3) *Metrics*

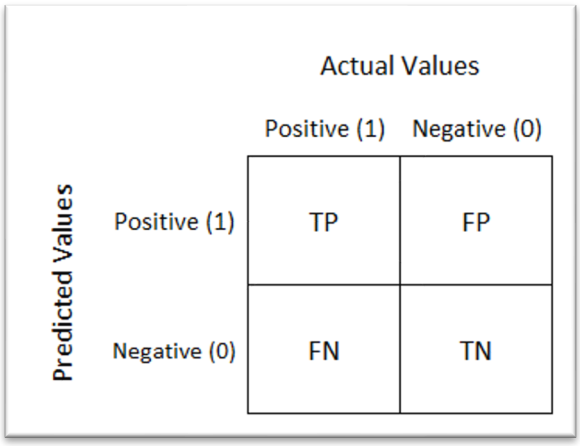
   Metrics yang digunakan untuk mengevaluasi kinerja model, seperti akurasi.

2.1.11 Evaluasi Model

Evaluasi model merupakan pengujian performansi dari sistem yang dibangun. Beberapa cara untuk melakukan evaluasi adalah dengan menghitung akurasi (*accuracy*), presisi (*precision*), dan *recall* dari hasil analisis sistem (Abdullah, et al., 2021).

C*onfusion matrix* merupakan sebuah tabel yang digunakan untuk menampilkan jumlah data uji yang diklasifikasikan dengan benar dan salah, sehingga memudahkan dalam mengevaluasi akurasi suatu sistem klasifikasi.

Dengan menggunakan *confusion matrix*, kita dapat melihat secara detail kinerja suatu sistem klasifikasi dan mengidentifikasi dimana terjadi kesalahan klasifikasi (Nurhidayat & Dewi, 2023).



Gambar 2. 8 *Confusion Matrix*
Sumber: Narkhede, 2018

Dalam Gambar 2.3, terdapat empat istilah yang menggambarkan hasil dari proses klasifikasi dalam *confusion matrix*, yaitu:

1) *True Positive* (TP): Data positif yang diklasifikasikan dengan benar.
2) *True Negative* (TN): Data negatif yang diklasifikasikan dengan benar.
3) *False Positive* (FP): Kesalahan tipe 1, di mana data negatif diklasifikasikan sebagai positif.
4) *False Negative* (FN): Kesalahan tipe 2, di mana data positif diklasifikasikan sebagai negatif.

Evaluasi model adalah proses untuk menilai kinerja model setelah pelatihan menggunakan data uji, yang merupakan data yang belum pernah dilihat sebelumnya. Dalam proses evaluasi ini, beberapa metrik yang digunakan secara luas dapat dilihat dengan lengkap besetar rumusnya pada Tabel 2.1 (Romadloni, et al., 2022).

Tabel 2. 1 Metriks yang umum digunakan

No	Nama Metrik	Rumus	Keterangan
1	Akurasi	$\frac{TP + TN}{TP + FP + FN + TN}$	Persentase prediksi yang benar dari total prediksi yang dilakukan oleh model.
2	Presisi	$\frac{TP}{TP/FP}$	Mengukur proporsi prediksi positif yang benar dibandingkan dengan total prediksi positif. Ini penting dalam situasi di mana biaya kesalahan positif tinggi.
3	*Recall*	$\frac{TP}{TP/FN}$	Mengukur proporsi positif yang benar di antara total aktual positif. Ini menilai kemampuan model untuk menangkap semua kasus positif.
4	*F1-Score*	$2 \times \frac{(Recall \times Presisi)}{(Recall + Presisi)}$	Merupakan rata-rata harmonis dari presisi dan *recall*, memberikan keseimbangan antara kedua metrik ini. *F1-Score* berguna ketika *dataset* tidak seimbang.

2.1.12 Bahasa Pemrograman *Python*

Python adalah bahasa pemrograman komputer yang biasa dipakai untuk membangun situs, *software*/aplikasi, mengotomatiskan tugas dan melakukan analisis data. Bahasa pemrograman ini termasuk bahasa tujuan umum. Artinya,

ia bisa digunakan untuk membuat berbagai program berbeda, bukan khusus untuk masalah tertentu saja (Dicoding, 2023).

Python terkenal dengan sintaksisnya yang lugas dan mudah dipahami, sehingga menjadikannya pilihan yang tepat bagi para pemula dan ahli. Fleksibilitas *Python* membuatnya berguna di banyak bidang, termasuk pengembangan *web*, analisis data, pemrosesan *batch*, dan otomatisasi tugas.

Berikut ini adalah beberapa *framework* dan *libray python* yang digunakan pada penelitian ini, dapat dilihat pada Tabel 2.2.

Tabel 2. 2 *Library* & *Framework Python*

No	Nama	Kegunaan
1	*TensorFlow*	*Platform machine learning* untuk pengembangan model AI
2	*Keras*	API untuk membuat dan melatih *neural network*
3	*Numpy*	Operasi numerik dan *array* multidimensi yang efisien
4	*Matplotlib*	Visualisasi data dalam bentuk grafik dan plot

2.1.13 *Software*

a. *Google Chrome*

Google Chrome adalah *platform browser web* yang dikembangkan oleh *Google*. *Platform browser web* ini pertama kali dirilis pada tahun 2008 untuk *Microsoft Windows* dan yang dibangun dengan komponen *software* gratis dari *Apple Webkit* dan *Mozilla Firefox* (Ranti, 2022).

Google Chrome dirancang untuk menyediakan *browsing* yang cepat, aman, dan efisien. Salah satu fiturnya adalah antarmuka pengguna yang intuitif dan sensitif yang memungkinkan pengguna mengakses berbagai fitur dengan

mudah. *Google Chrome* juga dilengkapi dengan sistem keamanan yang kuat, termasuk pembaruan otomatis untuk melindungi pengguna dari potensi pelanggaran keamanan dan *malware*.

b. *Google Colab*

Google Colab merupakan layanan dari *Google* untuk menjalankan dan mengelola *notebook Jupyter* berbasis *cloud* secara gratis. Ini memungkinkan pengguna untuk menulis dan menjalankan kode *Python*, menyimpan dan berbagi proyek (Zain & Fauzan, 2022).

Google Colab menyediakan lingkungan yang sangat berguna bagi para peneliti, pengembang, dan pelajar untuk melakukan eksperimen menggunakan kode *Python* tanpa memerlukan instalasi perangkat keras lokal.

2.2 Penelitian Yang Relevan

Penelitian yang relevan merupakan acuan bagi peneliti dalam mebuat penelitian. Penelitian yang relevan ini berisikan tetang penelitian orang lain yang dijadikan sebagai sumber atau bahan dalam pembuatan penelitian dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Penelitian yang Relevan

No	Penulis/ Tahun	Topik Penelitian	Metode Pengembangan Penelitian	Teknik yang Digunakan	Hasil
1	Zeng, 2021	*Research on Similar Animal Classification Based on CNN Algorithm*	CNN	Penggunaan *Regularization*, *Epoch* sebanyak 50, *Batch size*nya 32	Akurasi dari model sebesar 96,88%
2	Faizal dkk., 2022	*Wild Animal Classifier Using CNN*	CNN, *K-means*, *InceptionResNet V2*	*K-means segmentation,* menghapus *background* gambar, melakukan *fine tuning*	Model *training accuracy* mencapai 95% dan model *testing accuracy* mencapai 91%
3	Huda dkk., 2023	Analisis Performa Inceptionv3 *Convolutional Network* pada Klasifikasi Varietas Daun *Grapevine*	CNN *InceptionV3*	*Input shape* yang digunakan 300x300x3, dengan *Epoch* 150	Akurasi yang didapat mencapai 99,5%
4	Kosman dkk., 2024	Pengujian Algoritma InceptionV3 dalam Mengidentifikasi Penyakit *Tuberculosis* dan *Pneumonia*	CNN *InceptionV3*	*Epoch* yang digunakan sebanyak 100	Akurasi dicapai rata di atas 99,5%
5	Aprilia dkk., 2024	*Image Captioning* Menggunakan Metode *InceptionV3* dan *Transformer*	CNN *InceptionV3*	*Vocab size*, *Seq Length*, *Embed dim*, *Num heads*, ff *dims*, *Batch size*, *epoch*	Akurasi yang didapat mencapai 80%

BAB III

METODOLOGI PENELITIAN

3.1 Tinjauan Umum

Penelitan ini berfokus pada pengembangan model klasifikasi citra untuk fauna Dayak, dengan menggunakan *dataset* citra yang telah dikurasi secara menyeluruh dari berbagai sumber yang terpercaya. Berhubung penelitian ini tidak dilakukan di lokasi fisik tertentu, data utama diperoleh melalui *platform* publik seperti *Kaggle, Image.CV*, dan *Roboflow* yang menyediakan citra berkualitas tinggi lengkap dengan metadata yang menunjang untuk kebutuhan klasifikasi.

Demi memastikan variasi dan akurasi *dataset*, data tambahan juga dikumpulkan melalui dokumentasi dari lokasi konservasi dan kolaborasi dengan lembaga-lembaga yang memiliki arsip visual fauna Dayak. Kurasi yang cermat dilaksanakan untuk memastikan representasi yang memadai dari setiap spesies, termasuk Burung Tingang, Kahiu, Kalawet, Bakara, dan Kalasi yang memiliki nilai ekologis dan budaya yang sangat penting bagi masyarakat Dayak. Uraian rinci mengenai komposisi, sumber, dan metode kurasi *dataset* ini disajikan untuk memperkuat validitas data yang digunakan dalam pengembangan model klasifikasi.

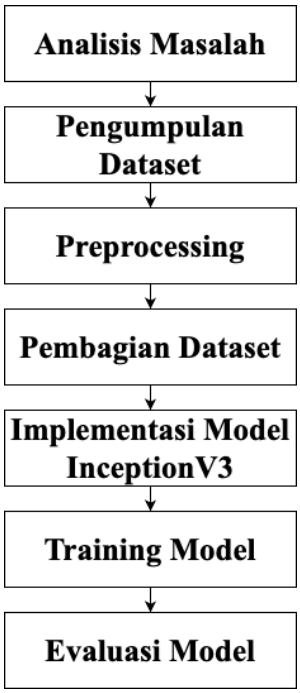
3.2 Jenis penelitian

Jenis Penelitian yang digunakan dalam penelitian ini adalah penelitian kuantitatif. Karena melibatkan pengumpulan dan analisis data numerik dari

hasil evaluasi model, termasuk akurasi, presisi, *recall*, dan *F1-score*. Data ini akan digunakan untuk mengetahui efektivitas klasifikasi yang dihasilkan model.

3.3 Desain Penelitian

Desain penelitian yang ditunjukkan pada Gambar 3.1 merupakan representasi sistematis dari langkah-langkah yang dirancang untuk menyelesaikan penelitian ini. Alur ini mencakup serangkaian tahapan yang dirancang secara terencana dan sistematis untuk memastikan setiap proses berjalan dengan baik dan efisien.



Gambar 3. 1 Desain Penelitian

Secara keseluruhan, desain penelitian ini mencakup beberapa tahapan utama, yaitu sebagai berikut:

a. Analisis Masalah

Penelitian ini dimulai dengan identifikasi masalah utama yang menjadi latar belakang. Penulis menemukan bahwa meskipun fauna khas Dayak

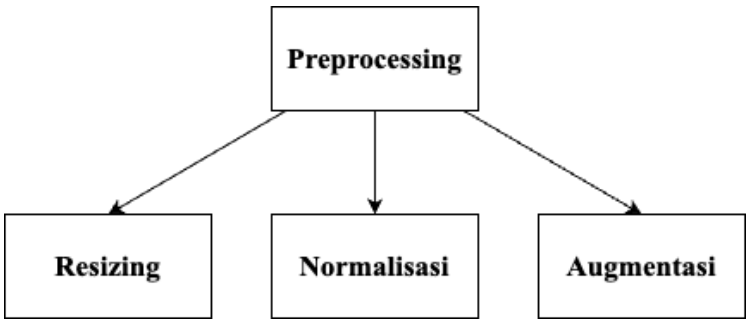
memiliki nilai budaya dan ekologis yang sangat penting, pengetahuan tentang fauna tersebut masih sangat terbatas, khususnya di kalangan generasi muda. Hal ini disebabkan oleh kurangnya upaya pelestarian yang memanfaatkan teknologi *modern* dan interaktif. Sebagai solusi, penulis mengusulkan penerapan teknologi *Convolutional Neural Network* (CNN) untuk melakukan klasifikasi citra fauna khas Dayak. Solusi ini bertujuan tidak hanya untuk mengenalkan fauna tersebut, tetapi juga untuk mendukung pelestarian budaya Dayak dengan pendekatan berbasis kecerdasan buatan.

b. Pengumpulan *Dataset*

Penulis menggunakan *dataset* yang diperoleh dari dua sumber utama, yaitu sumber publik dan pengumpulan mandiri. *Dataset* publik diambil dari website *online* yang menyediakan gambar fauna khas Dayak, sementara data mandiri dikumpulkan melalui pengambilan gambar *web scraping*. Kombinasi kedua metode ini bertujuan untuk memastikan bahwa *dataset* yang digunakan memiliki variasi yang cukup dalam hal sudut pandang, pencahayaan, dan latar belakang, sehingga model yang dikembangkan dapat memiliki kemampuan generalisasi yang baik.

c. *Preprocessing*

Setelah dilakukan pengumpulan *dataset* pada tahap *ini*, penulis melakukan beberapa langkah untuk mempersiapkan data sebelum pelatihan model. Terdapat tiga proses utama yaitu *resizing*, normalisasi, dan augmentasi yang dapat dilihat pada Gambar 3.2.



Gambar 3. 2 *Preprocessing*

1) *Resizing*

Semua gambar dalam *dataset* diubah ukurannya menjadi 299x299 piksel. Dimensi ini dipilih karena sesuai dengan kebutuhan input model *InceptionV3*, yang dirancang untuk menerima gambar dengan ukuran tersebut. Dengan melakukan perubahan ukuran, seluruh gambar menjadi konsisten dalam dimensi, sehingga dapat mengurangi masalah teknis yang mungkin muncul selama proses pelatihan model.

2) Normalisasi

Nilai piksel gambar dinormalisasi agar berada dalam rentang [0, 1] dengan membagi setiap nilai piksel gambar (0–255) dengan 255. Proses normalisasi ini bertujuan untuk mempercepat konvergensi model selama pelatihan dan mencegah masalah numerik, seperti gradien yang meledak atau menghilang. Dengan data yang telah dinormalisasi, model dapat lebih mudah memahami dan mempelajari pola-pola visual dalam gambar.

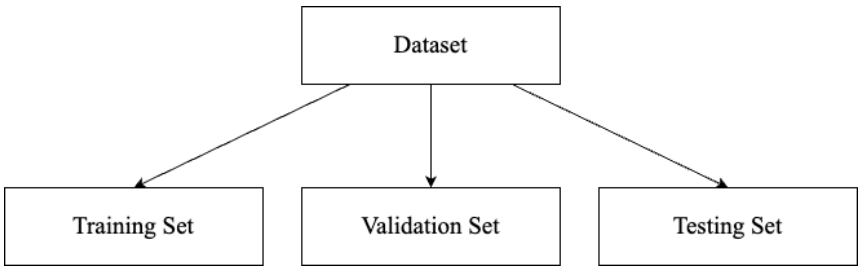
3) Augmentasi

Penulis menerapkan augmentasi data untuk meningkatkan variasi *dataset* tanpa menambah jumlah data mentah. Teknik augmentasi yang digunakan meliputi:

a) Rotasi: mengubah sudut gambar untuk menangkap variasi orientasi objek.
b) Pergeseran posisi (*width shift* dan *height shift*): menggeser posisi objek dalam gambar secara horizontal atau vertikal.
c) *Flipping horizontal*: membalik gambar secara horizontal untuk menciptakan variasi cermin dari objek.
d) *Shear Transformation*: menerapkan distorsi miring untuk menciptakan variasi bentuk.
e) *Zoom*: Memperbesar atau memperkecil objek dalam gambar.
f) *Fill Mode*: Mengisi area kosong yang muncul akibat augmentasi dengan nilai piksel terdekat.

d. Pembagian *Dataset*

Setelah dilakukan *preprocessing*, *dataset* dibagi menjadi tiga bagian utama, yaitu *training set* (80%), *validation set* (10%), dan *testing set* (10%). Pembagian ini dilakukan untuk memastikan bahwa model dapat belajar, divalidasi, dan diuji dengan data yang berbeda, guna menghindari *overfitting*. Pembagian ini dilakukan secara merata, memastikan distribusi kelas dalam setiap *set* tetap seimbang dan mewakili seluruh data. Pembagian *dataset* dapat dilihat pada Gambar 3.3.



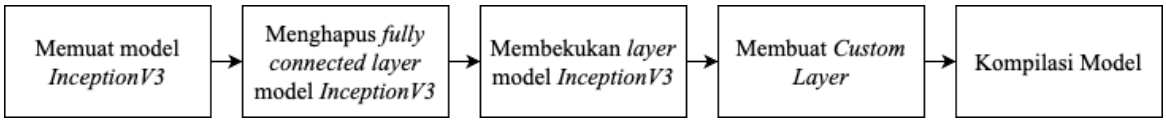
Gambar 3. 3 Pembagian *Dataset*

1) *Training* set: *dataset* ini digunakan untuk melatih model. Gambar-gambar dalam *training set* membantu model mengenali pola-pola visual, seperti bentuk tubuh, warna, atau tekstur, yang khas dari setiap fauna Dayak.
2) *Validation set*: digunakan untuk memantau performa model selama proses pelatihan. Data ini tidak dilibatkan dalam pelatihan, sehingga memberikan gambaran objektif mengenai kemampuan model dalam mengenali pola pada data baru.
3) *Testing set*: digunakan untuk mengevaluasi kinerja akhir model. Data ini hanya digunakan setelah proses pelatihan selesai, untuk memastikan bahwa model dapat mengenali pola dengan akurat pada data yang belum pernah dilihat sebelumnya.

e. Implementasi Model *InceptionV3*

Pada tahap ini, penulis menerapkan model *InceptionV3* sebagai dasar untuk membangun arsitektur *Convolutional Neural Network* (CNN). *InceptionV3* adalah salah satu model yang telah dilatih sebelumnya dan populer, karena telah dilatih pada *dataset* besar seperti *ImageNet*, sehingga

memiliki kemampuan luar biasa dalam mengenali pola visual pada gambar. Implementasi model *InceptionV3* dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Implementasi Model *InceptionV3*

1) Memuat model *inceptionV3*

Langkah pertama adalah memuat arsitektur model *InceptionV3* yang telah dilatih sebelumnya (*pretrained model*) dengan menggunakan bobot dari dataset *ImageNet*. Bobot *pretrained* ini digunakan sebagai dasar bagi model untuk mengenali pola-pola visual dasar seperti tepi, tekstur, dan bentuk. Pada tahap ini, penulis juga menetapkan dimensi input model sebesar 299x299 piksel dengan 3 saluran warna (RGB), sesuai dengan kebutuhan model *InceptionV3* untuk memastikan konsistensi pemrosesan data.

2) Menghapus *fully connected layer* model *InceptionV3*

Setelah model dimuat, penulis menghapus *fully connected layer* bawaan dari model *InceptionV3*. Lapisan ini awalnya dirancang untuk klasifikasi pada *dataset* besar seperti *ImageNet*, yang memiliki lebih dari 1.000 kelas, sehingga tidak sesuai untuk tugas klasifikasi fauna khas Dayak yang hanya melibatkan beberapa kelas spesifik. Penghapusan lapisan ini memungkinkan penyesuaian arsitektur model. Sebagai gantinya, penulis menambahkan lapisan khusus (*custom layers*) yang dirancang sesuai dengan kebutuhan klasifikasi pada *dataset* fauna khas Dayak.

3) Membekukan *layer* model *InceptionV3*

Pada tahap selanjutnya, penulis membekukan (*freeze*) *layer-layer* awal pada model *InceptionV3* untuk memastikan bobot pada *layer-layer pretrained* tersebut tidak mengalami perubahan selama proses pelatihan. *Layer-layer* awal ini telah dilatih untuk mengenali fitur-fitur dasar seperti garis, sudut, dan pola tekstur, yang bersifat universal dan tidak bergantung pada *dataset* tertentu. Pembekuan ini dilakukan untuk mempercepat waktu pelatihan, karena hanya lapisan khusus (*custom layers*) yang akan dilatih, serta menjaga stabilitas model dengan mempertahankan fitur dasar yang sudah optimal.

4) Kompilasi model

Setelah arsitektur model selesai disusun, langkah berikutnya adalah mengompilasi model dengan parameter yang dirancang khusus untuk tugas klasifikasi. Proses kompilasi ini bertujuan untuk mengintegrasikan arsitektur model dengan fungsi *loss*, *optimizer*, dan metrik evaluasi yang akan digunakan selama pelatihan. Beberapa parameter yang digunakan antara lain:

a) *Optimizer*

Penulis memilih Adam dengan *learning rate* sebesar karena kemampuannya menyesuaikan *learning rate* secara adaptif, sehingga pelatihan menjadi lebih stabil.

b) *Loss Function*

Categorical crossentropy digunakan sebagai fungsi *loss* karena sesuai untuk tugas klasifikasi multikelas, seperti mengenali berbagai jenis fauna khas Dayak.

c) Metrik Evaluasi

Accuracy dipilih sebagai metrik utama untuk memantau kinerja model selama proses pelatihan dan validasi.

f. *Training Model*

Setelah proses implementasi model *InceptionV3* dan model dikompilasi, pelatihan dilakukan menggunakan *training set*. Penulis menerapkan *early stopping* untuk menghentikan pelatihan jika nilai *loss* pada *validation set* tidak mengalami perbaikan dalam 10 *epoch* berturut-turut, dengan tujuan untuk mencegah *overfitting* dan menghemat waktu pelatihan. Selama proses pelatihan, model belajar untuk mengenali pola visual khas dari setiap fauna Dayak. *Validation set* digunakan untuk memantau kinerja model, sehingga penulis dapat mengevaluasi apakah model hanya "menghafal" data *training* atau benar-benar mampu mengidentifikasi pola secara umum.

g. Evaluasi Model

Setelah tahap *training* selesai dijalankan maka tahap evaluasi dilakukan untuk mengukur kinerja akhir model menggunakan *testing set*. Penulis menggunakan metrik seperti akurasi, presisi, *recall*, dan *F1-Score* untuk menilai hasil model.

1) Jika akurasi model lebih dari 90%, maka model dianggap sangat baik dan siap digunakan dalam aplikasi praktis, seperti pendidikan atau pelestarian budaya.
2) Jika akurasi berada di antara 80% hingga 90%, model dianggap cukup baik, tetapi masih memerlukan perbaikan, seperti menambah data atau memperbaiki tahap *preprocessing*.
3) Jika akurasi di bawah 80%, penulis akan meninjau kembali langkah-langkah sebelumnya, termasuk pemilihan parameter model, variasi data, atau strategi augmentasi.

Evaluasi ini memberikan gambaran yang jelas tentang seberapa baik model dalam mengenali fauna khas Dayak dan memastikan model mampu menghasilkan hasil yang akurat dalam implementasinya.

3.4 Instrumen Penelitian

Studi pustaka untuk mendukung landasan teori penelitian, literatur seperti jurnal ilmiah, buku, dan artikel digunakan dalam penelitian ini. Hal ini dilakukan untuk memastikan bahwa metode dan pendekatan yang digunakan memiliki dasar ilmiah yang kuat dan sesuai dengan standar terbaik dalam penelitian sejenis. Penelitian ini juga mencakup pemahaman mengenai teknik CNN dan *transfer learning*, metode *preprocessing* citra, dan metode evaluasi model yang digunakan, seperti *F1 score*.

3.5 Analisis Kebutuhan

Pada penelitan ini diperlukan spesifikasi komputer dan beberapa perangkat lunak yang diperlukan untuk membangun model CNN ini.

Berikut adalah persyaratan minimal komputer dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	*Processor*	*Intel Core* i5 / AMD *Ryzen* 5 (atau lebih tinggi)
2	GPU	NVIDIA GTX 1060 atau lebih tinggi
3	RAM	8 GB (16 GB disarankan atau lebih tinggi)
4	Penyimpanan	256 GB atau lebih tinggi

Berikut ini adalah beberapa perangkat lunak yang dibutuhkan dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Perangkat Lunak

No	Nama	Keterangan
1	*Google Chrome*	*Browser*
2	*Google Colab Pro*	Lingkungan Pengembangan
3	*Python*	Bahasa Pemrograman
4	*TensorFlow*	*Framework*
5	*Keras*	*Library*
6	*Numpy*	*Library*
7	*Matplotlib*	*Library*

3.5.1 Analisis Data

Penelitian ini melibatkan kumpulan data yang beragam, yang dikumpulkan melalui sumber-sumber publik dan pengumpulan sendiri, serta diperkaya dengan teknik augmentasi untuk menambah variasi data. Data yang digunakan meliputi gambar fauna khas Kalimantan dalam format RGB untuk menjaga informasi warna yang penting bagi model klasifikasi *Convolutional Neural Network* (CNN). *Dataset* ini berfokus pada lima spesies fauna ikonik dari Kalimantan Tengah, yaitu:

1) Kahiu
2) Burung Tingang
3) Bakara
4) Kalasi
5) Kalawet

Penentuan spesies-spesies tersebut didasari oleh keunikan dan pentingnya spesies tersebut dalam budaya dan ekologi Kalimantan. *Dataset* yang digunakan memiliki rincian sebagai berikut:

a) Jumlah data pelatihan: 3,618 gambar
b) Jumlah data validasi: 450 gambar
c) Jumlah data pengujian: 458 gambar

Berikut merupakan distribusi jumlah data pada masing-masing kelas dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Distribusi Jumlah Data

Kelas	Pelatihan	Validasi	Pengujian	Total
Kahiu	1,093	136	138	1,367
Burung Tingang	1,186	148	149	1,483
Bakara	1,036	129	131	1,296
Kalasi	170	21	22	213
Kalawet	127	16	18	161

Setiap kelas memiliki karakteristik visual yang khas, meliputi warna, pola, dan bentuk tubuh tertentu, yang berperan penting dalam klasifikasi menggunakan CNN. Secara keseluruhan, *dataset* menunjukkan ketimpangan jumlah data antar kelas, dengan spesies seperti Kahiu, Burung Tingang, dan Bakara memiliki data yang jauh lebih banyak dibandingkan dengan Kalasi dan Kalawet. Untuk mengatasi ketimpangan ini, langkah augmentasi data dilakukan secara menyeluruh.

Berikut ini adalah tahapan utama dalam proses pengumpulan dan analisis data.

a. Pengambilan dari *dataset* publik

Dataset publik yang diperoleh dari *platform* seperti *Kaggle*, *Image.CV*, dan *Roboflow* menyediakan koleksi gambar fauna yang telah dikurasi oleh komunitas. *Dataset* ini digunakan sebagai bagian utama dari *dataset* pelatihan karena memiliki kualitas yang tinggi dan keragaman yang signifikan dalam membantu meningkatkan kinerja model. Seluruh data publik ini disimpan dalam format RGB untuk menjaga kualitas visual yang diperlukan untuk klasifikasi.

b. *Web Scraping*

Metode *web scraping* digunakan untuk mengumpulkan data tambahan, yang meliputi gambar dan informasi tentang fauna Dayak. Alat-alat seperti *Beautiful Soup* atau *Selenium* dapat digunakan untuk mengumpulkan data dari berbagai situs *web* yang relevan.

c. Pelabelan Data

Semua gambar yang dikumpulkan secara *public*, mandiri, dan juga *webscraping* telah melalui proses pelabelan yang cermat. Setiap gambar dikategorikan berdasarkan spesies, seperti Kahiu, Burung Tingang, Bakara, Kalasi, dan Kalawet, untuk memastikan model dapat secara akurat mengidentifikasi setiap spesies. Pelabelan dilakukan secara manual atau semi-otomatis untuk mendapatkan akurasi yang tinggi, sehingga model CNN dapat mengenali setiap kategori dengan benar.

d. Augmentasi Data

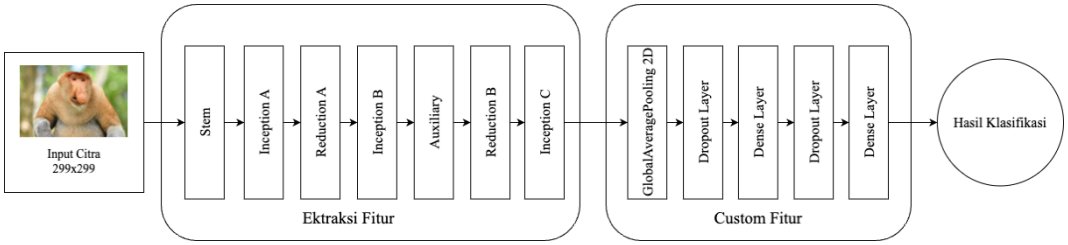
Guna menambah variasi dalam *dataset* dan memperkuat kemampuan generalisasi model, augmentasi data dilakukan pada gambar RGB. Teknik augmentasi mencakup transformasi seperti rotasi, *flipping*, *zoom*, dan penyesuaian kontras. Beberapa teknik ini tidak hanya secara efektif dapat meningkatkan jumlah sampel data, tetapi juga meningkatkan kemampuan model untuk mengidentifikasi spesies dari variasi gambar yang berbeda tanpa mengubah karakteristik visual utama dari spesies tersebut. Augmentasi data ini juga membantu model menghindari *overfitting* dan meningkatkan akurasi pada data uji.

3.5.2 Analisis Proses

a. Arsitektur CNN dengan *InceptionV3*

Berdasarkan arsitektur CNN dengan *transfer learning* menggunakan model *InceptionV3* yang dapat dilihat pada Gambar 2.7, penulis melakukan beberapa penyesuaian untuk tugas klasifikasi pada *dataset* baru. Pertama, lapisan *fully connected* asli dihapus karena hanya membutuhkan sejumlah kelas tertentu (5 kelas). Kedua, lapisan yang telah dilatih sebelumnya dibekukan untuk mempertahankan fitur umum dari *dataset ImageNet*, seperti pola deteksi tepi,

tekstur, dan bentuk. Terakhir, penulis menambahkan lapisan *custom* di atas model *InceptionV3* untuk memungkinkan klasifikasi pada *dataset* baru.



Gambar 3. 5 Arsitektur CNN *InceptionV3*

Seperti yang terlihat pada Gambar 3.5. Proses ini melibatkan beberapa langkah dan komponen arsitektur model, berikut adalah penjelasan pada *custom* fitur:

a) *GlobalAveragePooling2D*

Setelah lapisan *convolutional* terakhir pada model *InceptionV3* dipangkas atau dihilangkan, penulis menambahkan lapisan *pooling* berupa *GlobalAveragePooling2D*. Lapisan ini berfungsi untuk menghitung rata-rata nilai dari setiap *feature map* pada lapisan *convolutional* terakhir. Setiap *feature map* berbentuk matriks 2D diringkas menjadi satu nilai rata-rata untuk setiap *channel*. Proses ini menghasilkan representasi data yang lebih sederhana namun tetap mempertahankan informasi penting. Dengan metode *pooling* rata-rata, lapisan ini menjaga informasi utama dari fitur-fitur yang telah diekstraksi oleh model *pretrained* sekaligus mengurangi dimensi data. Selain itu, *GlobalAveragePooling2D* mempermudah integrasi antara lapisan

convolutional pretrained model dengan lapisan *dense* yang berada setelahnya.

b) *Dropout Layer* Pertama

Setelah lapisan *GlobalAveragePooling2D*, ditambahkan lapisan *Dropout* pertama dengan *dropout rate* sebesar 0.5. *Dropout* berfungsi untuk secara acak "menonaktifkan" 50% *neuron* selama proses pelatihan. Hal ini memberikan efek regularisasi pada model, sehingga membantu mencegah *overfitting*, yaitu kondisi di mana model terlalu terfokus pada data pelatihan dan kesulitan menggeneralisasi pada data baru. *Dropout* pertama ini berperan dalam memastikan bahwa representasi fitur yang telah diringkas oleh *pooling* layer tetap kokoh dan tidak mudah terpengaruh oleh *noise* atau ketergantungan pada pola *neuron* tertentu.

c) *Dense Layer*

Setelah Dropout *Layer* pertama, ditambahkan *Dense Layer* dengan 256 *neuron* yang menggunakan fungsi aktivasi ReLU (*Rectified Linear Unit*). Lapisan *dense* ini berfungsi untuk mempelajari hubungan *non-linear* dalam data yang telah diringkas oleh model *pretrained* sebelumnya. Fungsi aktivasi ReLU dipilih karena kesederhanaannya dan kemampuannya yang efektif dalam menangani hubungan *non-linear*, dengan mempertahankan nilai positif dan mengabaikan nilai negatif. Lapisan *dense* ini menjadi elemen penting dalam menyesuaikan fitur-fitur umum dari model *pretrained InceptionV3* agar sesuai dengan pola-pola spesifik dalam dataset baru, yakni klasifikasi citra fauna khas Dayak.

d) *Dropout Layer* Kedua

Setelah *Dense Layer*, ditambahkan *Dropout Layer* kedua dengan *dropout rate* sebesar 0.5. Seperti pada *Dropout* pertama, tujuan utama lapisan ini adalah mencegah *overfitting*. Namun, pada tahap ini, regularisasi difokuskan pada *neuron-neuron* hasil *Dense Layer*, yang memiliki jumlah parameter lebih besar. Dengan secara acak menonaktifkan 50% dari total *neuron* selama proses pelatihan, lapisan ini membantu memastikan model tidak terlalu bergantung pada *neuron* tertentu dalam *Dense Layer*, sehingga meningkatkan kemampuan model untuk membuat prediksi yang lebih general.

e) *Output Layer*

Lapisan terakhir dalam model adalah *Output Layer* yang terdiri dari 5 *neuron*, dengan fungsi aktivasi *Softmax*. Setelah data melewati lapisan-lapisan sebelumnya, yaitu *GlobalAveragePooling2D*, *Dropout Layer* pertama, *Dense Layer*, dan *Dropout* Layer kedua, model telah memiliki representasi fitur yang cukup kuat dan umum. *Output Layer* ini berfungsi untuk menghasilkan *output* berupa probabilitas untuk setiap kelas dalam *dataset*. Fungsi aktivasi *Softmax* memastikan bahwa *output* setiap *neuron* berada dalam rentang 0 hingga 1, dengan total probabilitas seluruh kelas yang bernilai 1. Hal ini memungkinkan model untuk memberikan prediksi yang lebih akurat. Kelas dengan probabilitas tertinggi akan dipilih sebagai hasil akhir dari klasifikasi

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Implementasi

Pada tahap ini, dilakukan pengujian terhadap empat model dengan berbagai konfigurasi *hyperparameter* dan penerapan augmentasi data untuk memperoleh hasil pelatihan yang optimal. Setiap model dirancang dan diimplementasikan untuk menganalisis dampak perubahan konfigurasi terhadap akurasi pada proses pelatihan, validasi, dan pengujian.

a. Konfigurasi Model dan Hasil Pelatihan

Langkah-langkah implementasi model meliputi pengaturan *hyperparameter*, penggunaan augmentasi data, dan proses pelatihan model. Penjelasan setiap langkah beserta hasilnya dirangkum dalam tabel dan uraian berikut.

1. Konfigurasi *Hyperparameter*

   Untuk memulai implementasi, dilakukan pengujian terhadap berbagai konfigurasi hyperparameter yang berperan penting dalam pelatihan model. *Hyperparameter* seperti *batch size*, *learning rate*, regularisasi L2, dan jumlah *epoch* diatur dengan tujuan menemukan keseimbangan optimal antara akurasi pelatihan dan validasi.

Tabel 4. 1 *Hyperparameter* yang di-*tunning*

Model	***Hyperparameter***				
	Batch Size		***Learning Rate***	**L2**	***Epoch***
	Training	***Validation***			
1	32	16	0,001		20
2	32	16	0,001	0,01	20
3	32	16	0,001		20
4	64	32	0,0001		50

Dapat dilihat pada Tabel 4.1 menunjukkan konfigurasi *hyperparameter* yang diterapkan pada setiap model. Pengaturan ini dirancang untuk mengeksplorasi kombinasi terbaik yang dapat meningkatkan kinerja model, baik selama pelatihan maupun evaluasi.

2. Teknik Augmentasi Data

Selanjutnya, augmentasi data diterapkan untuk memperkaya variasi dataset dan meningkatkan kemampuan generalisasi model. Teknik augmentasi yang diterapkan pada setiap model tetap sama, kecuali pada bagian zoom dan rotasi yang berbeda, untuk mencegah *overfitting* dan membuat model lebih responsif terhadap berbagai pola.

Tabel 4. 2 Augmentasi data yang diterapkan

Model	**Augmentasi Data**	
	Zoom	**Rotasi (Derajat)**
1	30%	60
2	30%	40
3	30%	30
4	20%	30

Dapat dilihat Tabel 4.2 menggambarkan strategi augmentasi yang diterapkan pada setiap model. Teknik augmentasi ini disesuaikan untuk mencapai keseimbangan antara variasi data dan kompleksitas pelatihan.

3. Hasil Akurasi Model

Hasil pelatihan dari keempat model diukur berdasarkan akurasi data pelatihan, validasi, dan pengujian.

Tabel 4. 3 Akurasi dari semua model

Model	***Accuracy***		
	Training	***Validation***	***Testing***
1	98,2%	99,5%	99,3%
2	96,9%	98,4%	98,6%
3	98,4%	99,1%	99,3%
4	98,1%	99,3%	99,5%

Dapat dilihat pada Tabel 4.3 menyajikan hasil performa masing-masing model. Model keempat mencatatkan akurasi tertinggi pada data pengujian, yaitu 99,5%, dengan perbedaan akurasi yang sangat kecil antara pelatihan, validasi, dan pengujian.

b. Analisis Performa model

Analisis berikut mengevaluasi performa setiap model berdasarkan konfigurasi *hyperparameter* dan teknik augmentasi data yang diterapkan. Penjelasan disusun secara rinci untuk masing-masing model agar lebih mudah dipahami.

1. Model Pertama

Model pertama dilatih dengan pengaturan dasar tanpa regularisasi L2, menggunakan *batch size* 32 dan *learning rate* 0,001 selama 20 *epoch*. Augmentasi data yang diterapkan meliputi *zoom* hingga 30% dan rotasi sebesar 60°. Berikut ini adalah pemaparan hasil yang diperoleh:

a) Akurasi Pelatihan: 98,2%

b) Akurasi Validasi: 99,5%

c) Akurasi Pengujian: 99,3%

Hasil yang diperoleh menunjukkan akurasi validasi yang tinggi. Namun, terdapat perbedaan yang cukup besar antara akurasi pelatihan dan validasi, yang mengindikasikan adanya potensi *overfitting*. Hal ini menunjukkan bahwa model mungkin terlalu menyesuaikan diri dengan data pelatihan, sehingga kurang mampu melakukan generalisasi pada data yang belum pernah dilihat sebelumnya.

2. Model Kedua

Pada model kedua, diterapkan regularisasi L2 sebesar 0,01 untuk mengurangi kemungkinan terjadinya overfitting. Batch size tetap 32, learning rate 0,001, dengan augmentasi data yang mencakup zoom 30% dan rotasi 40°. Berikut adalah pemaparan hasil yang diperoleh:

a) Akurasi Pelatihan: 96,9%

b) Akurasi Validasi: 98,4%

c) Akurasi Pelatihan: 98,6%

Penerapan regularisasi L2 berhasil mengurangi *overfitting*, yang terlihat dari perbedaan akurasi yang lebih kecil antara pelatihan, validasi, dan pengujian. Namun, terdapat sedikit penurunan akurasi pelatihan karena pembatasan fleksibilitas model.

3. Model Ketiga

Model ini dilatih tanpa menggunakan regularisasi L2, dengan batch size 32 dan learning rate 0,001 selama 20 epoch. Teknik augmentasi yang

diterapkan mencakup zoom hingga 30% dan rotasi 30°. Berikut ini adalah pemaparan hasil yang diperoleh:

a) Akurasi Pelatihan: 98,4%

b) Akurasi Validasi: 99,1%

c) Akurasi Pelatihan: 99,3%

Akurasi pelatihan meningkat dibandingkan model kedua, namun generalisasi sedikit menurun. Selisih akurasi antara pelatihan dan validasi menunjukkan potensi overfitting yang masih ada.

4. Model Keempat

Model keempat dilatih dengan batch size 64, learning rate 0,0001, dan teknik augmentasi yang meliputi zoom 20% serta rotasi 30°. Jumlah epoch diperpanjang menjadi 50 untuk meningkatkan stabilitas pelatihan. Berikut ini adalah pemaparan hasil yang diperoleh:

a) Akurasi Pelatihan: 98,1%

b) Akurasi Validasi: 99,3%

c) Akurasi Pelatihan: 99,5%

Model keempat menunjukkan performa terbaik dengan selisih akurasi yang sangat kecil antara pelatihan, validasi, dan pengujian. Hal ini menandakan kemampuan generalisasi yang sangat baik. Konfigurasi ini dianggap optimal untuk tahap pengujian selanjutnya.

c. Evaluasi Kinerja Model Berdasarkan Grafik

Pada bagian ini, grafik yang menunjukkan hasil pelatihan dan validasi model dimanfaatkan untuk menilai kinerja dari keempat model yang telah dikembangkan. Grafik-grafik tersebut meliputi loss dan akurasi selama proses pelatihan.

1. Model Pertama

Model pertama ini menunjukkan pola pelatihan yang stabil namun masih menghadapi tantangan dalam generalisasi data baru.



Gambar 4. 1 Grafik Model Pertama

Pada Gambar 4.1 pada bagian *train loss* dan *validation loss*, terlihat bahwa *loss* pada data pelatihan terus menurun seiring bertambahnya jumlah *epoch*. Namun, *loss* pada data validasi cenderung stabil lebih awal dan nilainya lebih rendah dibandingkan *loss* pelatihan. Hal ini menunjukkan bahwa model dapat mempelajari pola dari data pelatihan dengan baik, meskipun ada indikasi potensi *overfitting* akibat perbedaan *loss* yang terlihat.

Sementara itu pada bagian *train accuracy* dan *validation accuracy*, untuk akurasi data pelatihan meningkat secara bertahap hingga mencapai sekitar 98%, sedangkan akurasi validasi lebih tinggi, hampir menyentuh 99,5%.

Perbedaan akurasi yang cukup signifikan ini juga mengindikasikan adanya tanda-tanda *overfitting*. Meskipun model pertama menunjukkan kinerja yang baik pada akurasi validasi dan pengujian, terdapat peluang untuk meningkatkan stabilitas pelatihan dan kemampuan generalisasi model.

2. Model Kedua

Model kedua memanfaatkan regularisasi L2 untuk meningkatkan stabilitas generalisasi pada data validasi.



Gambar 4. 2 Grafik Model Ketiga

Pada Gambar 4.2 bagian train loss dan validation loss, terlihat bahwa regularisasi L2 sebesar 0.01 berhasil menekan nilai loss untuk data validasi, yang cenderung lebih rendah dibandingkan data pelatihan. Selain itu, loss pelatihan cenderung lebih stabil dibandingkan model pertama, meskipun nilai loss validasi menunjukkan sedikit fluktuasi. Hal ini menunjukkan bahwa regularisasi L2 membantu meningkatkan generalisasi model dan mencegah overfitting secara signifikan.

Sementara itu pada bagian train accuracy dan validation accuracy, terlihat bahwa akurasi validasi mengalami fluktuasi yang cukup besar pada

beberapa epoch awal, meskipun rata-rata tetap berada pada kisaran 98%. Fluktuasi ini kemungkinan terjadi karena kombinasi regularisasi L2 dan augmentasi data yang agresif (rotasi hingga 40°). Akurasi pelatihan juga sedikit menurun dibandingkan model pertama, yang merupakan efek dari regularisasi L2. Model ini menunjukkan upaya untuk mengurangi *overfitting*, tetapi fluktuasi pada akurasi validasi menunjukkan bahwa model memerlukan stabilitas lebih baik dalam proses pelatihannya.

3. Model Ketiga

Model ketiga tidak menggunakan regularisasi L2, sehingga menunjukkan performa yang lebih variatif.



Gambar 4. 3 Grafik Model Keempat

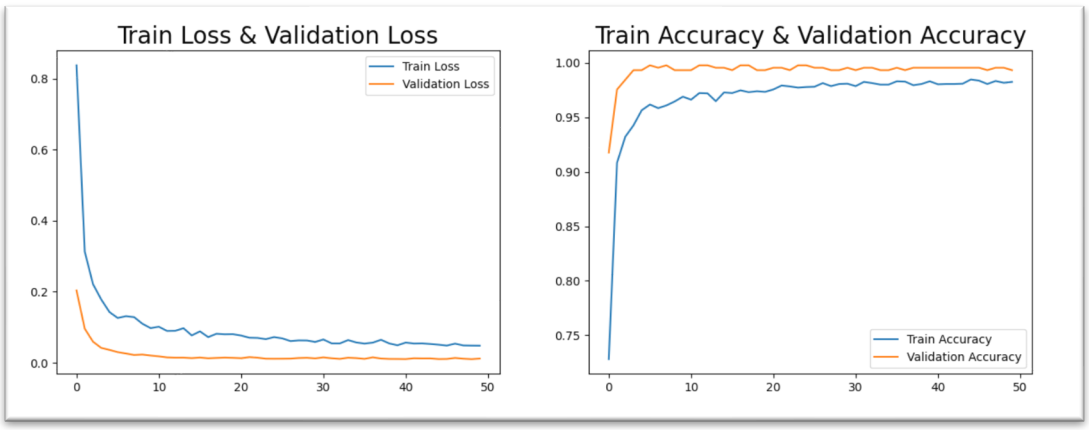
Pada Gambar 4.3 bagian *train loss* dan *validation loss*, terlihat bahwa perbedaan antara *loss* pelatihan dan validasi menjadi lebih kecil dibandingkan model sebelumnya. *Loss* pelatihan terus menurun dengan stabil hingga sekitar *epoch* ke-10, sedangkan *loss* validasi relatif stabil, mencerminkan pembelajaran yang lebih konsisten. Meskipun begitu, *loss* validasi yang

sedikit lebih rendah daripada *loss* pelatihan dapat menjadi indikasi adanya *overfitting* dalam skala kecil.

Sementara itu pada bagian *train accuracy* dan *validation accuracy*, akurasi validasi tetap tinggi, berada di atas 99%, sementara akurasi pelatihan perlahan meningkat hingga mencapai sekitar 98,4%. Perbedaan yang lebih kecil antara akurasi pelatihan dan validasi dibandingkan model sebelumnya menunjukkan perbaikan dalam kemampuan generalisasi model. Dengan penghapusan regularisasi L2, model memiliki fleksibilitas lebih besar untuk mempelajari pola dari data pelatihan. Namun, meskipun perbedaan antara akurasi pelatihan dan validasi telah berkurang, potensi *overfitting* masih menjadi perhatian yang perlu diatasi lebih lanjut.

4. Model Keempat

Model keempat menggunakan kombinasi *batch size* besar dan *learning rate* rendah untuk mencapai performa terbaik.



Gambar 4. 4 Grafik Model Keempat

Pada Gambar 4. 4 bagian *train loss* dan *validation loss*, peningkatan *batch size* menjadi 64 menghasilkan penurunan *loss* yang lebih stabil dan teratur. Dengan *learning rate* yang lebih rendah (0,0001) dan jumlah *epoch* yang diperpanjang hingga 50, model memiliki cukup waktu untuk belajar secara bertahap. Di akhir pelatihan, *loss* pada data pelatihan dan validasi hampir sejajar, menunjukkan bahwa model dapat belajar tanpa mengalami *overfitting*.

Pada bagian *train accuracy* dan *validation accuracy*, akurasi pelatihan dan validasi mendekati angka 99% pada akhir pelatihan dan hampir identik. Hal ini menandakan bahwa model keempat memiliki kemampuan generalisasi terbaik dibandingkan model lainnya. Perbedaan yang sangat kecil antara akurasi pelatihan dan validasi, ditambah dengan tren yang stabil, merupakan hasil dari konfigurasi pelatihan yang dirancang dengan lebih optimal.

d. Evaluasi Model Berdasarkan *Classification Report*

Ditahap ini menganalisis *classification report* untuk menilai kinerja model pada setiap kelas. Proses ini mencakup penghitungan *precision*, *recall*, dan *F1-Score* yang digunakan untuk memahami performa model dari keempat dalam mengidentifikasi citra dengan benar.

1. Model Pertama

```
Classification Report:
                precision    recall  f1-score   support

        Bakara       0.99      1.00      1.00       131
Burung Tingang       1.00      0.99      1.00       149
         Kahiu       0.99      1.00      0.99       138
        Kalasi       1.00      0.91      0.95        22
       Kalawet       1.00      1.00      1.00        18

      accuracy                           0.99       458
     macro avg       1.00      0.98      0.99       458
  weighted avg       0.99      0.99      0.99       458
```

Gambar 4. 5 *Classification Report* Model Pertama

Pada Gambar 4.5, nilai *precision*, *recall*, dan *F1-score* menunjukkan hasil yang sangat tinggi di hampir semua kelas, dengan rata-rata *weighted* mencapai 99%. Namun, analisis lebih mendalam menunjukkan bahwa *recall* untuk kelas Kalasi hanya sebesar 0,91, yang lebih rendah dibandingkan kelas lainnya. Hal ini mengindikasikan bahwa model menghadapi kesulitan dalam mendeteksi semua sampel dari kelas tersebut.

Masalah ini kemungkinan disebabkan oleh ketidakseimbangan dalam *dataset*, di mana kelas Kalasi hanya memiliki 22 sampel, menjadikannya kelas dengan jumlah data paling sedikit. Meskipun akurasi keseluruhan cukup tinggi, yaitu 99%, perbedaan performa antar kelas menunjukkan bahwa model masih memerlukan peningkatan untuk mengatasi ketidakseimbangan ini dan meningkatkan generalisasi, terutama pada kelas minoritas.

2. Model Kedua

```
Classification Report:
                precision    recall  f1-score   support

        Bakara       0.99      0.99      0.99       131
Burung Tingang       1.00      1.00      1.00       149
        Kahiu        0.97      1.00      0.98       138
       Kalasi        1.00      0.86      0.93        22
      Kalawet        1.00      0.89      0.94        18

     accuracy                           0.99       458
    macro avg       0.99      0.95      0.97       458
 weighted avg       0.99      0.99      0.99       458
```

Gambar 4. 6 *Classification Report* Model Kedua

Pada Gambar 4.6, terdapat penurunan performa pada beberapa metrik. *Precision* dan *recall* untuk kelas Kalasi masing-masing turun menjadi 1,00 dan 0,86, dengan *F1-score* menurun menjadi 0,93. Selain itu, kelas Kalawet mencatat nilai *recall* sebesar 0,89. Meskipun *precision* untuk kedua kelas ini tetap tinggi, penurunan pada *recall* menunjukkan bahwa model lebih sering gagal mendeteksi sampel dari kedua kelas ini.

Secara keseluruhan, akurasi tetap stabil di angka 99%. Namun, nilai *macro average recall* turun menjadi 0,95, yang menandakan bahwa model kurang optimal dalam mengenali kelas minoritas dibandingkan model sebelumnya. Penurunan ini kemungkinan berkaitan dengan fluktuasi selama pelatihan, sebagaimana terlihat pada grafik sebelumnya, di mana model kesulitan mencapai stabilitas.

3. Model Ketiga

```
Classification Report:
                precision    recall  f1-score   support

        Bakara       1.00      0.99      1.00       131
Burung Tingang       1.00      1.00      1.00       149
         Kahiu       0.99      1.00      0.99       138
        Kalasi       0.95      0.95      0.95        22
       Kalawet       1.00      0.94      0.97        18

      accuracy                           0.99       458
     macro avg       0.99      0.98      0.98       458
  weighted avg       0.99      0.99      0.99       458
```

Gambar 4. 7 *Classification Report* Model Ketiga

Pada Gambar 4.7 menunjukkan peningkatan yang signifikan dibandingkan model kedua. *Recall* untuk kelas Kalasi naik menjadi 0,95, yang setara dengan nilai *precision*-nya. Kelas Kalawet juga mengalami peningkatan, dengan *recall* mencapai 0,94 dan *F1-score* sebesar 0,97. Hal ini menunjukkan bahwa model memiliki kemampuan yang lebih baik dalam mendeteksi semua kelas secara lebih seimbang, meskipun performa pada kelas dengan jumlah sampel kecil masih sedikit tertinggal dibandingkan kelas lainnya.

Macro average precision, *recall*, dan *F1-score* masing-masing tercatat di angka 0,99, 0,98, dan 0,98. Ini menunjukkan bahwa model lebih efektif dalam menangani ketidakseimbangan data. Namun, meskipun performa model ini sudah cukup baik, masih terdapat ruang untuk perbaikan, terutama

dalam meningkatkan deteksi pada kelas minoritas jika dibandingkan dengan model berikutnya.

4. Model Keempat

Classification Report:

	precision	recall	f1-score	support
Bakara	0.99	1.00	1.00	131
Burung Tingang	1.00	1.00	1.00	149
Kahiu	0.99	0.99	0.99	138
Kalasi	1.00	0.95	0.98	22
Kalawet	1.00	1.00	1.00	18
accuracy			1.00	458
macro avg	1.00	0.99	0.99	458
weighted avg	1.00	1.00	1.00	458

Gambar 4. 8 *Clasisification Report* Model Keempat

Pada Gambar 4.8 model memberikan performa terbaik di antara semua model, dengan *precision*, *recall*, dan *F1-score* sempurna (1.00) untuk hampir semua kelas. Hal ini menunjukkan bahwa model dapat mendeteksi seluruh sampel dari semua kelas tanpa kesalahan, termasuk kelas minoritas seperti Kalasi dan Kalawet.

Akurasi keseluruhan mencapai 100%, dan nilai *macro average* serta *weighted average* juga mencapai angka sempurna (1.00). Ini menunjukkan bahwa model ini berhasil mengatasi ketidakseimbangan data dengan sangat baik. Berdasarkan grafik dan hasil evaluasi, stabilitas selama pelatihan serta jumlah *epoch* yang cukup berperan besar dalam peningkatan performa ini.

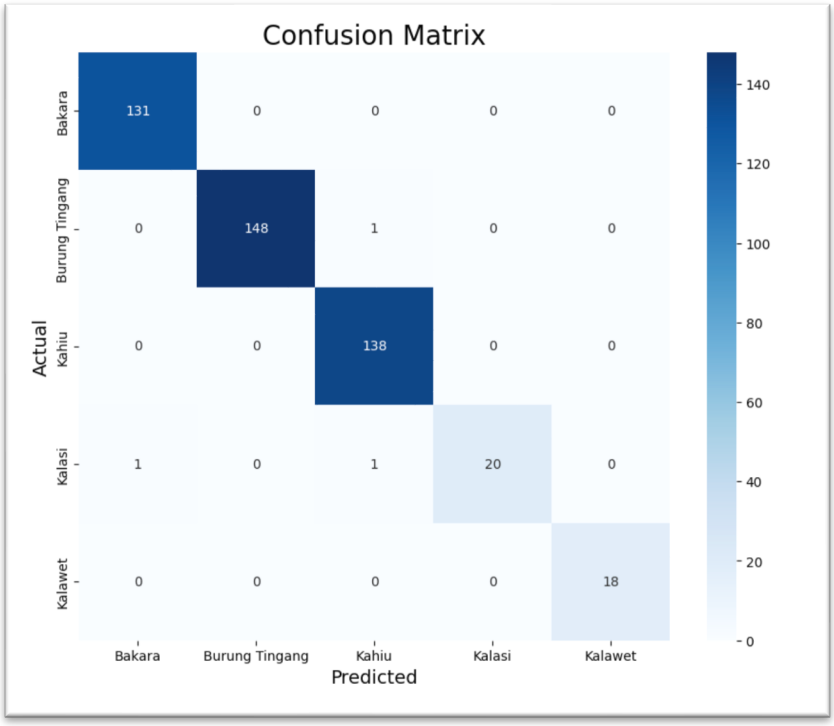
e. Evaluasi Model Berdasarkan *Confusion Matrix*

Pada tahap ini, dilakukan evaluasi menggunakan *confusion matrix* untuk menggambarkan hasil prediksi model terhadap data uji. Langkah ini bertujuan untuk mengidentifikasi distribusi prediksi benar dan salah serta mengamati pola kesalahan yang muncul pada tiap kelas.

Cara membaca confusion matrix adalah sebagai berikut;

1) Angka Nol (0): Menunjukkan tidak ada kesalahan pada kelas tertentu (misalnya, jika kolom prediksi salah bernilai nol, model tidak salah memprediksi kelas tersebut).
2) Diagonal Matriks (Miring): Angka pada diagonal menunjukkan jumlah prediksi benar (*true positive*). Semakin tinggi angka, semakin baik kinerja model untuk kelas tersebut.
3) Angka di Luar Diagonal: Menunjukkan prediksi salah (*false positive* atau *false negative*). Misalnya, angka di luar diagonal menunjukkan berapa banyak sampel dari kelas tertentu yang salah diprediksi ke kelas lain.

1. Model Pertama



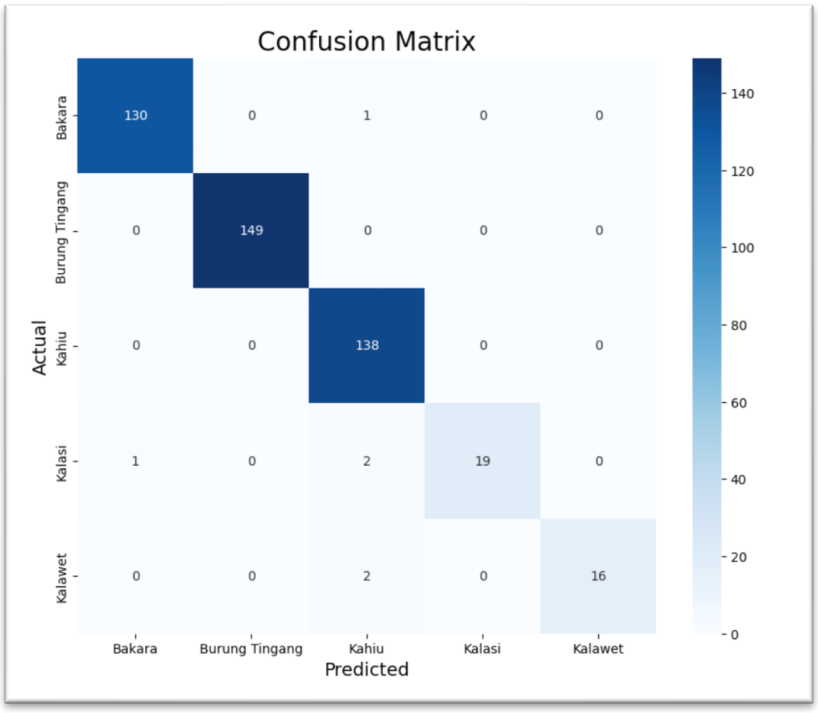
Gambar 4. 9 *Confusion Matrix* Model Pertama

Pada Gambar 4.9 untuk model pertama menunjukkan bahwa model memiliki kinerja yang sangat baik, terutama pada kelas mayoritas seperti Bakara, Burung Tingang, dan Kahiu. Pada kelas Bakara, seluruh 131 sampel berhasil diklasifikasikan dengan benar tanpa kesalahan. Hal serupa terlihat pada kelas Burung Tingang, dengan hanya satu kesalahan di mana satu sampel salah diklasifikasikan sebagai Kahiu, sedangkan 148 sampel lainnya diklasifikasikan dengan benar. Pada kelas Kahiu, model juga menunjukkan akurasi tinggi dengan seluruh 138 sampel diklasifikasikan dengan benar.

Namun, kesulitan lebih terlihat pada kelas minoritas seperti Kalasi dan Kalawet, yang memiliki jumlah sampel lebih kecil. Untuk kelas Kalasi,

terdapat dua kesalahan klasifikasi, yaitu satu sampel salah diklasifikasikan sebagai Bakara dan satu lainnya sebagai Kahiu, sementara 20 sampel diklasifikasikan dengan benar. Di sisi lain, performa pada kelas Kalawet lebih baik, dengan semua 18 sampel berhasil diklasifikasikan dengan benar. Kinerja yang lebih rendah pada kelas Kalasi dibandingkan Kalawet kemungkinan disebabkan oleh pola yang lebih sulit dikenali atau distribusi data yang tidak merata dalam proses pelatihan. Secara keseluruhan, model pertama menunjukkan hasil yang baik, tetapi masih memerlukan peningkatan dalam menangani kelas minoritas untuk meningkatkan generalisasi.

2. Model Kedua



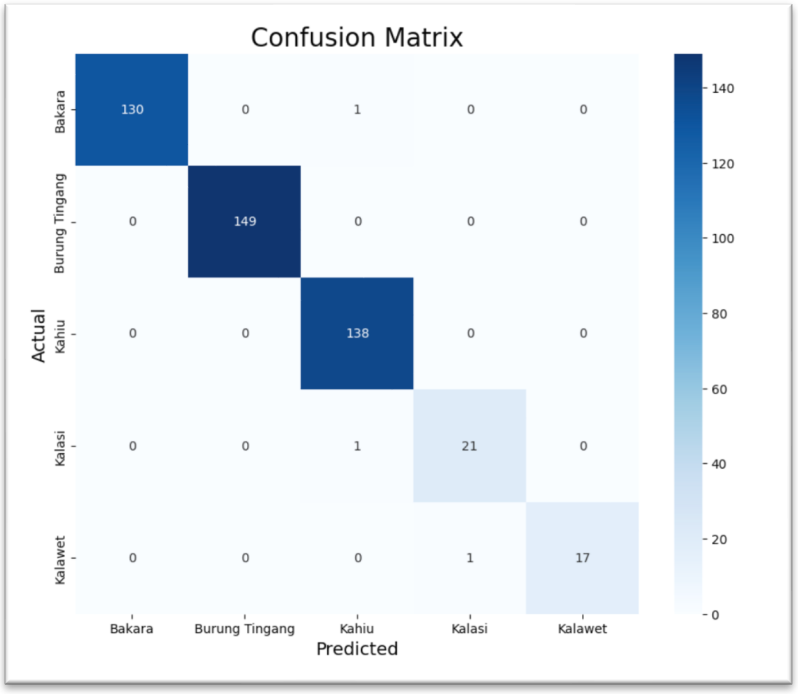
Gambar 4. 10 *Confusion Matrix* Model Kedua

Pada Gambar 4.10 untuk model kedua menunjukkan adanya penurunan performa dibandingkan model pertama, terutama dalam klasifikasi kelas

minoritas. Pada kelas mayoritas seperti Bakara, terdapat satu kesalahan, di mana satu sampel salah diklasifikasikan sebagai Kahiu, sementara 130 lainnya berhasil diklasifikasikan dengan benar. Hal serupa juga terjadi pada kelas Kahiu, di mana hanya satu sampel yang salah diklasifikasikan, sedangkan selebihnya diklasifikasikan dengan benar. Untuk kelas Burung Tingang, model menunjukkan performa sempurna dengan seluruh 149 sampel berhasil diklasifikasikan dengan benar.

Pada kelas minoritas, seperti Kalasi, penurunan performa terlihat lebih signifikan. Dari total 22 sampel, hanya 19 yang diklasifikasikan dengan benar, sementara dua sampel salah diklasifikasikan sebagai Kahiu dan satu sampel sebagai Bakara. Hal serupa terjadi pada kelas Kalawet, di mana dua dari 18 sampel salah diklasifikasikan sebagai Kalasi, sehingga hanya 16 sampel yang berhasil diklasifikasikan dengan benar. Penurunan kinerja ini menunjukkan kelemahan model kedua dalam mengenali kelas-kelas dengan jumlah data lebih kecil, yang mengindikasikan bahwa model mengalami kesulitan untuk melakukan generalisasi pada distribusi data yang tidak merata.

3. Model Ketiga

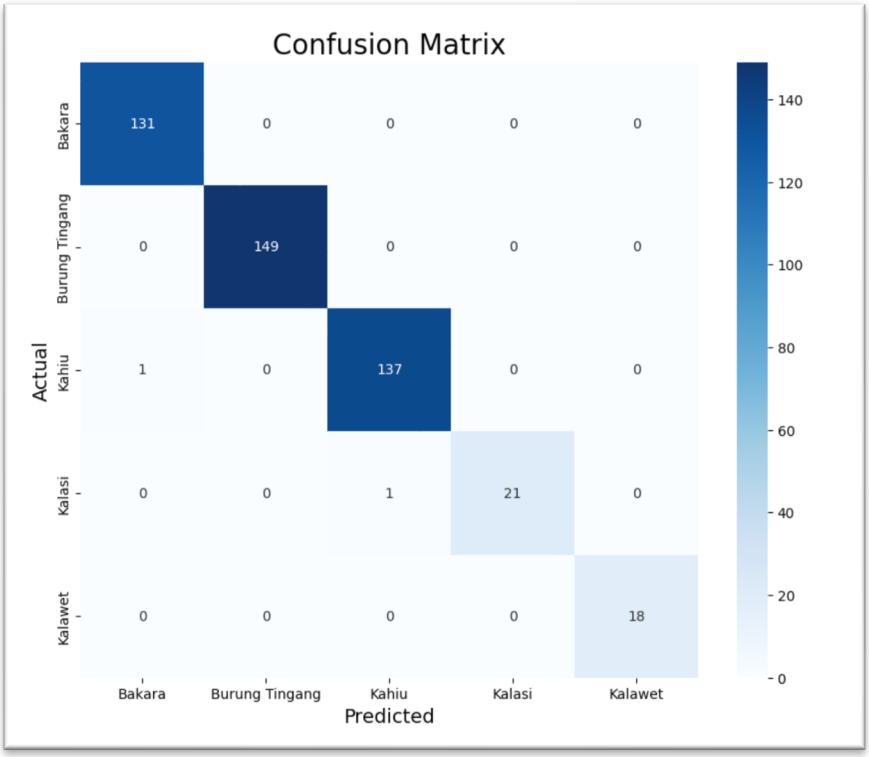


Gambar 4. 11 *Confusion Matrix* Model Ketiga

Pada Gambar 4.11 untuk model ketiga menunjukkan peningkatan performa yang signifikan dibandingkan model kedua, terutama dalam klasifikasi kelas minoritas. Pada kelas mayoritas seperti Bakara, model hanya membuat satu kesalahan, di mana satu sampel salah diklasifikasikan sebagai Kahiu, sementara 130 sampel lainnya berhasil diklasifikasikan dengan benar. Untuk kelas Burung Tingang, seluruh 149 sampel diklasifikasikan dengan benar tanpa kesalahan. Hal serupa terlihat pada kelas Kahiu, dengan hanya satu kesalahan klasifikasi, sedangkan semua sampel lainnya berhasil dikenali dengan benar.

Pada kelas minoritas, seperti Kalasi, model menunjukkan perbaikan yang mencolok. Dari 22 sampel, hanya satu yang salah diklasifikasikan sebagai Kahiu, sementara 21 sampel lainnya diklasifikasikan dengan benar. Pada kelas Kalawet, terdapat satu kesalahan, di mana satu sampel salah diklasifikasikan sebagai Kalasi, sedangkan 17 lainnya berhasil dikenali dengan benar. Perbaikan ini mengindikasikan bahwa model ketiga memiliki kemampuan generalisasi yang lebih baik dibandingkan model kedua, terutama dalam menangani kelas-kelas dengan jumlah data yang lebih kecil. Meski demikian, keberadaan beberapa kesalahan klasifikasi menunjukkan bahwa tantangan dalam mengatasi distribusi data yang tidak seimbang masih memerlukan perhatian lebih lanjut.

4. Model Keempat



Gambar 4. 12 *Confusion Matrix* Model Keempat

Pada Gambar 4.12 untuk model keempat menunjukkan performa terbaik dibandingkan semua model sebelumnya. Model ini berhasil mengklasifikasikan hampir seluruh kelas dengan akurasi tinggi, baik pada kelas mayoritas maupun minoritas. Untuk kelas Bakara, seluruh 131 sampel diklasifikasikan dengan benar tanpa kesalahan. Hal serupa terlihat pada kelas Burung Tingang, di mana semua 149 sampel berhasil dikenali dengan tepat. Pada kelas Kahiu, hanya satu kesalahan yang tercatat, yaitu satu sampel salah diklasifikasikan sebagai Kalasi, sementara 137 sampel lainnya diklasifikasikan dengan benar.

Kinerja model pada kelas minoritas juga menunjukkan hasil yang hampir sempurna. Untuk kelas Kalasi, 21 dari 22 sampel diklasifikasikan dengan benar, dengan hanya satu sampel yang salah diklasifikasikan sebagai Kahiu. Sementara itu, pada kelas Kalawet, model mencapai performa sempurna dengan semua 18 sampel diklasifikasikan dengan benar. Hasil ini mengindikasikan bahwa model keempat memiliki kemampuan generalisasi yang sangat baik, termasuk dalam menangani distribusi data yang tidak seimbang.

Peningkatan signifikan ini kemungkinan besar disebabkan oleh optimalisasi parameter pelatihan, seperti peningkatan *batch size*, penurunan *learning rate*, dan jumlah *epoch* yang lebih banyak, yang memberikan model waktu lebih lama untuk belajar secara mendalam dan stabil dari pola data yang ada. Hal ini menunjukkan bahwa model keempat tidak hanya efektif

dalam mengenali pola dari data mayoritas tetapi juga mampu mengatasi tantangan klasifikasi pada kelas minoritas.

4.1.2 Pengujian

Pada tahap pengujian ini, model yang digunakan adalah model keempat karena telah menunjukkan performa terbaik berdasarkan evaluasi sebelumnya. Pengujian dilakukan menggunakan platform Google Colab, yang menyediakan lingkungan berbasis cloud dengan dukungan GPU untuk mempermudah pelaksanaan proses.

Proses pengujian dilakukan menggunakan 10 gambar, yang terdiri dari 2 sumber utama: gambar dari internet dan hasil jepretan kamera ponsel. Gambar-gambar ini mencakup 5 kelas fauna khas Dayak, dengan distribusi sebagai berikut:

1. Kahiu: 2 gambar
2. Burung Tingang: 2 gambar
3. Bakara: 2 gambar
4. Kalasi: 2 gambar
5. Kalawet: 2 gambar

Setiap gambar diuji untuk memprediksi kelas fauna yang sesuai, dan hasil prediksi dibandingkan dengan label sebenarnya untuk menilai akurasi model. Hasil pengujian dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Pengujian Model

No	Kelas	Gambar	Prediksi	Keterangan
1	Kahiu	Gambar ke-1	Bakara	Tidak Sesuai
2	Kahiu	Gambar ke-2	Bakara	Tidak Sesuai
3	Burung Tingang	Gambar ke-1	Burung Tingang	Sesuai
4	Burung Tingang	Gambar ke-2	Burung Tingang	Sesuai
5	Bakara	Gambar ke-1	Kahiu	Tidak Sesuai
6	Bakara	Gambar ke-2	Kahiu	Tidak Sesuai
7	Kalasi	Gambar ke-1	Kalasi	Sesuai
8	Kalasi	Gambar ke-2	Kalasi	Sesuai
9	Kalawet	Gambar ke-1	Kalawet	Sesuai
10	Kalawet	Gambar ke-2	Kalawet	Sesuai

Bisa dilihat dari Tabel 4.4, kelas dengan jumlah data yang lebih besar seperti Burung Tingang menunjukkan performa yang sangat baik karena model memiliki cukup variasi data untuk mempelajari pola visual yang relevan. Sebaliknya, meskipun kelas seperti Kahiu dan Bakara memiliki jumlah data yang tidak tergolong kecil, kedua kelas ini tetap menunjukkan pola kesalahan yang konsisten. Hal ini menunjukkan bahwa karakteristik visual dari kedua kelas tersebut memiliki kesamaan yang membuat model kesulitan membedakannya.

a. Kelas Kahiu

Hasil pengujian pada Tabel 4.4 menunjukkan bahwa kelas Kahiu secara konsisten diklasifikasikan sebagai Bakara. Hal ini mengindikasikan adanya kesamaan pola visual yang signifikan antara kedua kelas, yang membuat model kesulitan dalam membedakannya. Kesalahan ini kemungkinan diperburuk oleh ketidakseimbangan distribusi *dataset* atau kurangnya variasi dalam data untuk kelas Kahiu.

Untuk mengatasi masalah ini, beberapa solusi potensial dapat diterapkan. Salah satunya adalah memperluas *dataset* untuk kelas Kahiu dengan menambahkan

gambar-gambar baru yang memiliki variasi lebih besar, seperti dari sudut pandang, latar belakang, atau kondisi pencahayaan yang berbeda. Selain itu, teknik augmentasi data dapat diterapkan untuk meningkatkan keragaman data pelatihan, sehingga model memiliki lebih banyak informasi untuk mempelajari perbedaan antara kedua kelas tersebut.

b. Kelas Burung Tingang

Pada Tabel 4.4 kelas ini menunjukkan performa yang cukup baik selama pengujian. Berdasarkan hasil *confusion matrix* dan tabel pengujian, model mampu mengklasifikasikan sebagian besar sampel dari kelas ini dengan benar. Keberhasilan ini kemungkinan besar dipengaruhi oleh jumlah *dataset* yang memadai serta karakteristik visual yang lebih khas dibandingkan dengan kelas lainnya, yang memungkinkan model untuk mempelajari pola-pola yang relevan secara efektif.

c. Kelas Bakara

Berdasarkan Tabel 4.4 mirip dengan kelas Kahiu, kelas Bakara juga menunjukkan pola kesalahan yang konsisten, di mana kelas ini sering diklasifikasikan sebagai Kahiu. Kesalahan tersebut terjadi akibat kesamaan pola visual yang signifikan antara kedua kelas, yang menyebabkan model kesulitan mengenali perbedaan di antara mereka. Selain itu, ketidakseimbangan distribusi data atau kurangnya variasi dalam dataset menjadi faktor lain yang berkontribusi terhadap masalah ini.

Untuk mengurangi kesalahan klasifikasi, langkah perbaikan pada *dataset* sangat diperlukan, seperti menambahkan lebih banyak gambar untuk kelas Bakara

serta meningkatkan variasi gambar, baik dari segi sudut pandang, latar belakang, maupun pencahayaan. Upaya lain yang dapat dilakukan adalah memperjelas perbedaan visual antara kelas Bakara dan Kahiu agar model lebih mudah membedakan keduanya selama pelatihan.

d. Kelas Kalasi

Berdasarkan Tabel 4.4 meskipun data untuk kelas Kalasi tergolong terbatas, performa pengujian menunjukkan hasil yang cukup memuaskan. Namun, terdapat beberapa kesalahan klasifikasi, terutama ketika model kesulitan membedakan kelas ini dari kelas lain dengan pola visual yang mirip. Untuk meningkatkan akurasi, diperlukan penambahan dataset dengan variasi visual yang lebih luas, seperti sudut pandang yang berbeda atau kondisi pencahayaan yang beragam, agar model dapat mengenali ciri khas kelas ini dengan lebih baik.

e. Kelas Kalawet

Pada Tabel 4.4 kelas Kalawet menunjukkan performa yang cukup konsisten, dengan sebagian besar sampel berhasil diklasifikasikan dengan benar oleh model. Hal ini diduga karena pola visual pada kelas ini cukup unik dan mudah dikenali. Meski demikian, jumlah dataset yang terbatas dapat menjadi kendala, terutama jika gambar uji memiliki perbedaan yang signifikan dibandingkan dengan data pelatihan. Untuk menjaga konsistensi performa dalam berbagai kondisi, penambahan dataset dengan variasi yang lebih beragam tetap diperlukan.

4.2 Pembahasan

Hasil implementasi dan pengujian menunjukkan bahwa keempat model memiliki karakteristik performa yang berbeda, tergantung pada konfigurasi

hyperparameter dan metode augmentasi data yang digunakan. Pengujian dilakukan dengan gambar-gambar nyata dari internet untuk mengevaluasi kemampuan model mengklasifikasikan lima kelas: Kahiu, Burung Tingang, Bakara, Kalasi, dan Kalawet. Gambar-gambar ini menghadirkan tantangan realistis berupa variasi pencahayaan, sudut pandang, resolusi, dan *noise* yang tidak terdapat dalam *dataset* pelatihan, mensimulasikan kondisi data dunia nyata yang penuh variasi.

Model pertama mencapai akurasi validasi dan pengujian yang tinggi, masing-masing 99,5% dan 99,3%. Namun, perbedaan signifikan antara akurasi pelatihan (98,2%) dan validasi menunjukkan indikasi *overfitting*. Saat diuji dengan gambar nyata, model berhasil mengklasifikasikan sebagian besar kelas mayoritas, seperti Burung Tingang dan Bakara, dengan baik. Namun, performa menurun pada kelas minoritas seperti Kalasi, mencerminkan kurangnya kemampuan generalisasi terhadap distribusi data yang tidak merata. Augmentasi data seperti *zoom* hingga 30% dan rotasi hingga 60° memberikan kontribusi pada peningkatan performa, tetapi belum cukup untuk mengatasi tantangan tersebut.

Model kedua menerapkan regularisasi L2 sebesar 0,01, yang berhasil mengurangi *overfitting*. Hal ini terlihat dari selisih akurasi pelatihan (96,9%) dan validasi (98,4%) yang lebih kecil. Meski begitu, recall pada kelas minoritas seperti Kalasi dan Kalawet menurun selama pengujian. Beberapa sampel Kalasi salah diklasifikasikan sebagai Kahiu atau Bakara, menunjukkan bahwa regularisasi saja tidak cukup untuk menangani kemiripan pola visual antar kelas. Fluktuasi selama pelatihan juga berdampak pada pengujian, membuat model kesulitan mencapai stabilitas dalam mengenali gambar dengan karakteristik berbeda dari data pelatihan.

Model ketiga, tanpa regularisasi L2, mencatat peningkatan akurasi pelatihan menjadi 98,4%. Model ini memberikan hasil lebih baik dalam mengenali kelas Kalasi, dengan recall meningkat menjadi 95%. Namun, tanda-tanda *overfitting* kembali muncul, terutama pada gambar nyata dari kelas dengan pola visual serupa, seperti Kahiu dan Bakara. Kesalahan klasifikasi antara kedua kelas ini masih sering terjadi, menunjukkan bahwa augmentasi data sederhana, seperti *zoom* sebesar 30% dan rotasi hingga 30°, belum cukup untuk membantu model membedakan pola visual yang mirip.

Model keempat, menggunakan konfigurasi *hyperparameter* optimal (*batch size* 64, *learning rate* 0,0001, dan 50 *epoch*), menunjukkan performa terbaik dalam implementasi maupun pengujian. Hasil pengujian mencatat akurasi tertinggi, yaitu 99,5%, dengan selisih kecil antara akurasi pelatihan dan validasi. Model ini mampu mengenali kelas mayoritas seperti Burung Tingang dengan sempurna. Selain itu, kelas minoritas seperti Kalasi dan Kalawet menunjukkan peningkatan performa yang signifikan dibandingkan model sebelumnya, meskipun beberapa kesalahan kecil masih terjadi. Augmentasi data yang lebih moderat (*zoom* sebesar 20% dan rotasi hingga 30°) membantu model fokus pada pola utama, sementara pengaturan *learning rate* yang lebih rendah memungkinkan pembelajaran yang lebih stabil.

Pengujian juga menyoroti tantangan utama dalam membedakan kelas dengan pola visual serupa, seperti Kahiu dan Bakara. Kesamaan pola ini sering menyebabkan kesalahan klasifikasi di mana sampel dari satu kelas salah dikenali sebagai kelas lain. Selain itu, variasi pencahayaan dan sudut pandang yang signifikan pada gambar uji menjadi penyebab utama kesalahan klasifikasi.

Secara keseluruhan, model keempat terbukti unggul dalam implementasi dan pengujian. Kombinasi konfigurasi *hyperparameter* yang tepat, augmentasi data yang moderat, dan jumlah *epoch* yang cukup memungkinkan model ini mencapai akurasi tinggi sekaligus kemampuan generalisasi yang baik terhadap data dunia nyata.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian mengenai Klasifikasi Citra Hewan Khas Dayak Menggunakan *Convolutional Neural Network* (CNN), kesimpulan yang dapat diambil adalah:

a. Model *Convolutional Neural Network* (CNN) berbasis *transfer learning* dengan arsitektur *InceptionV3* berhasil diperbaiki untuk mengklasifikasikan hewan khas suku Dayak.
b. Evaluasi menunjukkan bahwa model utama yang dihasilkan memiliki tingkat akurasi mencapai 99,5% dan *F1-Score* sebesar 1,00, sehingga terbukti sangat akurat dalam mengenali dan mengklasifikasikan fauna.
c. Proses *preprocessing*, seperti pengubahan ukuran gambar (*resizing*), normalisasi, dan augmentasi data, terbukti efektif dalam meningkatkan kualitas dataset dan mengatasi ketidakseimbangan jumlah data antar kelas.
d. Teknologi yang dihasilkan mendukung upaya pelestarian fauna khas suku Dayak dan dapat menjadi sarana untuk meningkatkan kesadaran masyarakat terhadap pentingnya konservasi spesies lokal.
e. Penerapan *transfer learning* memungkinkan pemanfaatan *dataset* yang terbatas dengan hasil yang sangat optimal, sehingga cocok untuk diterapkan pada penelitian serupa di masa mendatang.

5.2 Saran

Sebagai langkah pengembangan dan tindak lanjut penelitian ini, berikut beberapa saran yang dapat diajukan untuk mendukung peningkatan kualitas penelitian serupa di masa mendatang:

a. Peningkatan variasi *dataset* perlu dilakukan, baik dari segi jumlah citra maupun keragaman kondisi lingkungan, seperti pencahayaan, sudut pandang, dan latar belakang, untuk meningkatkan kemampuan generalisasi model pada kondisi dunia nyata.
b. Integrasi model ke dalam aplikasi berbasis *web* atau mobile disarankan agar hasil penelitian ini dapat diakses secara luas oleh masyarakat, sehingga memberikan dampak nyata dalam edukasi dan pelestarian budaya lokal.
c. Melibatkan ahli biologi dan konservasi fauna dalam pengembangan sistem klasifikasi untuk memberikan validasi lebih lanjut atas pengenalan citra dan menambah informasi ekologi pada *platform*.
d. Penelitian lanjutan juga dapat mencakup klasifikasi *multi*-spesies fauna lokal lainnya, sehingga memperluas cakupan konservasi dan edukasi pada ekosistem lokal Kalimantan Tengah.

DAFTAR PUSTAKA

Abdullah, R. R. F., Alhamad, A. & Hasan, M., 2021. Penerapan Convolutional Neural Network (CNN) Untuk Klasifikasi Penggunaan Masker. *Jurnal Balok,* Volume I, pp. 79-84.

Alghifari, F. & Juardi, D., 2021. Penerapan Data Mining Pada Penjualan Makanan Dan Minuman Menggunakan Metode Algoritma Naïve Bayes. *Jurnal Ilmiah Informatika,* Volume IX.

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, pp.1-33.

Aprilia, N.P. & Rochadiani, T.H., 2024. Image Captioning untuk Gambar Rambu Lalu Lintas Indonesia Menggunakan Pretrained CNN dan Transformer. Indonesian Journal of Computer Science, 13(3), p.4627. Available at: https://ijcs.stmikindonesia.ac.id [Accessed 7 Nov. 2024].

Azhar, K. M., Santoso, I. & Soetrisno, Y. A. A., 2021. Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network dan Algoritma Yolo dalam Sistem Pendeteksi Uang Kertas Rupiah Bagi Penyandang Low

Bellboy, M., 2024. 12 Hewan Khas Kalimantan yang Unik dan Terancam Punah. [Online] Available at: https://www.traveloka.com/id-id/explore/destination/hewan-khas-kalimantan-yang-unik-dan-terancam-punah-acc/356783 [Accessed 7 November 2024].

Bisong, E., 2019. *Building Machine Learning and Deep Learning Models on Google Cloud Platform.* 1st Edition ed. Ottawa: Apress Berkeley.

Brital, A., 2021. Inception V3 CNN Architecture Explained. [Online] Available at: https://medium.com/@AnasBrital98/inception-v3-cnn-architecture-explained-691cfb7bba08 [Accessed 25 January 2025].

Coursera, 2024. *What Is NumPy?.* [Online] Available at: https://www.coursera.org/articles/what-is-numpy [Accessed 6 October 2024].

Diantoro, K. & Adriansyah, B., 2019. Sistem Identifikasi Jenis Burung Dengan Image Classification Menggunakan OpenCV. *TEKINFO,* Volume XX, pp. 96-105.

Dicoding, 2023. *Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya.* [Online] Available at: https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/ [Accessed 5 October 2024].

Dumane, G., 2020. Introduction to Convolutional Neural Network (CNN) using Tensorflow. [Online] Available at:

https://towardsdatascience.com/introduction-to-convolutional-neural-network-cnn-de73f69c5b83 [Accessed 25 January 2025].

Ebbehoj, A., Thunbo, M.Ø., Andersen, O.E., Glindtvad, M.V. & Hulman, A., 2022. Transfer learning for non-image data in clinical research: A scoping review. PLOS Digital Health, pp.1-22.

Faizal, S. & Sundaresan, S., 2022. Wild Animal Classifier Using CNN. Chennai: Vellore Institute of Technology, School of Computer Science and Engineering.

Gibbonesia, 2022. Owa Peliharaan Diserahkan Warga ke BKSDA Kalteng. [Online] Available at: https://gibbonesia.id/owa-peliharaan-diserahkan-warga-ke-bksda-kalteng/ [Accessed 25 January 2025].

Goodfellow, I., Bengio, Y. and Courville, A., 2016. Deep learning. Cambridge: MIT Press.

Huda, N., Mahiruna, A., Sulistijanti, W. & Noor Santi, R.C., 2023. Analisis Performa InceptionV3 Convolutional Network pada Klasifikasi Varietas Daun Grapevine. Jurnal Sains Komputer dan Teknologi Informasi, 5(2), pp.48-54.

Istiqomah, N., 2024. 5 Spesies Monyet Dunia Lama yang Berstatus Hewan Endemik Indonesia. [Online] Available at: https://www.idntimes.com/science/discovery/nisa-istiqomah-1/spesies-monyet-endemik-indonesia-c1c2 [Accessed 25 January 2025].

KLHK, 2017. Cegah Karhutla, untuk Selamatkan Orangutan. [Online] Available at: https://ppid.menlhk.go.id/berita/berita-foto/709/cegah-karhutla untuk-selamatkan-orangutan [Accessed 25 January 2025].

Kosman, A.W., Wahyuningsih, Y. & Mahendrasusila, F., 2024. Pengujian Algoritma InceptionV3 dalam Mengidentifikasi Penyakit Tuberculosis dan Pneumonia. Jurnal Teknik Informatika dan Komputer, e-ISSN 2829-0208. Available at: https://journal.uhamka.ac.id/index.php/jutikom [Accessed 7 Nov. 2024].

Maulana, M. I., Nishom, M. & Af'idah, D. I., 2022. Pengolahan Citra untuk Identifikasi Pelat Nomor Kendaraan Mobil Menggunakan Metode Haar Cascade dan Optical Character Recognition. *Jurnal Bumigora Information Technologi (BITe),* Volume IV, pp. 1-16.

Mediabudayaindonesia, 2022. Burung Enggang, Burung Yang Dikeramatkan Oleh Suku Dayak. [Online] Available at: https://mediabudayaindonesia.com/news/show/2505202213-burung-enggang-burung-yang-dikeramatkan-oleh-suku-dayak [Accessed 25 January 2025].

Narkhede, S., 2018. Understanding Confusion Matrix. [Online] Available at: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62 [Accessed 25 January 2025].

Nurhidayat, R. & Dewi, K. E., 2023. Penerapan Algoritma K-Nearest Neighbor dan Fitur Ekstraksi N-Gram dalam Analisis Sentimen Berbasis Aspek. *KOMPUTA,* Volume XII, pp. 91-100.

Radya, M., 2024. *Mengenal Matplotlib Untuk Visualisasi Data.* [Online] Available at: https://blog.indobot.co.id/mengenal-matplotlib-untuk-visualisasi-data/
[Accessed 6 October 2024].

Radya, M., 2024. *Yuk Mengenal Apa Itu TensorFlow? Dan Apa Fungsinya?.* [Online]
Available at: https://blog.indobot.co.id/yuk-mengenal-apa-itu-tensorflow-dan-apa-fungsinya/
[Accessed 6 October 2024].

Ranti, S., 2022. *Sejarah hingga Perkembangan Google Chrome.* [Online] Available at: https://tekno.kompas.com/read/2022/09/18/16030007/sejarah-hingga-perkembangan-google-chrome?page=all#google_vignette
[Accessed 6 October 2024].

Romadloni, P. L., Kusuma, B. A. & Baihaqi, W. M., 2022. Komparasi Metode Pembelajaran Mesin untuk Implementasi Pengambilan Keputusan dalam Menentukan Promosi Jabatan Karyawan. *JATI,* Volume XI, pp. 622-628.

Sefrila, E.D., Rahmat, B. & Sihananto, A.N., 2024. Implementasi Arsitektur InceptionV3 dengan Optimasi Adam, SGD dan RMSP Pada Klasifikasi Penyakit Malaria. BRIDGE: Jurnal Publikasi Sistem Informasi dan Telekomunikasi, 2(2), pp.69-84.

TensorFlow, 2023. *Keras: The high-level API for TensorFlow.* [Online] Available at: https://www.tensorflow.org/guide/keras
[Accessed 6 October 2024].

Widyadhana, A., Wahyuningsih, Y. & Mahendrasusila, F., 2024. Pengujian metode Inception V3 dalam mengidentifikasi penyakit kanker kulit. Jurnal Teknologi Informatika dan Komputer, 10(1), pp. 136. Available at: http://journal.thamrin.ac.id/index.php/jtik/article/view/1940/pdf [Accessed 23 January 2025].

Wijaya, A., 2023. Gubernur Kalsel Dukung Pelepasan Sembilan Bekantan. [Online] Available at: https://www.rri.co.id/daerah/223949/gubernur-kalsel-dukung-pelepasan-sembilan-bekantan [Accessed 25 January 2025].

Yu, C.H., Jia, Z., Zheng, Z., Shao, J., Fan, H., Liu, Y., Zhou, Y., Li, M., Huang, G., Wang, J., Shen, H. & Wang, Y., 2023. RAF: Holistic Compilation for Deep Learning Model Training. *arXiv.*

Zain, H. H. & Fauzan, M. N., 2022. *Peramalan Terhadap Permintaan Produk Dalam Sektor Logistik di Indonesia.* Bandung: Penerbit Buku Pedia.

Zeng, P., 2021. Research on Similar Animal Classification Based on CNN Algorithm. Journal of Physics: Conference Series, 2132(1), p. 012001. IOP Publishing. doi:10.1088/1742-6596/2132/1/012001.

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M. & Parmar, M., 2024. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*.

LAMPIRAN

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS

No.337/STMIK-3.C.2/KP/XI/2024

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan nama- nama tersebut di bawah ini :

1. Nama : Dr. Maura Widyaningsih, S.Kom., M.Cs.
   NIK : 197502232009001
   Sebagai Pembimbing I Dalam Pembuatan Program
2. Nama : Lili Rusdiana, M.Kom.
   NIK : 198707282011007
   Sebagai Pembimbing II Dalam Penulisan Tugas Akhir

Untuk membimbing Tugas Akhir mahasiswa :

Nama	:	Jeki
NIM	:	C2155201001
Program Studi	:	TEKNIK INFORMATIKA (55201)
Tanggal Perubahan	:	13 November 2024
Judul Tugas Akhir	:	Klasifikasi Citra Hewan Khas Suku Dayak Dengan Menggunakan CNN
Keterangan	:	Perubahan judul TA setelah seminar proposal TA

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 13 November 2024
Ketua Program Studi Teknik Informatika,

Lili Rusdiana, M.Kom.
NIK. 198707282011007

Tembusan :

1. Pembimbing I dan II
2. Mahasiswa yang bersangkutan
3. Arsip

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangkaraya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

KARTU KEGIATAN KONSULTASI TUGAS AKHIR

Nama Mahasiswa : Jeki
NIM : C2155201001
Tanggal Persetujuan Judul :
Judul Tugas Akhir : Klasifikasi Dengan Klasifikasi CNN Menggunakan Citra Hewan Khas Dayak

No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
1	19/9	19/9	Teknik penyusunan Proposal TA	
2	5/10	7/10	menambahkan referensi dan Latar belakang	
3	10/10	10/10	Revisi Judul	
4	27/9	28/9	lengkapi bab 1, lanjut bab 2	
5	4/10	4/10	lengkapi bab 2, lanjut bab 3	
6	15/10	15/10	lengkapi bab 3	
7	16/10	16/10	Kapikan dan lengkapi semua penulisan	
8	18/10	18/10	bab 1-3	
9	18/12 2024	18/1 2025	Rancangan Ide program	
10	11/1 2025	11/1 2025	~~Acc~~, konsultasi program	
11	16/1 2025	16/1 2025	penulisan isi bab 4 dan bab 5	
12	17/1/25	17/1 25	Acc maju sidang	
13	14/12 24	14/12 24	Lanjutkan bab 4	
14	15/10 25	15/10 25	Penulisan bab 4 perbaikan dgn isian pada judul & pedoman	
15	17/1 25	17/1 25	lengkapi semua penulisan	
16	18/1 25	18/1 25	Acc sidang	

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS PENGUJI TUGAS AKHIR

No. 11/STMIK-3.C.2/KP/I/2025

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan kepada nama- nama berikut :

1. Nama : Sulistyowati, S.Kom., M.Cs.
   NIK : 198212162007002
   Sebagai Ketua
2. Nama : Elok Faiqotul Himmah, S.Si., M.Sc.
   NIK : 198503092009003
   Sebagai Sekretaris
3. Nama : Hotmian Sitohang, M.Kom.
   NIK : 198503282008002
   Sebagai Anggota
4. Nama : Dr. Maura Widyaningsih, S.Kom., M.Cs.
   NIK : 197502232009001
   Sebagai Anggota
5. Nama : Lili Rusdiana, M.Kom.
   NIK : 198707282011007
   Sebagai Anggota

Tim Penguji Tugas Akhir Mahasiswa :

Nama	:	Jeki
NIM	:	C2155201001
Hari/ Tanggal Ujian	:	Selasa, 21 Januari 2025
Waktu	:	08.00 WIB
Judul Tugas Akhir	:	Klasifikasi Citra Hewan Khas Suku Dayak Dengan Menggunakan CNN

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 18 Januari 2025
Ketua Program Studi Teknik Informatika,

Lili Rusdiana, M.Kom.
NIK. 198707282011007

Tembusan :
1. Dosen Penguji
2. Mahasiswa yang Bersangkutan

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

BERITA ACARA
UJIAN TUGAS AKHIR

Periode (Bulan) : Januari Tahun 2025

1. Hari/Tanggal Ujian : Selasa / 21 Januari 2025
2. Waktu (Jam) : 08.00 WIB sampai dengan 09.00WIB
3. Nama Mahasiswa : Jeki
4. Nomor Induk Mahasiswa : C2155201001
5. Program Studi : Teknik Informatika
6. Tahun Angkatan : 2021
7. Judul Tugas Akhir : Klasifikasi Citra Hewan Khas Suku Dayak Dengan Menggunakan CNN

8. Dosen Penguji :

	Nama		Nilai	Tanda Tangan
1.	Sulistyowati	=		( )
2.	Elok F.H	=		( )
3.	Hotmian S	=		( )
4.	Maura W	=		( )
5.	Lili Rusdiana	=		( )

9. Hasil Ujian : **LULUS** / ~~**TIDAK LULUS**~~ *) NILAI = 83,32

Dengan Perbaikan/ ~~*Tanpa Perbaikan*~~ *)

10. Catatan Penting : 1. Lama Perbaikan : 7 hari
    2. Jika lebih dari 1 (satu) bulan dikenakan sanksi berupa denda sebesar Rp. 600.000,- (Enam ratus ribu rupiah) per bulan dari tanggal ujian
    3. Jika lebih dari 3 (tiga) bulan dari tanggal ujian maka hasil ujian dibatalkan dan wajib mengajukan judul dan pembimbing baru

Palangka Raya, 21 - 1 - 2025

Mengetahui :
Ketua Program Studi Teknik Informatika,

Lili Rusdiana, M.Kom.
NIK. 198707282011007

Ketua Penguji,

NIK.

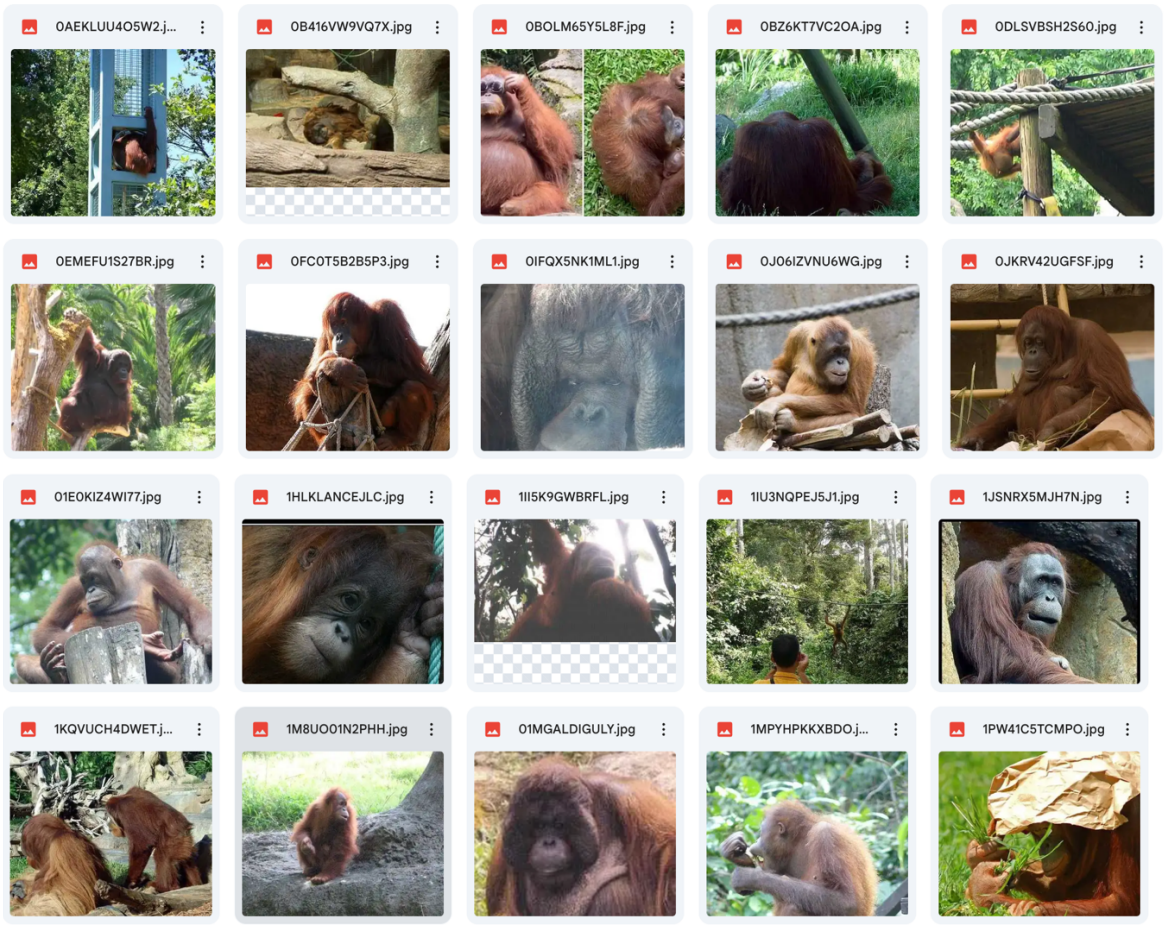
Tembusan:
1. Arsip Prodi Teknik Informatika
2. Mahasiswa yang bersangkutan

Dibawa saat konsultasi perbaikan dengan dosen penguji

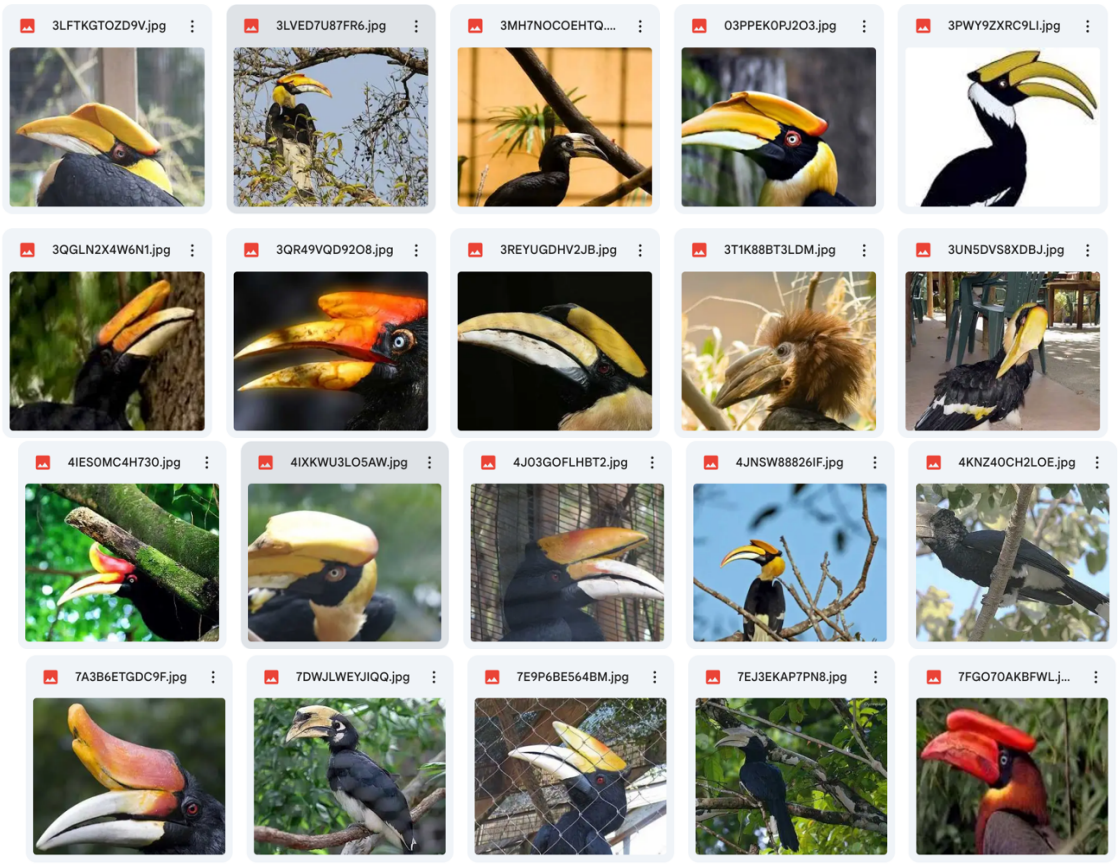
**) Coret yang tidak perlu*

Sampel Dataset

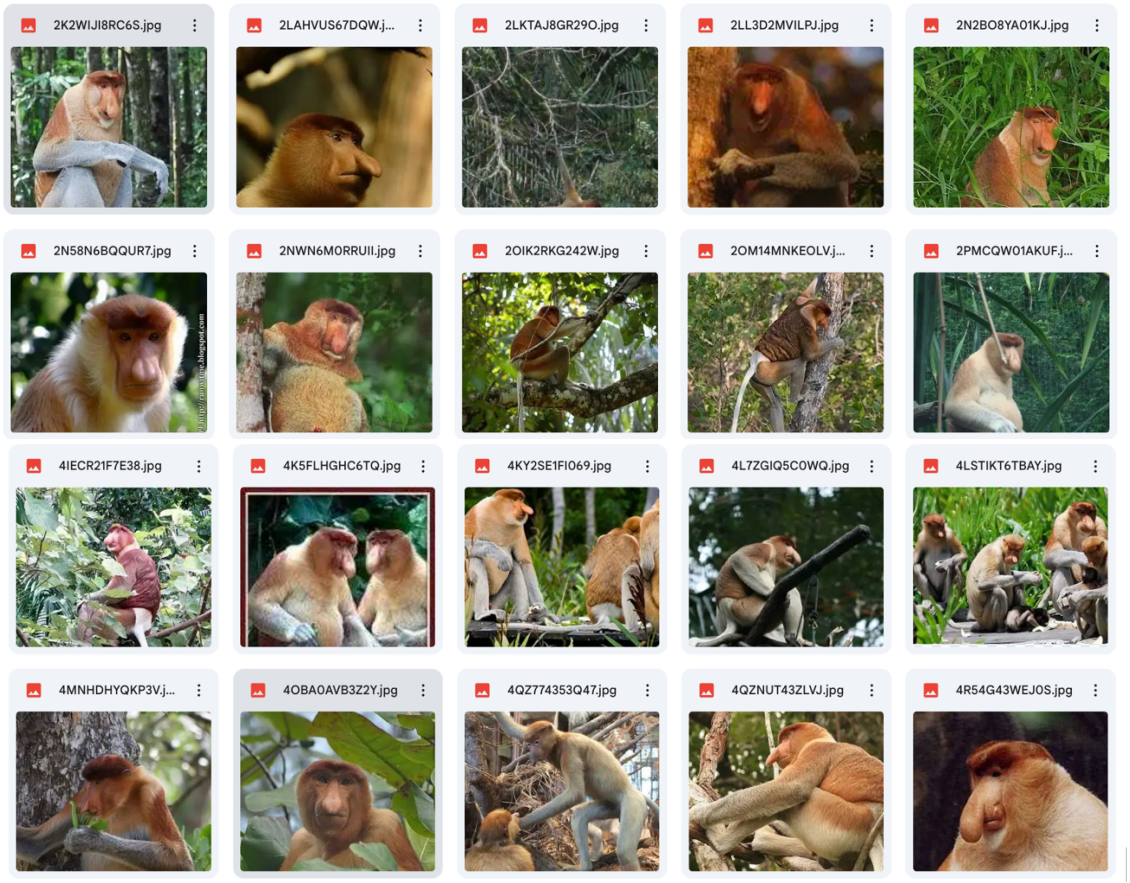
1. Kelas Kahiu
   Ditampilkan 20 gambar dari 1367 gambar.



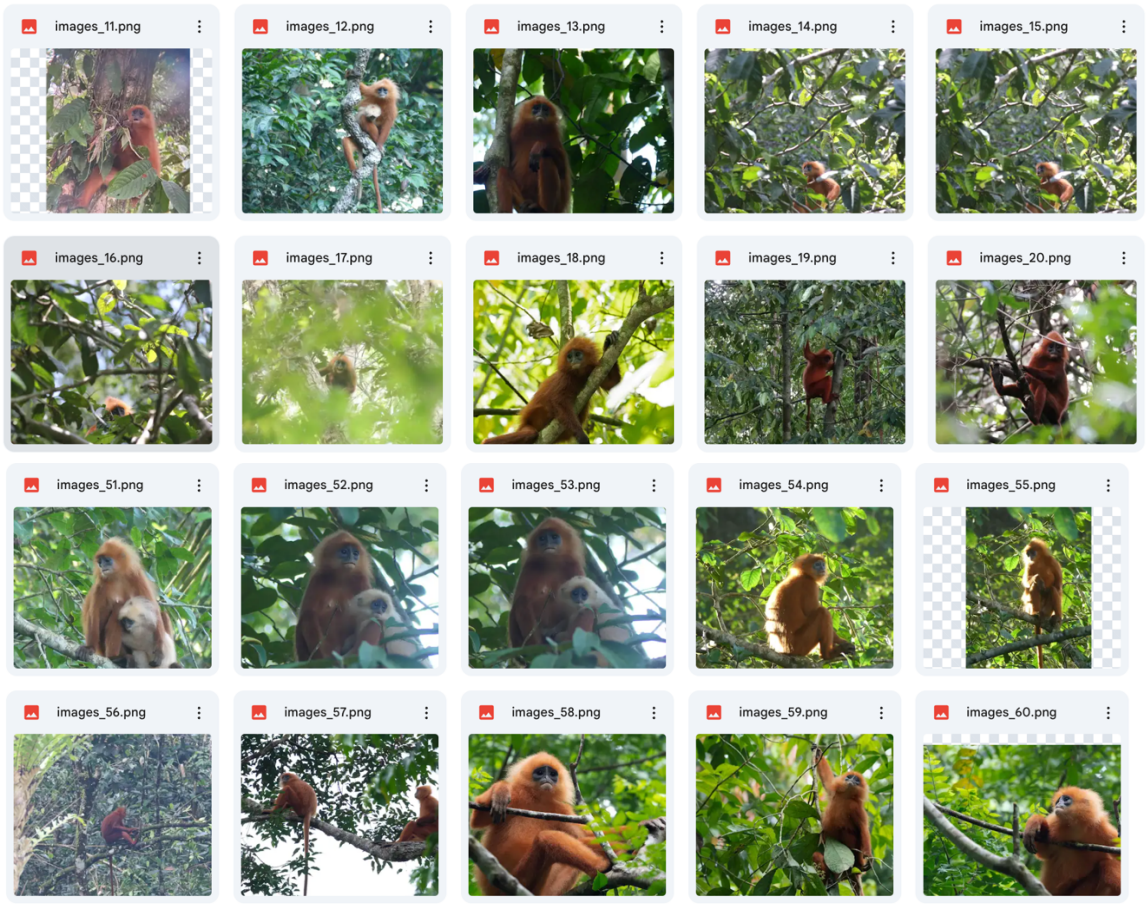
2. Kelas Burung Tingang
   Ditampilkan 20 gambar dari 1483 gambar.



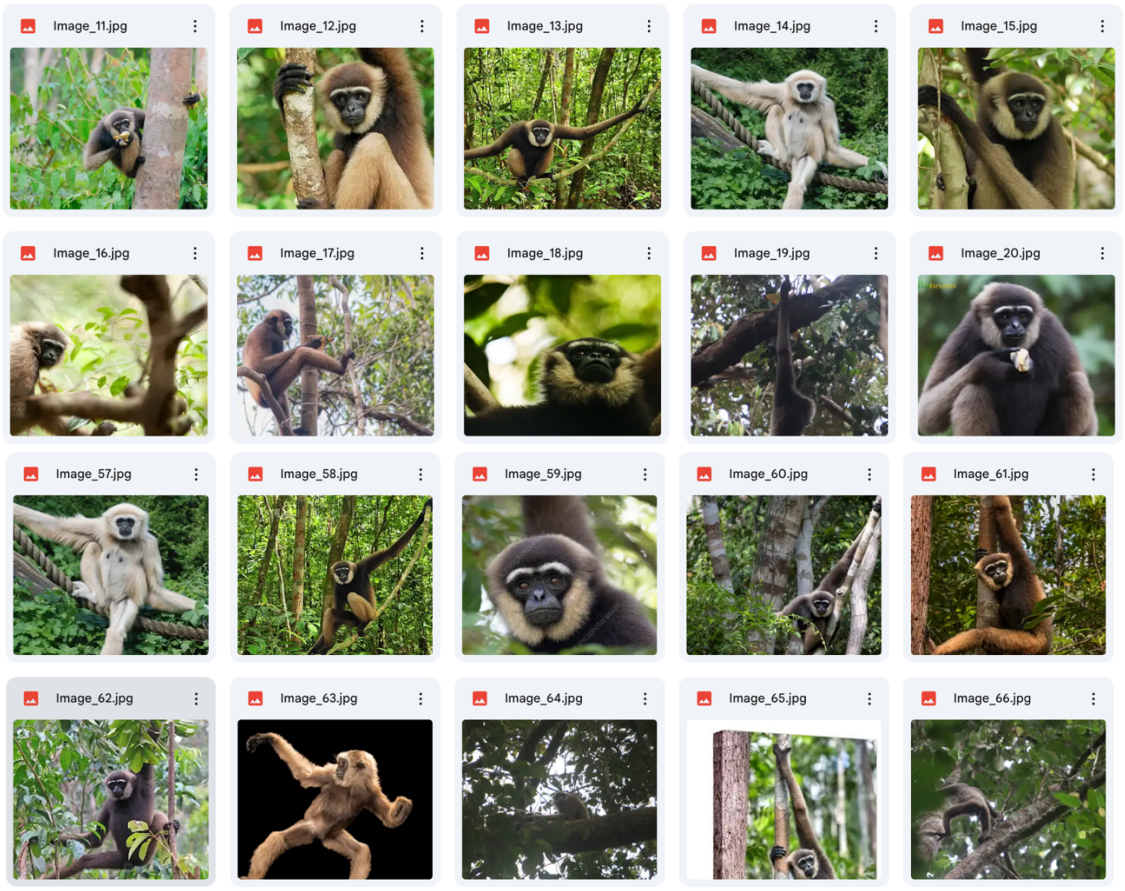
3. Kelas Bakara
   Ditampilkan 20 gambar dari 1296 gambar.



4. Kelas Kalasi
   Ditampilkan 20 gambar dari 213 gambar.



5. Kelas Kalawet
   Ditampilkan 20 gambar dari 161 gambar.



Listing Program

1. *Preprocessing*

```
train_datagen = ImageDataGenerator(rescale=1./255.,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   horizontal_flip=True,
                                   shear_range=0.2,
                                   rotation_range=30,
                                   zoom_range=0.2,
                                   fill_mode='nearest'
                                  )


train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(299, 299),
    batch_size=64,
    class_mode='categorical'
)

val_datagen = ImageDataGenerator(
    preprocessing_function=tf.keras.applications.inception_v3.preprocess_input
)

validation_generator = val_datagen.flow_from_directory(
    valid_dir,
    target_size=(299, 299),
    batch_size=32,
    class_mode='categorical'
)

test_datagen = ImageDataGenerator(
    preprocessing_function=tf.keras.applications.inception_v3.preprocess_input
)
test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(299, 299),
    batch_size=32,
    class_mode='categorical',
    shuffle=False  # Important to keep the order consistent
)
```

2. *Modeling*

```
def create_model():
    base_model = tf.keras.applications.InceptionV3(
        weights='imagenet',
        include_top=False,
        input_shape=(299, 299, 3)
    )
    base_model.trainable = False

    x = base_model.output
    x = GlobalAveragePooling2D()(x)
    x = Dropout(0.5)(x)
    x = Dense(256, activation='relu')(x)
    x = Dropout(0.5)(x)
    x = Dense(5, activation='softmax')(x)

    model = tf.keras.Model(inputs=base_model.input, outputs=x)

    return model
```

3. *Training*

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001),
              loss="categorical_crossentropy",
              metrics=["accuracy"])
```

```
early_stopping = EarlyStopping(monitor='val_loss',
                               patience=10,
                               restore_best_weights=True)
```

```
history = model.fit(
    train_generator,
    validation_data=validation_generator,
    epochs=50,
    callbacks=early_stopping
)
```

4. Evaluasi

```
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
plt.plot(histo['loss'],label='Train Loss')
plt.plot(histo['val_loss'],label='Validation Loss')
plt.title('Train Loss & Validation Loss',fontsize=20)
plt.legend()
plt.subplot(1,2,2)
plt.plot(histo['accuracy'],label='Train Accuracy')
plt.plot(histo['val_accuracy'],label='Validation Accuracy')
plt.title('Train Accuracy & Validation Accuracy',fontsize=20)
plt.legend()
```

5. *Confusion Matrix*

```
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

# Predict the classes of the test set
y_pred = model.predict(test_generator)
y_pred_classes = np.argmax(y_pred, axis=1)  # Get the predicted class indices
y_true = test_generator.classes  # True class indices

# Get class labels
class_labels = list(test_generator.class_indices.keys())

# Print classification report
print("\nClassification Report:")
print(classification_report(y_true, y_pred_classes, target_names=class_labels))

# Compute confusion matrix
cm = confusion_matrix(y_true, y_pred_classes)

# Plot confusion matrix
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=class_labels, yticklabels=class_labels)
plt.title('Confusion Matrix', fontsize=20)
plt.ylabel('Actual', fontsize=14)
plt.xlabel('Predicted', fontsize=14)
plt.show()
```

6. *Testing*

```
# Define class names (manually added based on dataset)
class_names = ['Kahiu', 'Bakara', 'Burung Tingang', 'Kalasi', 'Kalawet']
```

```
def load_and_preprocess_image(img_path, target_size=(299, 299)):
    img = image.load_img(img_path, target_size=target_size)
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)  # Add batch dimension
    img_array = img_array / 255.0  # Rescale image
    return img_array

# Predict image
def predict_image(model, img_path, class_labels):
    img_array = load_and_preprocess_image(img_path)
    predictions = model.predict(img_array)
    predicted_class_index = np.argmax(predictions)
    predicted_class_label = class_labels[predicted_class_index]
    return predicted_class_label, predictions[0][predicted_class_index]
```

```
# Upload an image
uploaded = files.upload()
for filename in uploaded.keys():
    img_path = filename
    predicted_label, confidence = predict_image(model, img_path, class_names)

    print(f"Uploaded image: {filename}")
    print(f"Predicted class: {predicted_label} with confidence: {confidence:.2f}")
```