



Implementasi CNN Model Resnet50 Klasifikasi Citra Penyakit *Lumpy Skin Disease* Pada Sapi

Willi Novrian¹, Yusran Panca Putra², Agus Susanto³, Julia Purnama Sari⁴, Rifaldo Pratama⁵

^{1,3,4}Sistem Informasi, Fakultas Teknik, Universitas Bengkulu

²Informatika, Fakultas Teknik, Universitas Bengkulu

⁵Sistem Informasi, Universitas Islam Sumatera Barat

willinovrian@unib.ac.id, Yusranpanca@unib.ac.id, Agus.susanto@unib.ac.id, Juliapurnama@unib.ac.id,
rifaldopr@gmail.com

Abstract

The livestock sector plays an important role in national food security but it faces challenges from infectious diseases such as Lumpy Skin Disease (LSD) which reduce productivity. This study aims to implement a Convolutional Neural Network (CNN) by using the ResNet-50 architecture to classify cattle skin images into healthy and LSD-infected categories. The dataset consists of 1,024 images that were split into training and testing sets with an 80:20 ratio. The model is trained by using a transfer learning approach with freeze and fine-tuning stages. Experimental results show that the model achieves an accuracy of 93.17% with balanced performance in detecting both classes. This study contributes to the application of ResNet-50-based deep learning for automated livestock disease diagnosis.

Keywords: Lumpy Skin Disease, Cattle, Convolutional Neural Network, ResNet-50, Deep Learning.

Abstrak

Sektor peternakan sapi memiliki peran penting dalam ketahanan pangan nasional, namun menghadapi tantangan penyakit menular seperti *Lumpy Skin Disease* (LSD) yang berdampak pada penurunan produktivitas. Penelitian ini bertujuan mengimplementasikan *Convolutional Neural Network* (CNN) dengan arsitektur ResNet-50 untuk mengklasifikasikan citra kulit sapi menjadi kategori sehat dan terinfeksi LSD. Dataset yang digunakan terdiri dari 1.024 citra yang dibagi menjadi data latih dan uji dengan rasio 80:20. Model dilatih menggunakan pendekatan *transfer learning* melalui tahap *freeze* dan *fine-tuning*. Hasil pengujian menunjukkan model mampu mencapai akurasi sebesar 93,17% dengan performa yang seimbang dalam mendeteksi kedua kelas. Penelitian ini berkontribusi pada penerapan *deep learning* berbasis ResNet-50 untuk diagnosis penyakit ternak secara otomatis.

Kata Kunci: *Lumpy Skin Disease, Sapi, Convolutional Neural Network, ResNet-50, Deep Learning.*



1. Pendahuluan

Penelitian mengenai deteksi penyakit pada ternak, khususnya berbasis citra, telah mengalami perkembangan pesat seiring dengan kemajuan teknologi *Artificial Intelligence* (AI), terutama pada bidang *computer vision* dan *deep learning*. Salah satu pendekatan yang paling banyak digunakan adalah *Convolutional Neural Network* (CNN), yang terbukti unggul dalam mengekstraksi fitur visual secara otomatis dibandingkan metode konvensional.

Beberapa penelitian terdahulu menunjukkan bahwa CNN mampu memberikan performa yang tinggi dalam klasifikasi citra medis maupun citra hewan. Misalnya, model berbasis CNN mampu mengklasifikasikan lesi kulit dengan tingkat akurasi yang tinggi. Hal ini menunjukkan bahwa pendekatan berbasis *deep learning* telah menjadi *state of the art* dalam bidang klasifikasi citra penyakit [1,3]. Penyakit ini menyebabkan tingkat morbiditas yang tinggi, menggerus margin keuntungan peternak, mengganggu rantai perdagangan, dan mengancam ketahanan pangan. Oleh karena itu, deteksi dini dan penanganan yang cepat menjadi sangat penting [7].

Dalam konteks peternakan, penerapan CNN juga telah dilakukan untuk berbagai kebutuhan, seperti deteksi kondisi reproduksi dan klasifikasi produk ternak. berhasil mengimplementasikan model ResNet-50 untuk mendeteksi fase estrus pada sapi dengan akurasi mencapai 96% pada data pelatihan dan 80% pada data pengujian [14]. Selain itu, menggunakan ResNet-50 untuk klasifikasi citra daging sapi dan babi dengan akurasi sebesar 97,83%. Hasil ini menunjukkan bahwa arsitektur CNN, khususnya ResNet-50, memiliki kemampuan yang sangat baik dalam mengenali pola visual pada objek terkait peternakan [3].

Lebih lanjut, arsitektur ResNet50 merupakan salah satu model yang termasuk dalam kategori *state of the art* karena kemampuannya dalam mengatasi masalah *vanishing gradient* melalui pendekatan *residual learning*. Keunggulan ini memungkinkan model untuk membangun jaringan yang lebih dalam tanpa kehilangan performa, sehingga mampu menangkap fitur yang lebih kompleks dan detail. Selain itu, dukungan terhadap *transfer learning* menjadikan ResNet50 sangat efisien untuk diterapkan pada dataset dengan jumlah terbatas, seperti citra penyakit pada ternak.

Meskipun demikian, penelitian terkait deteksi *Lumpy Skin Disease* (LSD) pada sapi berbasis citra masih relatif terbatas, khususnya di Indonesia. Sebagian besar penelitian sebelumnya masih berfokus pada identifikasi gejala secara manual atau berbasis observasi langsung. Oleh karena itu, terdapat *research gap* dalam pemanfaatan teknologi *deep learning*, khususnya CNN dengan arsitektur ResNet50, untuk deteksi otomatis penyakit LSD pada ternak sapi. Kebaruan dalam penelitian ini terletak pada penerapan model CNN dengan arsitektur ResNet50 untuk deteksi penyakit *Lumpy Skin Disease* (LSD) pada ternak sapi berbasis citra, yang masih jarang diteliti. Selain itu, penelitian ini juga mengoptimalkan pendekatan *transfer learning* pada dataset terbatas serta berfokus pada pengembangan sistem deteksi dini yang lebih cepat dan non-invasif. Dengan demikian, penelitian ini tidak hanya mengadopsi metode yang sudah ada, tetapi juga memberikan kontribusi baru dalam konteks implementasi pada penyakit spesifik di sektor peternakan.

Berdasarkan tinjauan tersebut, dapat disimpulkan bahwa penggunaan CNN, khususnya ResNet50, merupakan pendekatan *state of the art* dalam klasifikasi citra penyakit. Namun, penerapannya pada kasus spesifik *Lumpy Skin Disease* pada sapi masih memerlukan pengembangan lebih lanjut. Oleh karena itu, penelitian ini berupaya mengisi kesenjangan tersebut dengan mengembangkan model deteksi berbasis citra menggunakan ResNet50 guna meningkatkan akurasi dan efisiensi diagnosis penyakit pada ternak sapi.

2. Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimental di bidang ilmu komputer (*deep learning*). Tahapan penelitian meliputi pengumpulan data, prapemrosesan (augmentasi), perancangan model ResNet-50, pelatihan model (*training*), pengujian (*testing*), dan evaluasi kinerja. Eksperimen dilakukan menggunakan platform *Google Colaboratory* dengan akselerasi GPU dan pustaka TensorFlow/Keras. Rangkaian proses tahapan penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Metode Penelitian

2.1. Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini memiliki ketidakseimbangan kelas (*class imbalance*), di mana jumlah citra kulit sapi sehat (700 citra) lebih banyak dibandingkan dengan citra yang terindikasi *Lumpy Skin Disease* (LSD) sebanyak 324 citra. Ketidakseimbangan ini berpotensi menyebabkan model cenderung bias terhadap kelas mayoritas, sehingga performa dalam mendeteksi kelas minoritas menjadi kurang optimal. Oleh karena itu, diperlukan teknik penanganan imbalance dataset untuk meningkatkan kemampuan generalisasi model, khususnya dalam mengidentifikasi penyakit LSD.

2.2 Preprocessing dan Augmentasi Data

Prapemrosesan dilakukan dengan mengubah ukuran citra (*resize*) menjadi 224×224 piksel agar sesuai dengan standar *input* arsitektur ResNet-50. Nilai intensitas piksel dinormalisasi menggunakan fungsi *preprocess_input* dari Keras. Untuk meningkatkan variasi data dan mencegah *overfitting*, dilakukan augmentasi data pada set pelatihan menggunakan *ImageDataGenerator*. Pemilihan parameter augmentasi didasarkan pada karakteristik citra kulit sapi serta kondisi pengambilan gambar di lapangan yang bervariasi. Parameter *rotation_range* sebesar 25° digunakan untuk mensimulasikan variasi sudut pengambilan gambar, sementara *shear_range* (0.2) dan *zoom_range* (0.15) bertujuan merepresentasikan distorsi ringan dan variasi jarak objek terhadap kamera. Penyesuaian pencahayaan melalui *brightness_range* ([0.8, 1.2]) dilakukan untuk mengakomodasi perbedaan kondisi iluminasi. Selain itu, *width_shift_range* dan *height_shift_range* masing-masing sebesar 0.1 digunakan untuk mensimulasikan pergeseran posisi objek dalam citra. Penerapan horizontal dan vertical *flip* dilakukan untuk meningkatkan keragaman data dengan asumsi bahwa pola tekstur kulit tetap relevan terhadap transformasi orientasi. Nilai parameter dipilih dalam rentang moderat untuk menjaga agar hasil augmentasi tetap realistis dan tidak mengubah karakteristik utama pola penyakit. Sebanyak 15% dari data latih disisihkan sebagai data validasi selama proses pelatihan.

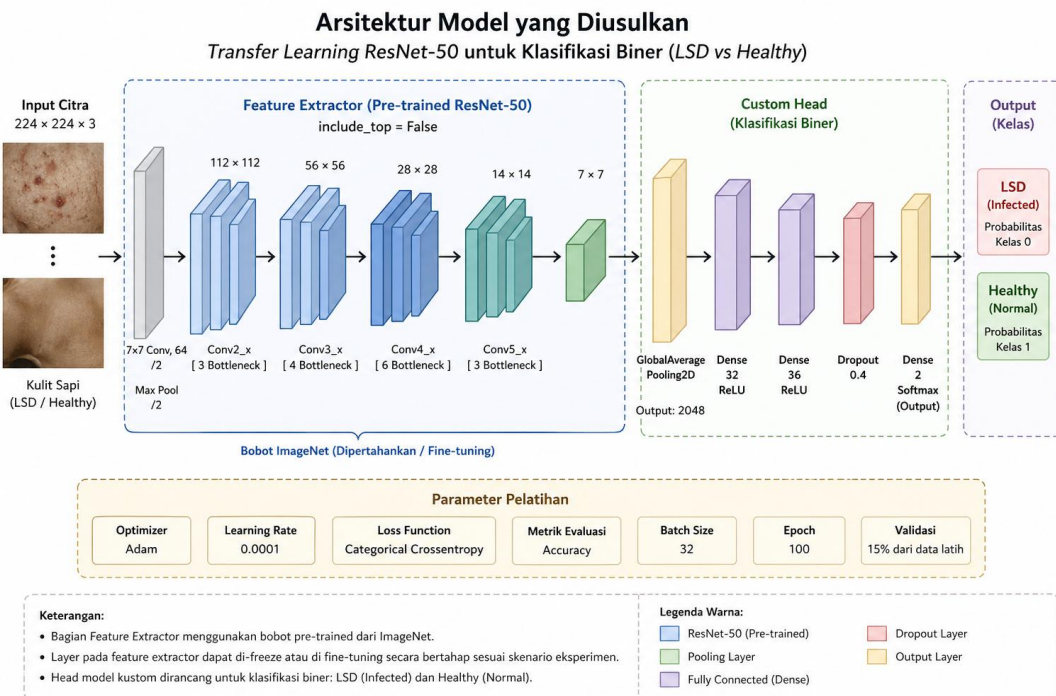
2.3 Arsitektur Model (ResNet-50)

Penelitian ini menggunakan konsep *Transfer Learning* dari model dasar ResNet-50 yang telah dilatih sebelumnya pada dataset *ImageNet*. Lapis klasifikasi asli dari ResNet-50 dibuang (*include_top=False*), dan diganti dengan *head* model kustom yang dirancang untuk klasifikasi *biner*. Secara arsitektural, model yang diusulkan terdiri dari bagian *feature extractor* (ResNet-50) yang diikuti oleh *classifier head* tambahan. Visualisasi arsitektur model ditunjukkan dalam bentuk diagram pada Gambar X, yang menggambarkan alur proses mulai dari input citra berukuran 224×224 piksel, ekstraksi fitur oleh ResNet-50, hingga proses klasifikasi oleh lapisan tambahan.

Struktur head model yang ditambahkan meliputi:

1. Lapisan *GlobalAveragePooling2D* untuk mereduksi dimensi spasial dan mengurangi risiko *overfitting*.
2. Lapisan *Dense* dengan 32 neuron dan fungsi aktivasi ReLU.
3. Lapisan *Dense* dengan 16 neuron dan fungsi aktivasi ReLU.
4. Lapisan *Dropout* dengan nilai 0.4 untuk meningkatkan generalisasi model.
5. Lapisan *Dense (output)* dengan 2 neuron menggunakan fungsi aktivasi *Softmax* untuk memprediksi probabilitas kelas (LSD atau *Healthy*).

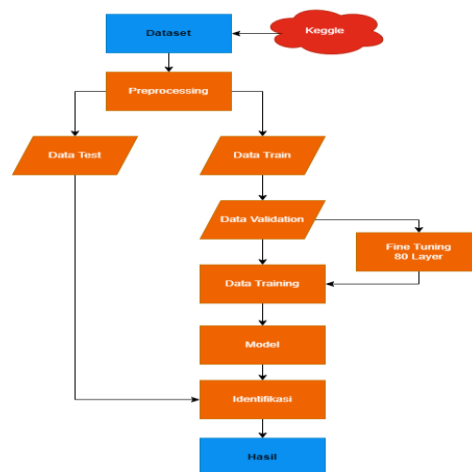
Dalam proses pelatihan, model dikompilasi menggunakan *optimizer* Adam dengan *learning rate* sebesar 0.0001. Fungsi *loss* yang digunakan adalah *categorical_crossentropy*, mengingat permasalahan yang dihadapi merupakan klasifikasi biner dengan representasi label dalam bentuk kategorikal. Selain itu, metrik evaluasi yang digunakan adalah *accuracy* untuk memantau performa model selama pelatihan dan validasi, sesuai dengan Gambar 2.



Gambar 2. Arsitektur model klasifikasi citra kulit sapi menggunakan pendekatan *Transfer Learning* berbasis ResNet-50.

2.4 Perancangan

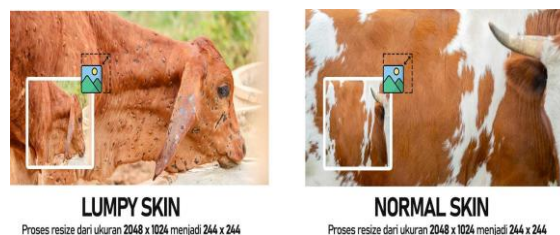
Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur ResNet50. Proses diawali dengan pengumpulan dataset citra kulit sapi, kemudian dilakukan *preprocessing* dengan mengubah ukuran citra menjadi 224×224 piksel sesuai standar *input* ResNet50. Selanjutnya, data diklasifikasikan menjadi dua kategori, yaitu kulit sapi sehat dan terinfeksi *Lumpy Skin Disease* (LSD). Untuk meningkatkan variasi data, dilakukan augmentasi seperti rotasi, *flip*, *zoom*, dan perubahan pencahayaan, yang bertujuan meningkatkan kemampuan generalisasi model.



Gambar 3. Diagram Alir Penelitian

Tahap berikutnya adalah pelatihan model menggunakan data *training* dan validasi. Pada awalnya, seluruh layer ResNet50 dibekukan dan dilatih selama 20 epoch, kemudian dilakukan *fine tuning* dengan membuka sebagian layer dan melanjutkan pelatihan selama 40 epoch. Model dikembangkan dengan penambahan *hidden layer* dan

dropout untuk mengurangi *overfitting*, serta menggunakan fungsi aktivasi ReLU dan *softmax* untuk klasifikasi dua kelas. Model akhir disimpan dalam format .h5 dan digunakan untuk mengklasifikasikan data uji sesuai tujuan penelitian.



Gambar 4. *Preprocessing* Citra kulit sapi

2.5. Implementasi

Pada tahap ini dilakukan implementasi algoritma *Convolutional Neural Network* (CNN) dengan arsitektur ResNet-50 sesuai dengan rancangan yang telah ditentukan. Model dilatih menggunakan data citra yang telah melalui tahap prapemrosesan dan augmentasi, kemudian diuji menggunakan data uji untuk mengevaluasi kinerjanya dalam mengklasifikasikan citra kulit sapi ke dalam dua kelas, yaitu *healthy* dan *infected (Lumpy Skin Disease)*. Proses implementasi dilakukan menggunakan bahasa pemrograman *Python* dengan pustaka *deep learning* seperti *TensorFlow* dan *Keras*. Model dilatih dengan parameter yang telah ditentukan, dan hasil pelatihan dievaluasi menggunakan beberapa metrik kinerja untuk mengukur kemampuan klasifikasi secara lebih komprehensif. Evaluasi model tidak hanya menggunakan akurasi, tetapi juga melibatkan *confusion matrix* yang terdiri dari *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Berdasarkan nilai tersebut, metrik evaluasi dihitung menggunakan persamaan berikut:

Akurasi (Accuracy) :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Presisi (Precision) :

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

Recall (Sensitivity) :

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

F1-Score :

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Penggunaan metrik tambahan seperti *precision*, *recall*, dan *F1-score* bertujuan untuk memberikan evaluasi yang lebih akurat, terutama dalam kondisi dataset yang tidak seimbang (*imbalanced dataset*), sehingga performa model dalam mendeteksi kelas minoritas (LSD) dapat dianalisis secara lebih mendalam.

2.6 Pengujian

Pada tahap ini dilakukan pengujian terhadap data testing menggunakan model yang telah dibangun dengan arsitektur ResNet-50. Sistem akan mengidentifikasi citra kulit sapi yang diuji, kemudian hasil klasifikasi dikumpulkan untuk dianalisis. Evaluasi performa model dilakukan dengan menghitung metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-Score*, guna menilai seberapa baik model dalam membedakan antara kulit sapi yang sehat dan yang terinfeksi penyakit *Lumpy Skin Disease* (LSD).

2.7 Analisis dan Hasil Pengujian

Pada tahap ini dilakukan analisis terhadap hasil pengujian untuk mengevaluasi tingkat keberhasilan metode yang digunakan. Evaluasi dilakukan dengan menggunakan *Confusion Matrix* untuk menghitung metrik evaluasi utama, yaitu *precision*, *recall*, *accuracy*, dan *F1-Score*.

Precision mengukur proporsi prediksi positif yang benar, yaitu jumlah citra yang benar-benar terklasifikasi sebagai terinfeksi dibagi dengan total citra yang diprediksi sebagai terinfeksi. *Recall* menunjukkan seberapa banyak citra yang benar-benar terinfeksi berhasil terdeteksi oleh model, yaitu jumlah citra terinfeksi yang terklasifikasi dengan benar dibandingkan dengan seluruh citra terinfeksi yang ada. *Accuracy* merupakan rasio dari jumlah seluruh prediksi yang benar dibandingkan dengan total jumlah prediksi yang dilakukan. Sementara itu, *F1-Score* dihitung sebagai rata-rata harmonik antara *precision* dan *recall*, yang memberikan gambaran seimbang antara keduanya terutama saat terjadi ketidakseimbangan kelas.

3. Hasil dan Pembahasan

Bab ini menyajikan hasil dari implementasi dan pengujian model *Convolutional Neural Network* (CNN) dengan arsitektur ResNet-50 untuk klasifikasi citra penyakit kulit *Lumpy Skin Disease* (LSD) pada ternak sapi. Hasil pengujian tidak hanya disajikan dalam bentuk nilai metrik evaluasi, tetapi juga divisualisasikan melalui *confusion matrix* serta grafik proses pelatihan (*training*) dan validasi (*validation*) untuk memberikan analisis yang lebih komprehensif.

Confusion matrix digunakan untuk menggambarkan performa model dalam mengklasifikasikan data uji ke dalam kelas yang benar dan salah. Matriks ini terdiri dari empat komponen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Visualisasi *confusion matrix* ditampilkan pada Gambar X, yang menunjukkan distribusi prediksi model terhadap masing-masing kelas (LSD dan *Healthy*). Berdasarkan matriks tersebut, dapat dianalisis kemampuan model dalam mendeteksi kelas minoritas (LSD) serta mengidentifikasi kesalahan klasifikasi yang terjadi.

Selain itu, proses pelatihan model divisualisasikan menggunakan grafik *training* dan *validation* yang meliputi nilai akurasi dan loss pada setiap epoch. Grafik ini ditampilkan pada Gambar Y, yang menunjukkan tren pembelajaran model selama proses pelatihan. Analisis terhadap grafik tersebut penting untuk mengidentifikasi apakah model mengalami *overfitting* atau *underfitting*. Sebagai contoh, apabila kurva akurasi training meningkat secara signifikan sementara akurasi validasi cenderung stagnan atau menurun, maka model mengindikasikan terjadinya *overfitting*.

Dengan menggabungkan analisis *confusion matrix* dan grafik pelatihan, penelitian ini memberikan evaluasi yang lebih menyeluruh terhadap performa model, tidak hanya dari sisi akurasi, tetapi juga dari kemampuan generalisasi model dalam mengenali pola penyakit pada citra kulit sapi.

3.1 Lingkungan Eksperimen dan Persiapan Implementasi

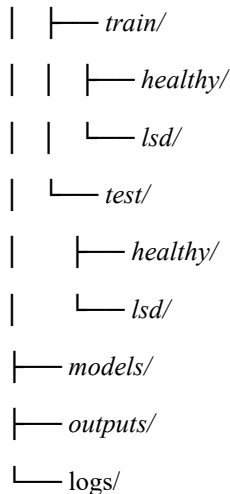
3.1.1 Platform Eksperimen

Seluruh proses implementasi dan pelatihan model dilakukan menggunakan *Google Colaboratory* (Colab) yang terintegrasi dengan *Google Drive* sebagai media penyimpanan utama. *Google Colab* dipilih karena menyediakan sumber daya komputasi berbasis GPU (*Graphics Processing Unit*) yang dapat mempercepat proses pelatihan model *deep learning*, khususnya *Convolutional Neural Network* (CNN). Konfigurasi Colab diatur melalui menu *Runtime* → *Change runtime type* dengan memilih *Hardware accelerator* = GPU. Penggunaan GPU secara signifikan menurunkan waktu komputasi dan memungkinkan model ResNet-50 untuk dilatih dalam waktu yang efisien.

3.1.2 Struktur Direktori Proyek di *Google Drive*

Agar proses penyimpanan model, dataset, dan hasil pelatihan terorganisasi dengan baik, proyek ini menggunakan struktur folder pada *Google Drive* seperti berikut:

```
MyDrive/  
├── ResNet50_LSD_Project/  
│   └── data/
```



Folder data dibagi menjadi dua subfolder utama, yaitu *train* dan *test*, yang masing-masing berisi dua kelas, yaitu *healthy* dan *lsd*. Setiap kelas terdiri atas kumpulan citra yang merepresentasikan kondisi terinfeksi LSD dan normal (*healthy*). Folder *models* diperuntukkan untuk penyimpanan model-model yang telah dilatih, disimpan dalam format *.h5*. Untuk keperluan dokumentasi dan analisis, folder *outputs* digunakan menyimpan berbagai grafik, hasil prediksi, dan visualisasi lainnya. Terakhir, folder *logs* berfungsi mencatat dan menyimpan *log* pelatihan, yang dapat dimanfaatkan untuk melacak proses pelatihan, misalnya melalui TensorBoard.

3.1.3 Proses *mounting google drive* dan persiapan lingkungan

Sebelum proses pelatihan dimulai, *Google Drive* di-*mount* agar Colab dapat mengakses seluruh *file* dan direktori proyek. Proses *mounting* dilakukan menggunakan perintah berikut:

```
from google.colab import drive
drive.mount('/content/drive')

# Path proyek
PROJECT_DIR = "/content/drive/MyDrive/ResNet50_LSD_Project"
DATA_DIR_TRAIN = f"{PROJECT_DIR}/data/train"
DATA_DIR_TEST = f"{PROJECT_DIR}/data/test"
OUT_DIR = f"{PROJECT_DIR}/outputs"
MODEL_DIR = f"{PROJECT_DIR}/models"
LOG_DIR = f"{PROJECT_DIR}/logs"

# Membuat direktori jika belum ada
import os
for d in [PROJECT_DIR, OUT_DIR, MODEL_DIR, LOG_DIR]:
    os.makedirs(d, exist_ok=True)
```

Langkah di atas memastikan Colab memiliki akses baca-tulis penuh terhadap *Drive*, sehingga seluruh artefak penelitian (model, *weights*, grafik hasil, log, dan *confusion matrix*) otomatis tersimpan secara permanen.

Selain itu, dilakukan pemeriksaan versi *library Python* dan *TensorFlow* untuk memastikan kompatibilitas lingkungan dengan perintah berikut :

```
!python -V
!pip show tensorflow
```

Hasil pemeriksaan menunjukkan bahwa penelitian ini menggunakan *Python* versi 3.12.12 dan *TensorFlow* versi 2.19.0, sesuai dengan kebutuhan implementasi model ResNet-50 berbasis Keras API.

3.1.4 Reprodusibilitas dan Keamanan Data

Pemanfaatan *Google Drive* sebagai repositori utama memiliki beberapa keuntungan yaitu sebagai berikut.

- a. Keamanan dan *Backup* Otomatis, setiap hasil eksperimen langsung tersimpan di *cloud* tanpa risiko kehilangan data akibat batas sesi Colab.

- b. *Reproducibility*, struktur folder yang konsisten memudahkan pengulangan eksperimen dengan parameter yang sama.
- c. Kolaborasi, seluruh tim peneliti dapat mengakses *notebook* dan hasil model secara sinkron melalui *Drive*.

Untuk menjamin hasil yang dapat direproduksi, seluruh proses pelatihan dilengkapi dengan seed random tetap pada *NumPy* dan *TensorFlow*:

```
import numpy as np, tensorflow as tf
np.random.seed(42)
tf.random.set_seed(42)
```

Langkah ini menjamin bahwa setiap kali pelatihan diulang, hasil yang diperoleh tetap konsisten meskipun dilakukan pada sesi Colab yang berbeda.

3.2 Pengambilan dan Persiapan Dataset

3.2.1 Sumber Dataset

Dataset yang digunakan dalam penelitian ini bersumber dari platform Kaggle, dengan URL resmi: <https://www.kaggle.com/datasets/warcoder/lumpy-skin-images-dataset>. Dataset ini terdiri dari 324 citra penyakit kulit *Lumpy Skin Disease* (LSD) dan 700 citra normal (*healthy*). Dataset diunduh langsung melalui *Google Colaboratory* menggunakan Kaggle API.

Langkah awal yang dilakukan adalah mengunggah file autentikasi *kaggle.json* yang berisi API token, ke direktori kerja Colab. Token ini memungkinkan akses aman ke akun Kaggle dan proses pengunduhan dataset secara langsung ke ruang kerja *Colab*. Berikut potongan kode yang digunakan untuk proses pengunduhan:

```
import os, shutil, stat

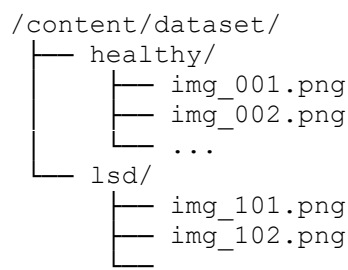
# Menyalin file autentikasi kaggle.json
KAGGLE_DIR = "/root/.kaggle"
os.makedirs(KAGGLE_DIR, exist_ok=True)
shutil.copy("/content/kaggle.json", f"{KAGGLE_DIR}/kaggle.json")
os.chmod(f"{KAGGLE_DIR}/kaggle.json", stat.S_IRUSR | stat.S_IWUSR)

# Mengunduh dataset dari Kaggle
!kaggle datasets download -d warcoder/lumpy-skin-images-dataset -p
/content/dataset --unzip
```

Perintah di atas secara otomatis mengekstrak dataset ke direktori `/content/dataset`, yang berisi dua subfolder utama, yaitu *healthy* dan *lsd*. Masing-masing folder memuat citra yang telah dilabeli sesuai kategori dalam format PNG.

3.2.2 Struktur Dataset Setelah Ekstraksi

Setelah proses ekstraksi selesai, struktur folder dataset menjadi sebagai berikut:



Dataset ini kemudian dipindahkan ke *Google Drive* agar tersimpan permanen menggunakan perintah:

```
import shutil, os
DATA_DIR_TRAIN = f"{PROJECT_DIR}/data/train"
DATA_DIR_TEST = f"{PROJECT_DIR}/data/test"
os.makedirs(DATA_DIR_TRAIN, exist_ok=True)
os.makedirs(DATA_DIR_TEST, exist_ok=True)
```

Langkah ini memastikan bahwa dataset tetap tersimpan meskipun sesi Colab berakhir.

3.2.3 Pembagian Data *Train* dan *Test*

Agar model dapat dilatih dan diuji secara objektif, dataset dibagi menjadi dua bagian, yaitu 80% untuk data pelatihan (*training*) dan 20% untuk data pengujian (*testing*). Proses pembagian dilakukan secara acak (*randomized split*) agar tidak terjadi bias akibat urutan *file*. Berikut potongan kode pembagian data:

```
import shutil, os
os.makedirs(DATA_DIR_TRAIN, exist_ok=True)
os.makedirs(DATA_DIR_TEST, exist_ok=True)

# Bagi data train/test (jika dataset belum terbagi). Contoh split 80/20:
import glob, random
random.seed(42)

def split_copy(src_class_dir, train_dest, test_dest, split=0.8):
    paths = glob.glob(os.path.join(src_class_dir, "**"))
    random.shuffle(paths)
    n = int(len(paths)*split)
    train, test = paths[:n], paths[n:]
    for p in train:
        os.makedirs(os.path.join(train_dest,
os.path.basename(src_class_dir)), exist_ok=True)
        shutil.copy(p, os.path.join(train_dest,
os.path.basename(src_class_dir)))
    for p in test:
        os.makedirs(os.path.join(test_dest,
os.path.basename(src_class_dir)), exist_ok=True)
        shutil.copy(p, os.path.join(test_dest,
os.path.basename(src_class_dir)))

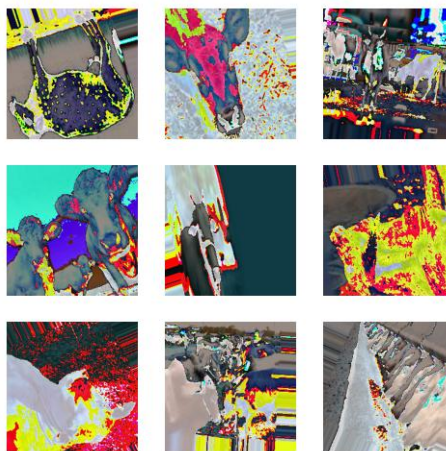
split_copy("/content/ds/healthy", f"{DATA_DIR_TRAIN}", f"{DATA_DIR_TEST}")
split_copy("/content/ds/lsd", f"{DATA_DIR_TRAIN}", f"{DATA_DIR_TEST}")
```

3.3 *Preprocessing* dan Augmentasi Data

Tahap *preprocessing* merupakan langkah penting sebelum data citra digunakan dalam pelatihan model *Convolutional Neural Network* (CNN). Tujuan utamanya adalah menyiapkan data agar sesuai dengan format input arsitektur ResNet-50, sekaligus mengurangi variasi yang tidak relevan dari citra asli.

Dalam penelitian ini, setiap citra terlebih dahulu diubah ukurannya menjadi 224×224 piksel, menyesuaikan dengan dimensi *input* ResNet-50 yang dilatih menggunakan dataset *ImageNet*. Proses ini dilakukan menggunakan pustaka *ImageDataGenerator* dari Keras, yang secara otomatis menyesuaikan ukuran citra saat dimuat dari direktori dataset.

Selain itu, dilakukan pula normalisasi intensitas piksel agar nilai setiap piksel berada dalam rentang 0–1. Normalisasi ini penting karena mempercepat proses konvergensi selama pelatihan dan mencegah dominasi nilai besar yang dapat mengganggu perhitungan gradien. Fungsi *preprocess_input* dari modul *tensorflow.keras.applications.resnet50* digunakan untuk menstandarkan nilai piksel sesuai skala yang digunakan dalam *pre-trained* model *ImageNet*.



Gambar 5. Contoh Hasil Augmentasi Data (224 x 224)

3.4 Perancangan dan Pembangunan Arsitektur Model ResNet-50

3.4.1 Pemilihan Arsitektur ResNet-50

Penelitian ini menggunakan arsitektur CNN ResNet-50, yang dikenal unggul dalam pengenalan citra karena mampu mengatasi masalah *vanishing gradient* melalui mekanisme residual *learning*. Keunggulan ResNet-50 terletak pada kemampuannya membangun jaringan yang lebih dalam tanpa menurunkan akurasi serta mempercepat konvergensi. Dalam penelitian ini, ResNet-50 dipilih karena mampu mengekstraksi fitur kompleks pada citra kulit sapi yang memiliki variasi tekstur dan warna, sehingga efektif untuk mendeteksi penyakit *Lumpy Skin Disease* (LSD).

3.4.2 Konsep *Transfer Learning*

Penelitian ini menggunakan *transfer learning* pada ResNet-50 untuk meningkatkan akurasi dan efisiensi pelatihan dengan dataset terbatas. Model memanfaatkan bobot pra-latih dari *ImageNet* sebagai *feature extractor* untuk mengenali pola dasar seperti tepi dan tekstur, sementara lapisan akhir dilatih ulang untuk mengklasifikasikan dua kelas, yaitu *healthy* dan LSD.

3.5 Tahap-1 Pelatihan (*Freeze Semua Layer Backbone*)

3.5.1 Strategi Pelatihan Tahap Awal

Pada tahap awal, seluruh lapisan dari model ResNet-50 dibekukan (*frozen*) sehingga bobot pra-latih (*pre-trained weights*) dari *ImageNet* tidak diperbarui. Tujuan strategi ini adalah menjaga kemampuan dasar model dalam mengekstraksi fitur umum seperti tepi dan tekstur, sembari melatih hanya lapisan tambahan (*Dense 32-16 + Dropout 0.4*) yang baru ditambahkan. Pelatihan dilakukan selama 20 epoch menggunakan optimizer Adam dan *loss function categorical_crossentropy* dengan *batch size = 32*.

3.5.2 Performa Selama Pelatihan

Hasil pelatihan menunjukkan bahwa model mengalami peningkatan kinerja yang konsisten dari awal hingga akhir proses:

Tabel 1. Hasil Performa Selama Pelatihan

<i>Epoch</i>	<i>Training Accuracy</i>	<i>Validation Accuracy</i>	<i>Training Loss</i>	<i>Validation Loss</i>
1	0.6047	0.6967	0.7327	0.5528
5	0.8737	0.8934	0.3461	0.2477
10	0.9079	0.9344	0.2647	0.1986
15	0.9062	0.9098	0.2719	0.2178
20	0.9234	0.9344	0.2395	0.1917

Secara keseluruhan, akurasi pelatihan meningkat dari 60,47 % menjadi 92,34 %, sedangkan akurasi validasi naik dari 69,67 % menjadi 93,44 %. Nilai loss pada kedua set data menurun stabil dari 0.73 → 0.23 (training) dan 0.55 → 0.19 (*validation*), menunjukkan proses konvergensi yang baik tanpa gejala *overfitting*. Visualisasi hasil menampilkan dua kurva yang berimpit dengan tren naik stabil—indikasi bahwa model belajar secara efektif dari data latih dan mampu menggeneralisasi pada data validasi.

3.6 Hasil dan Analisis Evaluasi

Evaluasi menghasilkan performa sebagai berikut:

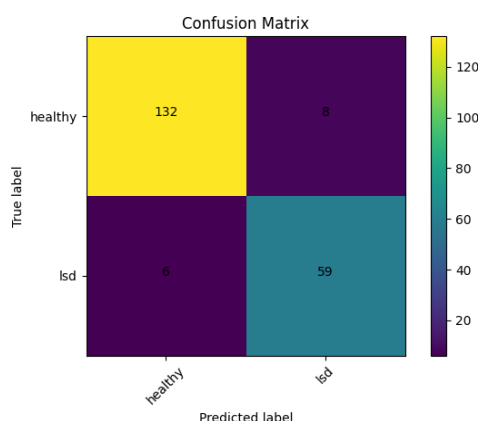
Tabel 2. Hasil dan Analisis Evaluasi

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<i>Healthy</i>	0.9565	0.9429	0.9496	140
LSD	0.8806	0.9077	0.8939	65
<i>Accuracy</i>			0.9317	205
<i>Macro Avg</i>	0.9186	0.9253	0.9218	205
<i>Weighted Avg</i>	0.9324	0.9317	0.9320	205

Hasil tersebut menunjukkan bahwa model mencapai akurasi keseluruhan 93,17% dengan nilai *F1-score* makro sebesar 0,92, menandakan keseimbangan yang baik antara ketepatan dan sensitivitas. Kelas *healthy* memiliki performa lebih tinggi dibandingkan LSD, karena citra kulit sehat relatif homogen dengan tekstur seragam. Sebaliknya, variasi tingkat keparahan dan pencahayaan pada citra LSD menyebabkan sedikit penurunan pada *precision*

3.6.1 Analisis Confusion Matrix

Confusion matrix pada Gambar 6 menggambarkan distribusi hasil prediksi model:



Gambar 6. Analisis *Confusion Matrix*

Interpretasi hasil:

1. Model benar mengklasifikasikan 132 dari 140 citra healthy dan 59 dari 65 citra LSD.
2. Terdapat 6 false negative (citra LSD diklasifikasikan sebagai healthy) dan 8 false positive (citra healthy diklasifikasikan sebagai LSD).
3. Kesalahan prediksi terutama muncul pada citra dengan lesi samar, pencahayaan redup, atau perbedaan kontras kecil antara kulit sehat dan lesi, yang membuat pola tekstur sulit dikenali.

Meskipun demikian, tingkat *recall* LSD sebesar 0,9077 menandakan model memiliki sensitivitas tinggi, sehingga kecil kemungkinan penyakit LSD terlewat saat deteksi dini. Hal ini penting dalam konteks penerapan sistem cerdas untuk monitoring kesehatan ternak di lapangan.

4. Kesimpulan

Model CNN berbasis ResNet-50 yang dikembangkan dalam penelitian ini terbukti efektif dalam mengklasifikasikan citra penyakit *Lumpy Skin Disease* (LSD) pada ternak sapi. Pendekatan *transfer learning* dan *fine-tuning* mampu meningkatkan kemampuan model dalam mengekstraksi fitur dan melakukan klasifikasi secara akurat. Secara keseluruhan, model menunjukkan kemampuan generalisasi yang baik dan berpotensi digunakan sebagai alat bantu deteksi dini penyakit pada sektor peternakan.

Daftar Rujukan

- [1] Adebiyi, A., Abdalnabi, N., Smith, E. H., Hirner, J., Simões, E. J., Becevic, M., & Rao, P. (2024). Accurate Skin Lesion Classification Using Multimodal Learning on the HAM10000 Dataset. <https://doi.org/10.1101/2024.05.30.24308213>
- [2] Asra, R., & Irwan, Muh. (2022). Analisa Potensi Jerami Padi Sebagai Pakan Ternak Sapi Potong Melalui Pendekatan Sistem Informasi Geografis Di Kabupaten Sidenreng Rappang. *Jurnal Sains Dan Teknologi Industri Peternakan*, 2(1), 29–35. <https://doi.org/10.53678/jstip.v2i1.622>
- [3] Dondapati, R. D., Sivaprakasam, T., & Kumar, K. V. (2024). Dermatological Decision Support Systems Using CNN for Binary Classification. *Engineering Technology & Applied Science Research*, 14(3), 14240–14247. <https://doi.org/10.48084/etasr.7173>
- [4] Efendi, R. (2023). Penerapan Algoritma Convolutional Neural Network Arsitektur ResNet-50 untuk Klasifikasi Citra Daging Sapi dan Babi. *JURIKOM (Jurnal Riset Komputer)*, 10(3). Tersedia di: <https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom/article/view/4176>
- [5] Gambo, P., Maguda, A. S., Adole, J. A., Dyek, D. Y., Ifende, V. I., Bot, C. J., & Adedeji, A. (2018). A Survey of Viral Diseases of Livestock Characterized by Skin Lesions in Kanam Local Government Area of Plateau State, Nigeria. *Nigerian Veterinary Journal*, 39(3), 250–262. <https://doi.org/10.4314/nvj.v39i3.8>
- [6] Imran, M., Hashmi, A. H., Khalique, F., & Iqbal, M. Z. (2022). Lumpy Skin Disease Emerging Problem in Pakistan. <https://doi.org/10.21203/rs.3.rs-1904208/v1>
- [7] Sendow, I., Assadah, N. S., Ratnawati, A., Dharmayanti, N. I., & Saepulloh, M. (2021). Lumpy Skin Disease: Ancaman Penyakit Emerging Bagi Kesehatan Ternak Sapi Di Indonesia. *Indonesian Bulletin of Animal and Veterinary Sciences*, 31(2), 85. <https://doi.org/10.14334/wartazoa.v31i2.2739>
- [8] Aritonang, M. A. S., Abrar Masril, M., Chaniago, D., Marshall Al Karim, M., Mahani Cunis, V., & Surgiwe, S. (2023). Perancangan Sistem Pendeteksi Penyakit Pada Rumput Laut Dengan Menggunakan Metode Convolutional Neural Network. *Jurnal Pustaka AI (Pusat Akses Kajian Teknologi Artificial Intelligence)*, 5(2), 408–417. <https://doi.org/10.55382/jurnalpustakaai.v5i2.1160>
- [9] Sentoso, T., Ardiansyah, F., Tamuntuan, V., Wangsa, S. S., Kusriani, K., & Kusnawi, K. (2024). Identification of Lumpy Skin Disease in Cattle With Image Classification Using the Convolutional Neural Network Method. *Sistemasi*, 13(3), 864. <https://doi.org/10.32520/stmsi.v13i3.2569>
- [10] Sugiharto, B. H., Solekan, M., Muslimin, Moh., & Hamzali, S. (2023). Pengaruh Peran Manajemen Produksi Dan Faktor Lingkungan Dalam Meningkatkan Efisiensi Dan Keberlanjutan Usaha Peternakan Sapi Di Jawa Timur. *Jurnal Multidisiplin West Science*, 2(11). <https://doi.org/10.58812/jmws.v2i11.784>
- [11] Uddin, M. A., Islam, Md. A., Rahman, A. K. M. A., Rahman, M. M., Khasruzzaman, A. K. M., Ward, M. P., & Hossain, M. T. (2022). Epidemiological Investigation of Lumpy Skin Disease Outbreaks in Bangladeshi Cattle During 2019–2020. *Transboundary and Emerging Diseases*, 69(6), 3397–3404. <https://doi.org/10.1111/tbed.14696>
- [12] Ardiansyah, H., & Desyani, T. (2025). Transfer Learning Menggunakan Model VGG16 untuk Klasifikasi Citra Hewan. *Jurnal Pustaka AI*, 5(2), 441–448. <https://doi.org/10.55382/jurnalpustakaai.v5i2.1174>
- [13] Widianingrum, D. C., & Septio, R. W. (2023). Peran Peternakan Dalam Mendukung Ketahanan Pangan Indonesia: Kondisi, Potensi, Dan Peluang Pengembangan. *National Multidisciplinary Sciences*, 2(3), 285–291. <https://doi.org/10.32528/nms.v2i3.298>
- [14] Yusuf, M., & Utamingrum, F. (2024). Deteksi Fase Estrus pada Sapi Betina berbasis Raspberry Pi menggunakan Deep Learning dengan ResNet-50. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 8(2). Tersedia di: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/13303>
- [15] Faturohman, A., Anggreani, D., & Yusliana Bakt, R. (2025). Model Deep Learning Berbasis Convolutional Neural Network Untuk Identifikasi Stroke Iskemik Pada Citra CT Scan. *Jurnal Pustaka AI (Pusat Akses Kajian Teknologi Artificial Intelligence)*, 5(2), 290–297. <https://doi.org/10.55382/jurnalpustakaai.v5i2.1150>