

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Berikut rangkaian penelitian terdahulu di bawah ini menjadi referensi kunci yang melandasi penelitian ini:

Tabel 2. 1 Penelitian Terdahulu

No	Tema Penelitian	Metode	Hasil dan Keterkaitan
1	<i>Transfer learning</i> pada citra LSD	DenseNet121 untuk klasifikasi citra sapi terinfeksi <i>Lumpy Skin Disease</i>	Model DenseNet121 memperoleh <i>accuracy</i> sebesar 80,21%. Penelitian ini menunjukkan bahwa arsitektur <i>deep learning</i> dapat digunakan untuk identifikasi citra LSD, tetapi performanya masih dapat ditingkatkan melalui evaluasi model lain [14].
2	CNN konvensional dan <i>transfer learning</i>	<i>Custom CNN</i> dan <i>transfer learning</i> untuk klasifikasi citra LSD dan non-LSD	Model memperoleh <i>accuracy</i> sebesar 86,54% pada dataset sebanyak 1.023 citra sapi. Penelitian ini menunjukkan bahwa CNN dapat digunakan untuk klasifikasi LSD, tetapi performanya masih memiliki ruang peningkatan [12].
3	<i>Machine learning</i> non-citra	Random Forest dan LightGBM berbasis fitur untuk prediksi <i>Lumpy Skin Disease</i>	Model memperoleh <i>F1-score</i> sebesar 98%. Penelitian ini menggunakan pendekatan <i>machine learning</i> klasik berbasis fitur, sehingga berbeda dengan penelitian berbasis klasifikasi citra menggunakan <i>transfer learning</i> [15].
4	CNN sebagai <i>feature extractor</i>	CNN sebagai <i>feature extractor</i> dan Extreme Learning Machine sebagai <i>classifier</i>	Model memperoleh <i>accuracy</i> sebesar 90,12%. Penelitian ini menunjukkan bahwa ekstraksi fitur berbasis CNN dapat dikombinasikan dengan classifier lain untuk mendeteksi area berisiko tinggi LSD [16].

No	Tema Penelitian	Metode	Hasil dan Keterkaitan
5	Perbandingan beberapa model <i>transfer learning</i>	DenseNet121, ResNet50V2, InceptionV3, Xception, dan MobileNetV2	DenseNet121 memperoleh <i>accuracy</i> tertinggi sebesar 90,36%. Penelitian ini menunjukkan bahwa performa setiap arsitektur CNN berbeda, sehingga diperlukan evaluasi model dalam skenario yang seragam [11].
6	Perbandingan model pretrained	VGG16, MobileNetV2, InceptionV3, ResNet50, Xception, dan beberapa model pretrained lainnya	MobileNetV2 memperoleh <i>accuracy</i> sebesar 96,39%, sedangkan VGG16 memperoleh <i>accuracy</i> sebesar 96,07%. Penelitian ini menunjukkan bahwa model pretrained dapat memberikan performa tinggi pada deteksi LSD [10].
7	Optimasi model CNN/pretrained	MobileNetV2 dengan optimasi RMSProp	Model memperoleh <i>accuracy</i> sebesar 95%. Penelitian ini menunjukkan bahwa pemilihan arsitektur dan optimizer dapat memengaruhi performa model dalam klasifikasi citra LSD [9].
8	EfficientNet dan ResNet pada citra LSD	ResNet50 dan EfficientNetB0	EfficientNetB0 memperoleh <i>accuracy</i> sebesar 93,57%, sedangkan ResNet50 memperoleh <i>accuracy</i> sebesar 76,43%. Penelitian ini menunjukkan bahwa arsitektur berbeda dapat menghasilkan performa yang signifikan berbeda pada kasus klasifikasi LSD [17].
9	<i>Hybrid feature extraction</i>	InceptionV3 sebagai <i>feature extractor</i> dan SVM sebagai <i>classifier</i>	Model memperoleh <i>accuracy</i> sebesar 84%, <i>precision</i> sebesar 80%, <i>recall</i> sebesar 83%, dan <i>F1-score</i> sebesar 82%. Penelitian ini menunjukkan bahwa fitur dari InceptionV3 dapat digunakan bersama SVM untuk klasifikasi citra LSD [18].
10	Optimasi CNN melalui <i>hyperparameter tuning</i>	CNN dengan <i>hyperparameter tuning</i> untuk klasifikasi LSD pada sapi perah	Model memperoleh <i>validation accuracy</i> sebesar 89,73%, <i>testing accuracy</i> sebesar 80,68%, dan AUC sebesar 0,88. Penelitian ini relevan karena menunjukkan pentingnya pengaturan <i>hyperparameter</i> dan evaluasi generalisasi model [19].

Berdasarkan Tabel 2.1 Penelitian Terdahulu, penelitian terdahulu terkait deteksi *Lumpy Skin Disease* dapat dikelompokkan ke dalam beberapa tema utama, yaitu CNN konvensional, *transfer learning*, pendekatan hybrid berbasis *feature extraction*, optimasi model, dan *machine learning* non-citra. Pengelompokan ini dilakukan agar pembahasan penelitian terdahulu tidak hanya menjelaskan artikel secara terpisah, tetapi juga menunjukkan pola metode, kontribusi, dan keterbatasan dari masing-masing pendekatan.

Kelompok pertama adalah penelitian yang menggunakan CNN konvensional dan *deep learning* dasar untuk klasifikasi citra LSD. Pendekatan ini menunjukkan bahwa CNN mampu mempelajari pola visual seperti tekstur kulit, nodul, lesi, dan perubahan permukaan kulit sapi. Penelitian dengan DenseNet121 memperoleh *accuracy* sebesar 80,21%, sedangkan penelitian berbasis *custom CNN* memperoleh *accuracy* sebesar 86,54% pada dataset 1.023 citra sapi [12], [14]. Hasil tersebut menunjukkan bahwa CNN dapat digunakan untuk identifikasi citra LSD, tetapi performanya masih memiliki ruang peningkatan, terutama apabila dataset terbatas dan variasi citra cukup tinggi.

Kelompok kedua adalah penelitian yang menggunakan model pretrained berbasis *transfer learning*. Beberapa penelitian telah menggunakan arsitektur seperti DenseNet121, ResNet50V2, InceptionV3, Xception, MobileNetV2, VGG16, dan EfficientNetB0 untuk klasifikasi citra LSD [10], [11], [17]. Hasil penelitian menunjukkan bahwa model pretrained dapat menghasilkan performa yang kompetitif, seperti DenseNet121 dengan *accuracy* sebesar 90,36%, MobileNetV2 sebesar 96,39%, VGG16 sebesar 96,07%, dan EfficientNetB0 sebesar 93,57% [10], [11], [17]. Perbedaan hasil tersebut menunjukkan bahwa setiap arsitektur memiliki kemampuan ekstraksi fitur yang berbeda, sehingga pemilihan model perlu mempertimbangkan karakteristik arsitektur, konfigurasi pelatihan, serta skenario evaluasi yang digunakan.

Kelompok ketiga adalah penelitian yang menggunakan pendekatan hybrid berbasis *feature extraction*. Pada pendekatan ini, CNN atau model pretrained tidak selalu digunakan sebagai classifier utama, tetapi dimanfaatkan untuk mengekstraksi

fitur visual yang kemudian diklasifikasikan menggunakan algoritma lain. Penelitian yang menggunakan CNN sebagai *feature extractor* dan Extreme Learning Machine sebagai *classifier* memperoleh *accuracy* sebesar 90,12% [16]. Sementara itu, penelitian lain menggunakan InceptionV3 sebagai *feature extractor* dan Support Vector Machine sebagai *classifier*, dengan hasil *accuracy* sebesar 84%, *precision* sebesar 80%, *recall* sebesar 83%, dan *F1-score* sebesar 82% [18]. Hasil tersebut menunjukkan bahwa fitur visual hasil ekstraksi CNN dapat digunakan bersama classifier lain, tetapi pendekatan ini belum sepenuhnya menunjukkan adaptasi model secara end-to-end melalui *fine-tuning*.

Kelompok keempat adalah penelitian yang berfokus pada optimasi model. Penelitian dengan MobileNetV2 dan optimizer RMSProp memperoleh *accuracy* sebesar 95%, yang menunjukkan bahwa pemilihan arsitektur dan optimizer dapat memengaruhi performa model dalam klasifikasi citra LSD [9]. Selain itu, penelitian yang menggunakan CNN dengan *hyperparameter tuning* memperoleh *validation accuracy* sebesar 89,73%, *testing accuracy* sebesar 80,68%, dan AUC sebesar 0,88 [19]. Hasil tersebut menunjukkan bahwa performa model tidak hanya dipengaruhi oleh jenis arsitektur, tetapi juga oleh konfigurasi pelatihan, seperti optimizer, *learning rate*, jumlah epoch, dan strategi evaluasi. Hal ini relevan dengan penelitian ini karena model *transfer learning* tidak hanya digunakan secara langsung, tetapi juga dioptimasi melalui *feature extraction*, *fine-tuning*, *class weight*, *early stopping*, dan *model checkpoint*.

Kelompok kelima adalah penelitian yang menggunakan pendekatan *machine learning* non-citra atau berbasis fitur. Salah satu penelitian menggunakan Random Forest dan LightGBM untuk prediksi LSD dan memperoleh *F1-score* sebesar 98% [15]. Meskipun hasil tersebut menunjukkan performa yang tinggi, pendekatan tersebut berbeda dengan penelitian ini karena tidak menggunakan citra digital sebagai input utama dan tidak memanfaatkan CNN untuk mempelajari fitur visual secara otomatis. Dengan demikian, penelitian berbasis *machine learning* non-citra tetap relevan sebagai pembanding pendekatan, tetapi tidak secara langsung menjawab kebutuhan klasifikasi citra LSD berbasis *computer vision*.

Pemilihan DenseNet121, EfficientNetV2S, InceptionV3, ResNet50, dan Xception dalam penelitian ini didasarkan pada dua pertimbangan utama, yaitu keterkaitan dengan penelitian terdahulu dan perbedaan karakteristik arsitektur. DenseNet121 dipilih karena merepresentasikan pendekatan *dense connectivity* yang mampu mempertahankan aliran informasi fitur dari lapisan awal hingga akhir. EfficientNetV2S dipilih karena merepresentasikan arsitektur efisien yang dirancang untuk menyeimbangkan performa dan kompleksitas model. InceptionV3 dipilih karena memiliki kemampuan ekstraksi fitur multi-skala yang relevan untuk mengenali variasi ukuran dan distribusi nodul atau lesi pada citra LSD. ResNet50 dipilih karena menggunakan *residual connection* yang mendukung pelatihan jaringan dalam secara lebih stabil. Sementara itu, Xception dipilih karena menggunakan *depthwise separable convolution* yang dapat mengekstraksi fitur secara lebih efisien. Dengan demikian, kelima model tersebut dipilih bukan hanya karena pernah digunakan pada penelitian terdahulu, tetapi karena masing-masing mewakili pendekatan ekstraksi fitur yang berbeda dalam klasifikasi citra.

Berdasarkan pengelompokan dan justifikasi tersebut, terdapat beberapa gap yang menjadi dasar penelitian ini. Pertama, hasil antarpelitian belum dapat dibandingkan secara langsung karena setiap penelitian menggunakan dataset, jumlah data, tahap *preprocessing*, strategi augmentasi, konfigurasi pelatihan, dan metrik evaluasi yang berbeda [10], [11], [17]. Kedua, beberapa penelitian masih berfokus pada performa klasifikasi, terutama *accuracy*, sehingga aspek efisiensi model seperti jumlah parameter, waktu pelatihan, dan *inference speed* belum banyak dianalisis secara menyeluruh [10], [11], [17]. Ketiga, interpretabilitas model melalui visualisasi seperti Grad-CAM belum banyak digunakan untuk menunjukkan area citra yang memengaruhi prediksi model [20], [21]. Keempat, belum banyak penelitian yang menempatkan CNN *baseline* dan beberapa model *transfer learning* dalam satu skenario optimasi dan evaluasi yang sama.

Oleh karena itu, penelitian ini dilakukan dengan menggunakan CNN sebagai *baseline* dan lima arsitektur *transfer learning*, yaitu DenseNet121, EfficientNetV2S, InceptionV3, ResNet50, dan Xception. Kelima model tersebut dioptimasi melalui *feature extraction*, *fine-tuning*, *class weight*, *early stopping*, dan

model checkpoint. Evaluasi dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, AUC, *confusion matrix*, jumlah parameter, waktu pelatihan, *inference speed*, *external inference*, dan Grad-CAM agar hasil evaluasi model dapat dianalisis secara lebih komprehensif.

2.2 Teori yang berkaitan

Bagian ini menjelaskan teori-teori utama yang menjadi dasar dalam penelitian deteksi *Lumpy Skin Disease* pada sapi menggunakan citra digital. Teori yang dibahas mencakup *computer vision*, *deep learning*, *transfer learning*, *data augmentation*, dan evaluasi model klasifikasi. Pemahaman terhadap teori-teori tersebut diperlukan agar proses pengolahan citra, pemilihan model, pelatihan, serta evaluasi performa dapat dijelaskan secara sistematis.

2.2.1 Teori Computer Vision

Computer vision merupakan bidang dalam kecerdasan buatan yang berfokus pada kemampuan komputer untuk memperoleh, memproses, menganalisis, dan memahami informasi visual dari citra atau video. Dalam konteks penelitian berbasis citra, *computer vision* digunakan untuk mengenali pola visual tertentu, seperti bentuk, tekstur, warna, dan struktur objek yang terdapat pada gambar. Pendekatan ini banyak digunakan dalam berbagai bidang karena mampu membantu proses pengamatan visual secara otomatis dan lebih konsisten dibandingkan pengamatan manual. Pada sektor peternakan, *computer vision* mulai banyak diterapkan untuk mendukung *precision livestock farming*, seperti pemantauan kondisi tubuh ternak, identifikasi individu, pengamatan perilaku, serta deteksi indikasi gangguan kesehatan berbasis citra [3], [22].

Dalam penerapannya pada kesehatan ternak, *computer vision* dapat digunakan sebagai metode pemantauan non-kontak karena data visual dapat diperoleh melalui kamera tanpa harus melakukan intervensi langsung terhadap hewan. Hal ini relevan dalam konteks peternakan modern karena pemeriksaan manual sering membutuhkan waktu, tenaga, dan keahlian khusus. Kajian terbaru menunjukkan bahwa pendekatan berbasis visi komputer dapat

membantu meningkatkan efisiensi manajemen peternakan melalui analisis citra yang lebih cepat dan terstruktur [3]. Selain itu, penerapan *computer vision* pada ternak juga dinilai potensial untuk mendukung pemantauan kesehatan dan kesejahteraan hewan karena mampu mengidentifikasi perubahan visual yang dapat berkaitan dengan kondisi fisik atau penyakit [22].

Pada penelitian ini, *computer vision* digunakan sebagai dasar pendekatan untuk menganalisis citra sapi sehat dan sapi yang terinfeksi *Lumpy Skin Disease*. Penyakit tersebut memiliki gejala visual berupa nodul atau lesi pada kulit, sehingga citra digital dapat digunakan sebagai sumber data untuk proses klasifikasi. Dengan memanfaatkan *computer vision*, sistem dapat diarahkan untuk mengenali perbedaan pola visual antara kulit sapi sehat dan kulit sapi yang terinfeksi. Oleh karena itu, *computer vision* menjadi landasan penting dalam penelitian ini karena memungkinkan proses deteksi penyakit dilakukan melalui analisis citra digital secara otomatis.

2.2.2 Teori Deep Learning

Deep learning ialah bentuk cabang dari *machine learning* yang menerapkan jaringan saraf tiruan yang berlapis untuk memahami representasi data secara otomatis. Berbeda dengan metode *machine learning* konvensional yang umumnya membutuhkan proses ekstraksi fitur dengan manual, *deep learning* dapat mempelajari fitur dari data mentah melalui beberapa lapisan pemrosesan. Setiap lapisan dalam jaringan mempelajari representasi yang berbeda, mulai dari fitur seperti tepi dan tekstur, serta fitur yang lebih kompleks seperti bentuk objek atau pola visual tertentu. Kemampuan tersebut menjadikan *deep learning* sangat relevan untuk tugas pengolahan citra, terutama klasifikasi citra berbasis objek atau kondisi tertentu [23], [24].

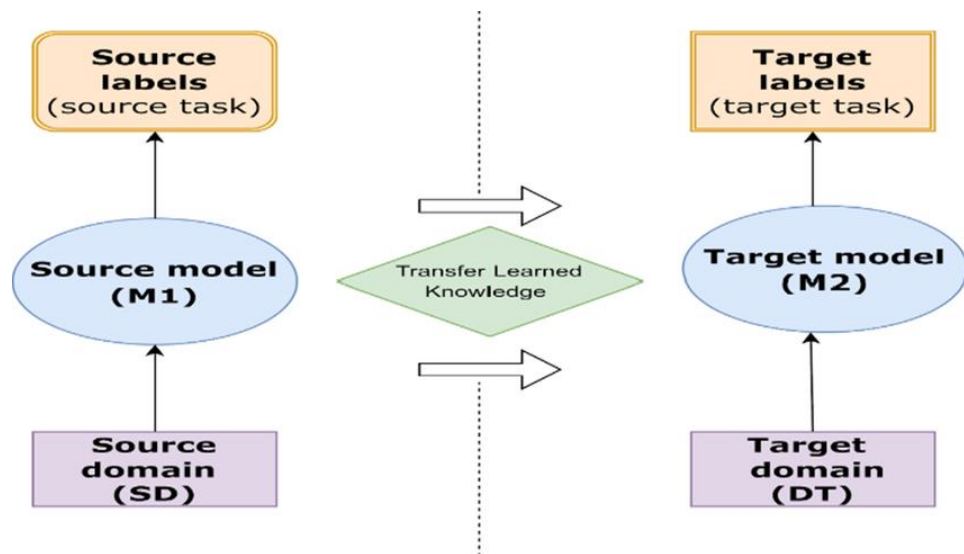
Salah satu bentuk arsitektur *deep learning* yang paling sering digunakan dalam mengolah citra adalah *Convolutional Neural Network* (CNN). CNN dirancang untuk memproses data berbentuk *grid* seperti citra digital yang memanfaatkan operasi konvolusi untuk mengekstraksi fitur spasial. Lapisan

konvolusi berfungsi mengenali pola lokal pada citra, sedangkan lapisan berikutnya menggabungkan pola tersebut menjadi representasi visual yang lebih kompleks. Tinjauan terbaru menunjukkan bahwa CNN banyak digunakan dalam berbagai tugas klasifikasi citra karena memiliki kemampuan yang kuat dalam mempelajari pola visual secara otomatis, termasuk pada bidang kesehatan, pertanian, dan peternakan [23], [25].

Dalam konteks deteksi penyakit berbasis citra, *deep learning* dapat membantu mengenali pola visual yang sulit didefinisikan secara eksplisit menggunakan aturan manual. Pada kasus *Lumpy Skin Disease*, fitur visual seperti nodul, lesi, perubahan tekstur kulit, dan pola distribusi kelainan dapat dipelajari oleh model CNN melalui proses pelatihan. Hal ini memungkinkan model membedakan citra sapi sehat dan citra sapi yang terinfeksi berdasarkan representasi fitur yang dipelajari dari data. Oleh karena itu, *deep learning* menjadi pendekatan utama dalam penelitian ini karena mampu mendukung proses klasifikasi citra penyakit ternak secara otomatis dan berbasis data.

2.2.3 Teori Transfer Learning

Transfer learning adalah sebuah pendekatan dalam *deep learning* yang menggunakan pengetahuan dari model yang telah dilatih sebelumnya pada dataset dalam skala besar untuk menyelesaikan tugas baru yang memiliki karakteristik serupa. Dalam konteks klasifikasi citra, model *pretrained* umumnya telah mempelajari fitur visual umum seperti tepi, tekstur, warna, bentuk, dan pola objek dari dataset besar, kemudian pengetahuan tersebut digunakan kembali pada dataset target yang lebih spesifik [22], [26]. Pendekatan ini berbeda dengan pelatihan model dari awal karena proses pembelajaran tidak dimulai dari bobot acak, melainkan dari bobot yang telah memiliki representasi fitur visual dasar. Oleh karena itu, *transfer learning* sering digunakan ketika jumlah dataset terbatas, waktu pelatihan ingin dipercepat, atau ketika model membutuhkan kemampuan generalisasi yang lebih baik pada data baru [27], [28].



Gambar 2. 1 Konsep *transfer learning* pada model pretrained untuk klasifikasi citra [29]

Berdasarkan Gambar 2.1 Konsep *transfer learning* pada model *pretrained* untuk klasifikasi citra, *transfer learning* memungkinkan model memanfaatkan pengetahuan yang telah dipelajari dari tugas sebelumnya untuk mendukung penyelesaian tugas baru. Dalam konteks penelitian ini, pengetahuan dari model *pretrained* digunakan sebagai dasar ekstraksi fitur visual, kemudian bagian klasifikasi akhir disesuaikan dengan kelas target, yaitu *Healthy Cows* dan *Lumpy Skin Disease*. Dengan demikian, model tidak perlu mempelajari seluruh representasi visual dari awal, tetapi dapat menggunakan fitur awal yang telah dipelajari sebelumnya untuk mendukung proses klasifikasi citra.

Dalam implementasinya, *transfer learning* dapat dilakukan melalui dua strategi utama, yaitu *feature extraction* dan *fine-tuning*. Pada strategi *feature extraction*, lapisan awal dari model *pretrained* digunakan sebagai ekstraktor fitur, sedangkan lapisan klasifikasi akhir diganti dan dilatih sesuai jumlah kelas pada dataset target. Sementara itu, pada strategi *fine-tuning*, sebagian lapisan dari model *pretrained* dibuka kembali untuk dilatih agar representasi fitur dapat menyesuaikan karakteristik dataset baru [28]. Pemilihan strategi tersebut bergantung pada ukuran dataset, kemiripan domain antara dataset sumber dan dataset target, serta kompleksitas tugas klasifikasi. Jika dataset target relatif kecil, penggunaan model *pretrained* sebagai *feature extractor* dapat membantu

mengurangi risiko *overfitting*. Namun, apabila dataset target memiliki jumlah data yang memadai, *fine-tuning* dapat digunakan untuk meningkatkan adaptasi model terhadap karakteristik citra yang lebih spesifik [26], [28].

Pada penelitian deteksi *Lumpy Skin Disease* berbasis citra, *transfer learning* menjadi pendekatan yang relevan karena dataset citra penyakit ternak umumnya tidak sebesar dataset citra umum. Gejala visual *Lumpy Skin Disease* berupa nodul, lesi, dan perubahan tekstur kulit memerlukan kemampuan ekstraksi fitur yang baik agar model dapat membedakan citra sapi sehat dan citra sapi terinfeksi secara akurat. Dengan menggunakan model *pretrained*, penelitian ini dapat memanfaatkan representasi visual yang telah dipelajari sebelumnya untuk mendukung klasifikasi citra penyakit. Selain itu, beberapa studi di bidang klasifikasi citra medis menunjukkan bahwa *transfer learning* dapat meningkatkan performa model pada kondisi dataset terbatas dan membantu mempercepat proses pelatihan [27], [28]. Oleh karena itu, pendekatan *transfer learning* digunakan dalam penelitian ini untuk membandingkan performa beberapa arsitektur seperti DenseNet121, EfficientNetV2S, InceptionV3, ResNet50, dan Xception dalam mendeteksi *Lumpy Skin Disease* pada sapi.

2.2.4 Teori Data Augmentation

Data augmentation merupakan teknik peningkatan variasi data dengan cara melakukan transformasi terhadap data pelatihan tanpa mengubah label kelas. Pada data citra, teknik ini dapat dilakukan melalui transformasi geometris seperti rotasi, translasi, *flipping*, *cropping*, dan *zooming*, maupun transformasi warna seperti perubahan kecerahan, kontras, saturasi, dan intensitas piksel [30]. Tujuan utama *data augmentation* adalah membuat model lebih robust terhadap variasi kondisi citra yang mungkin muncul pada data nyata. Dengan memperbanyak variasi citra selama proses pelatihan, model dapat belajar mengenali pola visual secara lebih fleksibel dan tidak terlalu bergantung pada karakteristik gambar tertentu dalam dataset pelatihan [30].

Dalam penelitian berbasis *deep learning*, *data augmentation* sering digunakan untuk mengurangi risiko *overfitting*, terutama ketika jumlah dataset terbatas. *Overfitting* terjadi ketika model memiliki performa tinggi pada data pelatihan, tetapi gagal melakukan generalisasi pada data validasi atau data uji. Kondisi ini sering ditemukan pada model CNN yang kompleks karena jumlah parameter yang besar membutuhkan data pelatihan yang memadai [31]. Dengan menerapkan *data augmentation*, variasi data pelatihan menjadi lebih beragam sehingga model memperoleh paparan terhadap variasi sudut pandang, skala, orientasi, dan pencahayaan. Hal ini dapat membantu meningkatkan kemampuan generalisasi model terhadap citra baru yang belum pernah dilihat sebelumnya [30].

Pada deteksi *Lumpy Skin Disease*, *data augmentation* menjadi penting karena citra sapi dapat memiliki variasi tinggi, seperti perbedaan sudut pengambilan gambar, jarak kamera, pencahayaan, latar belakang, ukuran lesi, dan distribusi nodul pada kulit. Variasi tersebut dapat memengaruhi kemampuan model dalam mengenali pola penyakit secara konsisten. Selain itu, keterbatasan dataset publik untuk klasifikasi *Lumpy Skin Disease* membuat *data augmentation* relevan untuk membantu memperkaya representasi data pelatihan. Oleh karena itu, teknik *data augmentation* dapat digunakan sebagai salah satu strategi untuk meningkatkan kemampuan generalisasi model *transfer learning* dalam membedakan citra sapi sehat dan citra sapi yang terinfeksi *Lumpy Skin Disease*.

2.2.5 Teori Evaluasi Model Klasifikasi

Evaluasi model klasifikasi merupakan tahap penting untuk mengukur kemampuan model dalam memprediksi kelas data secara benar. Pada penelitian berbasis klasifikasi citra, evaluasi tidak hanya dilakukan untuk mengetahui jumlah prediksi yang benar, tetapi juga untuk memahami jenis kesalahan prediksi yang dihasilkan oleh model. Secara umum, evaluasi model klasifikasi dapat dilakukan menggunakan *confusion matrix* dan beberapa metrik turunannya, seperti *accuracy*, *precision*, *recall*, dan *F1-score* [32]. Metrik-

metrik tersebut digunakan untuk memberikan gambaran performa model dari sudut pandang yang berbeda, sehingga evaluasi tidak hanya bergantung pada satu indikator saja.

Confusion matrix didefinisikan sebagai tabel evaluasi yang mengkomparasikan hasil prediksi model dengan label aktual pada dataset. Pada klasifikasi biner, *confusion matrix* terdiri atas empat komponen inti, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) [32], [33]. TP menampilkan jumlah data positif yang berhasil prediksi sebagai positif, sedangkan TN menampilkan jumlah data negatif yang berhasil diprediksi sebagai negatif. FP terjadi ketika data negatif salah diprediksi sebagai positif, sementara FN terjadi ketika data positif salah diprediksi sebagai negatif. Dalam konteks deteksi *Lumpy Skin Disease*, FN menjadi perhatian penting karena menunjukkan citra sapi yang sebenarnya terinfeksi tetapi diprediksi sebagai sapi sehat. Kesalahan tersebut berpotensi lebih berisiko dalam skenario deteksi penyakit karena dapat menyebabkan kasus infeksi tidak teridentifikasi secara dini.

Accuracy adalah sebuah metrik yang menyatakan proporsi prediksi benar terhadap keseluruhan data yang diuji. Metrik ini dihitung berdasarkan jumlah TP dan TN dibandingkan dengan seluruh hasil prediksi [32]. Secara matematis, *accuracy* dapat dinyatakan sebagai berikut:

(2. 1)

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

Meskipun *accuracy* lebih mudah dipahami dan lebih sering digunakan, metrik ini dapat menjadi kurang representatif apabila datasetnya memiliki komposisi distribusi kelas yang tidak atau kurang seimbang [34]. Pada kasus klasifikasi penyakit, model dapat terlihat memiliki *accuracy* tinggi apabila mampu memprediksi kelas mayoritas dengan baik, tetapi tetap gagal mendeteksi kelas positif yang lebih penting. Oleh karena itu, *accuracy* perlu didukung oleh metrik lain seperti *precision*, *recall*, dan *F1-score*.

Precision mengukur tingkat ketepatan model dalam memprediksi kelas positif. Metrik ini menunjukkan seberapa banyak data yang diprediksi positif benar-benar merupakan kelas positif [32], [33]. Secara matematis, *precision* dirumuskan sebagai berikut:

(2. 2)

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

Dalam penelitian deteksi *Lumpy Skin Disease*, *precision* dapat digunakan untuk mengetahui seberapa tepat model ketika menyatakan bahwa suatu citra termasuk kelas terinfeksi LSD. Nilai *precision* yang tinggi mengindikasikan bahwa model mempunyai tingkat kesalahan positif yang rendah, sehingga citra sapi sehat tidak banyak salah diklasifikasikan sebagai terinfeksi.

Recall umumnya digunakan untuk mengukur kemampuan model dalam menemukan seluruh data positif yang asli. Metrik ini menampilkan proporsi data positif aktual yang berhasil dikenali oleh model [32], [33]. Secara matematis, *recall* dapat dirumuskan sebagai berikut:

(2. 3)

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

Dalam konteks deteksi penyakit, *recall* memiliki peran krusial karena berkaitan dengan kemampuan model dalam mendeteksi kasus positif. Nilai *recall* yang rendah menunjukkan bahwa masih banyak citra sapi terinfeksi yang salah diklasifikasikan sebagai sapi sehat. Oleh karena itu, pada kasus deteksi *Lumpy Skin Disease*, *recall* perlu diperhatikan agar model tidak hanya akurat secara umum, tetapi juga mampu mengenali citra sapi yang benar-benar terinfeksi.

F1-score adalah metrik yang terdiri dari *precision* dan *recall* ke dalam satu nilai melalui rata-rata harmonik. Metrik ini dimanfaatkan untuk menilai keseimbangan antara ketepatan prediksi positif dan kemampuan model dalam menemukan seluruh data positif [32], [34]. Berikut adalah rumus dari *F1-score*:

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

F1-score menjadi relevan ketika dataset memiliki distribusi kelas yang tidak sepenuhnya seimbang atau ketika kesalahan FP dan FN sama-sama perlu diperhatikan. Nilai *F1-score* yang tinggi membuktikan bahwa model mempunyai keseimbangan yang baik antara *precision* dan *recall*. Dalam penelitian ini, *F1-score* digunakan untuk memudahkan dalam penentuan model yang tidak hanya memiliki *accuracy* tinggi, tetapi juga bisa menjaga keseimbangan performa dalam mendeteksi citra sapi sehat dan citra sapi yang terinfeksi *Lumpy Skin Disease*.

Berdasarkan penjelasan tersebut, penelitian ini menggunakan *accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix* sebagai metrik evaluasi model. Metrik yang digunakan tersebut bertujuan untuk memperoleh analisis performa yang lebih valid terhadap setiap arsitektur *transfer learning*. Dengan demikian, model terbaik yang sudah dibandingkan tidak hanya ditentukan berdasarkan nilai *accuracy*, tetapi juga berdasarkan kemampuan model dalam mengurangi kesalahan klasifikasi dan menjaga keseimbangan performa pada kedua kelas.

2.2.6 Teori Gradient-weighted Class Activation Mapping

Gradient-weighted Class Activation Mapping atau Grad-CAM merupakan salah satu metode *explainable artificial intelligence* yang digunakan untuk membantu menginterpretasikan keputusan model berbasis CNN. Pada model klasifikasi citra, Grad-CAM bekerja dengan memanfaatkan informasi gradien dari kelas target terhadap feature map pada layer konvolusi terakhir. Informasi tersebut kemudian digunakan untuk menghasilkan heatmap yang menunjukkan area citra yang memiliki kontribusi besar terhadap keputusan prediksi model. Area dengan intensitas aktivasi lebih tinggi menunjukkan bagian citra yang lebih diperhatikan oleh model dalam menentukan kelas prediksi. Dengan

demikian, Grad-CAM dapat membantu memberikan gambaran visual mengenai alasan model memberikan prediksi tertentu pada citra input [20], [21].

Dalam penelitian berbasis citra, Grad-CAM banyak digunakan untuk meningkatkan interpretabilitas model karena model deep learning sering dianggap sebagai *black box*. Interpretabilitas menjadi penting terutama pada konteks klasifikasi citra kesehatan, karena hasil prediksi model perlu dianalisis apakah didasarkan pada area visual yang relevan atau justru dipengaruhi oleh latar belakang dan objek lain yang tidak berkaitan dengan target klasifikasi. Beberapa studi menunjukkan bahwa visualisasi berbasis Grad-CAM dapat digunakan untuk membantu memahami area perhatian model, mengevaluasi kemungkinan bias pada citra, serta mendukung analisis hasil prediksi model CNN [20], [35].

Pada penelitian ini, Grad-CAM digunakan sebagai analisis interpretabilitas tambahan terhadap model yang telah dilatih. Visualisasi Grad-CAM diterapkan untuk melihat apakah model memberikan perhatian pada area tubuh sapi yang relevan dengan karakteristik visual *Lumpy Skin Disease*, seperti nodul, lesi, benjolan, dan perubahan tekstur kulit. Hasil Grad-CAM tidak digunakan sebagai dasar utama dalam menentukan model terbaik, tetapi sebagai pendukung interpretasi terhadap hasil prediksi model. Dengan demikian, penentuan model paling optimal tetap didasarkan pada hasil evaluasi kuantitatif, seperti *accuracy*, *precision*, *recall*, *F1-score*, *AUC*, *confusion matrix*, jumlah parameter, waktu pelatihan, dan *inference speed*.

2.3 Framework/Algoritma yang digunakan

Bagian ini menjelaskan *framework* penelitian dan algoritma yang digunakan dalam proses deteksi *Lumpy Skin Disease* pada sapi berbasis citra digital. *Framework* yang digunakan dalam penelitian ini adalah CRISP-DM karena penelitian berfokus pada proses analisis data, pengolahan dataset, pemodelan, dan evaluasi model, bukan pada pengembangan aplikasi perangkat lunak. Selain itu, penelitian ini menggunakan pendekatan *deep learning* berbasis *Convolutional Neural Network* (CNN) dan beberapa arsitektur *transfer learning*, yaitu

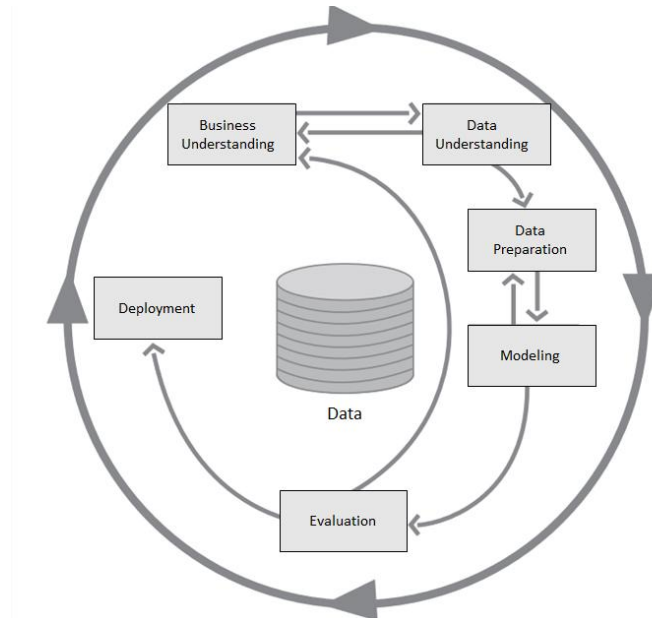
DenseNet121, EfficientNetV2S, InceptionV3, ResNet50, dan Xception. Pemilihan algoritma tersebut didasarkan pada perbedaan karakteristik arsitektur dalam mengekstraksi fitur citra, sehingga dapat digunakan untuk mengevaluasi model yang paling sesuai dalam klasifikasi citra sapi sehat dan sapi yang terinfeksi *Lumpy Skin Disease*.

2.3.1 Teori Metode CRISP-DM

Cross-Industry Standard Process for Data Mining (CRISP-DM) adalah salah satu metodologi yang sering digunakan dalam banyak penelitian yang berkaitan dengan *data mining*, *machine learning*, dan *data science* karena menyediakan alur kerja yang sistematis dari identifikasi masalah hingga evaluasi hasil model [36], [37]. Metodologi ini membantu peneliti dalam mengelola proses analisis data mulai dari pemahaman permasalahan, pemahaman karakteristik data, persiapan data, pembangunan model, hingga evaluasi hasil model. Dengan adanya tahapan yang jelas, CRISP-DM dapat digunakan untuk memastikan bahwa proses penelitian tidak hanya berfokus pada hasil akhir model, tetapi juga memperhatikan kualitas data, kesesuaian metode, serta validitas proses evaluasi.

CRISP-DM memiliki lima tahapan utama, yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, dan *evaluation* [36]. Kelima tahapan tersebut bersifat iteratif, dengan begitu peneliti dapat melakukan kembali tahap sebelumnya jika ditemukan permasalahan atau kebutuhan penyesuaian selama proses pengembangan model [37]. Dalam penelitian ini, CRISP-DM digunakan sebagai *framework* metodologis untuk mendukung proses deteksi Lumpy Skin Disease pada sapi berbasis citra digital. Penggunaan CRISP-DM membantu menyusun tahapan penelitian secara sistematis, mulai dari identifikasi kebutuhan deteksi awal penyakit, pemahaman dataset citra sapi, proses pembersihan dan persiapan data, pelatihan model *transfer learning*, hingga evaluasi performa model. Dengan demikian, CRISP-DM tidak hanya berfungsi sebagai alur kerja penelitian, tetapi juga sebagai dasar untuk memastikan bahwa proses pengolahan dataset, pembangunan model klasifikasi

citra, dan evaluasi performa dilakukan secara terarah dan dapat dipertanggungjawabkan.



Gambar 2. 2 Siklus *Cross-Industry Standard Process for Data Mining* (CRISP-DM) [36]

Berdasarkan Gambar 2.2 Siklus *Cross-Industry Standard Process for Data Mining* (CRISP-DM), setiap tahapan dalam CRISP-DM saling berhubungan dan membentuk suatu siklus pengembangan model yang berulang. Pada penelitian ini, setiap tahapan CRISP-DM diimplementasikan sesuai kebutuhan studi komparatif *transfer learning* untuk klasifikasi citra *Lumpy Skin Disease* pada sapi.

1. *Business Understanding*

Tahap pertama adalah *business understanding*. Tahap ini bertujuan untuk memahami permasalahan, kebutuhan, tujuan, serta konteks penggunaan dari proyek yang akan dilakukan. Pada tahap ini, permasalahan didefinisikan secara jelas agar proses analisis data memiliki arah yang sesuai dengan kebutuhan. Selain itu, tahap ini juga mencakup perumusan tujuan, identifikasi batasan, serta penentuan ukuran keberhasilan yang akan digunakan dalam proyek. Pemahaman

masalah yang baik pada tahap awal sangat penting karena akan memengaruhi keputusan pada tahap-tahap berikutnya [36].

2. *Data Understanding*

Tahap kedua adalah *data understanding*. Tahap ini berfokus pada proses pengumpulan dan pemahaman awal terhadap data yang tersedia. Aktivitas pada tahap ini dapat mencakup identifikasi sumber data, pemeriksaan struktur data, eksplorasi awal, analisis kualitas data, serta identifikasi potensi masalah seperti data hilang, data duplikat, data tidak konsisten, atau ketidakseimbangan distribusi data. Tahap ini penting karena kualitas dan karakteristik data akan sangat memengaruhi proses persiapan data serta performa model yang dibangun [36], [37].

3. *Data Preparation*

Tahap ketiga adalah *data preparation*. Tahap ini bertujuan untuk menyiapkan data supaya bisa digunakan dalam proses pemodelan. Aktivitas yang dilaksanakan pada tahap ini dapat mencakup pembersihan data, transformasi data, seleksi fitur, normalisasi, pengkodean label, penggabungan data, pemilahan dataset, serta proses lain yang diperlukan sesuai karakteristik data. Dalam proyek berbasis citra, tahap ini juga dapat mencakup proses pengubahan ukuran gambar, normalisasi piksel, dan peningkatan variasi data. Tahap *data preparation* umumnya menjadi salah satu tahap yang paling banyak memerlukan waktu karena data mentah sering kali belum berada dalam kondisi siap pakai [36].

4. *Modeling*

Tahap keempat adalah *modeling*. Pada tahap ini, metode atau algoritma yang sesuai dipilih dan diterapkan untuk membangun model. Proses *modeling* dapat mencakup pemilihan algoritma, penentuan parameter, pelatihan model, serta pengujian beberapa konfigurasi untuk

memperoleh performa yang optimal. Tahap ini sering dilakukan secara iteratif karena hasil awal model dapat menunjukkan perlunya penyesuaian pada parameter, teknik pemrosesan data, atau bahkan pemilihan algoritma lain. Oleh karena itu, tahap *modeling* memiliki hubungan yang erat dengan tahap *data preparation* dan *evaluation* [36], [37].

5. *Evaluation*

Tahap kelima adalah *evaluation*. Tahap ini bertujuan untuk menilai kualitas model atau hasil analisis yang telah diperoleh. Evaluasi dilakukan untuk memastikan bahwa model tidak hanya menghasilkan performa teknis yang baik, tetapi juga sesuai dengan tujuan yang telah ditetapkan pada tahap awal. Pada tahap ini, hasil model dapat dianalisis menggunakan metrik evaluasi yang sesuai dengan jenis permasalahan, misalnya klasifikasi, regresi, atau klusterisasi. Apabila hasil evaluasi belum memenuhi tujuan yang ditentukan, proses dapat kembali ke tahap sebelumnya untuk dilakukan perbaikan [36].

2.3.2 Teori Model Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu bentuk arsitektur utama dalam *deep learning* yang dibuat untuk mengolah data citra. Cara kerja CNN ialah dengan memanfaatkan lapisan konvolusi untuk mengekstraksi fitur visual dari citra, seperti tepi, tekstur, bentuk, dan pola objek. Dalam klasifikasi citra, CNN mampu mempelajari representasi fitur dengan otomatis tanpa diperlukannya ekstraksi fitur manual. Hal ini menjadikan CNN banyak digunakan dalam berbagai penelitian *computer vision*, termasuk deteksi penyakit berbasis citra pada bidang kesehatan, pertanian, dan peternakan [23].

Secara umum, CNN terdiri dari beberapa komponen utama, yaitu *convolutional layer*, *activation function*, *pooling layer*, *fully connected layer*, dan *output layer*. *Convolutional layer* berfungsi mengekstraksi fitur lokal dari

citra, sedangkan *pooling layer* digunakan untuk mereduksi dimensi fitur dan mempertahankan informasi penting. Pada tahap akhir, *fully connected layer* digunakan untuk melakukan klasifikasi berdasarkan fitur yang telah dipelajari. Dalam penelitian ini, CNN menjadi dasar konseptual bagi arsitektur *transfer learning* yang digunakan, karena model seperti DenseNet121, EfficientNetV2S, InceptionV3, ResNet50, dan Xception merupakan pengembangan dari prinsip dasar CNN.

2.3.3 Teori Model DenseNet121

DenseNet121 merupakan arsitektur CNN yang menggunakan konsep *dense connectivity*, yaitu setiap lapisan menerima masukan dari seluruh lapisan sebelumnya. Mekanisme ini memungkinkan informasi fitur mengalir lebih baik dari lapisan awal hingga lapisan akhir, sehingga dapat mengurangi risiko hilangnya informasi selama proses pembelajaran. Selain itu, *dense connectivity* dapat membantu meningkatkan efisiensi penggunaan fitur karena fitur yang telah dipelajari pada lapisan sebelumnya dapat digunakan kembali oleh lapisan berikutnya [11].

Dalam konteks klasifikasi citra *Lumpy Skin Disease*, DenseNet121 relevan karena citra penyakit memiliki detail visual berupa nodul, tekstur kulit, dan lesi yang perlu dikenali secara spesifik. Penelitian terdahulu juga telah menggunakan DenseNet dalam klasifikasi citra LSD, tetapi hasil performanya masih bervariasi antarpelelitian [23]. Oleh karena itu, DenseNet121 digunakan dalam penelitian ini untuk mengevaluasi apakah arsitektur berbasis *dense connectivity* dapat memberikan performa yang baik dalam membedakan citra sapi sehat dan sapi yang terinfeksi *Lumpy Skin Disease*.

2.3.4 Teori Model EfficientNetV2S

EfficientNetV2S merupakan salah satu varian dari keluarga EfficientNetV2 yang dikembangkan untuk meningkatkan efisiensi pelatihan dan performa klasifikasi citra. EfficientNetV2 dirancang dengan menggabungkan *training-*

aware neural architecture search dan strategi penskalaan model, sehingga mampu menghasilkan model yang lebih efisien dari sisi parameter dan kecepatan pelatihan [11], [13]. Dibandingkan keluarga EfficientNet sebelumnya, EfficientNetV2 memperkenalkan penggunaan operasi seperti *Fused-MBConv* untuk mempercepat proses pelatihan dan meningkatkan efisiensi komputasi [13].

Pemilihan EfficientNetV2S dalam penelitian ini didasarkan pada posisinya sebagai arsitektur CNN modern yang dirancang untuk menghasilkan keseimbangan antara performa dan efisiensi. Beberapa penelitian deteksi LSD sebelumnya telah menggunakan model *pretrained* seperti MobileNetV2, DenseNet, InceptionV3, Xception, dan EfficientNetV2S dalam skenario klasifikasi citra [10], [11]. Dengan menggunakan EfficientNetV2S, penelitian ini dapat mengevaluasi apakah arsitektur yang lebih baru mampu memberikan performa yang kompetitif dalam deteksi *Lumpy Skin Disease* pada sapi dibandingkan model *transfer learning* lainnya.

2.3.5 Teori Model InceptionV3

InceptionV3 merupakan arsitektur CNN yang dirancang untuk mengekstraksi fitur visual pada berbagai skala melalui struktur *inception module*. Arsitektur ini memungkinkan model memproses informasi dengan beberapa ukuran filter secara paralel, sehingga mampu menangkap pola visual kecil maupun besar dalam citra. Karakteristik tersebut relevan untuk klasifikasi citra penyakit kulit karena ukuran, bentuk, dan distribusi lesi dapat bervariasi antar citra [11].

Pada deteksi *Lumpy Skin Disease*, InceptionV3 dapat digunakan untuk mengenali variasi visual seperti nodul kecil, lesi menyebar, atau perubahan tekstur kulit pada area tertentu. Beberapa penelitian terdahulu juga memasukkan InceptionV3 sebagai salah satu model *pretrained* untuk klasifikasi citra LSD [10], [11]. Oleh karena itu, InceptionV3 digunakan dalam penelitian ini sebagai representasi arsitektur CNN berbasis ekstraksi fitur multi-skala.

2.3.6 Teori Model ResNet50

ResNet50 merupakan arsitektur CNN yang menggunakan mekanisme *residual connection* untuk mengatasi permasalahan degradasi performa pada jaringan yang semakin dalam. Dengan adanya *residual connection*, model dapat mempelajari selisih atau residual antara masukan dan keluaran lapisan, sehingga aliran gradien selama proses pelatihan menjadi lebih stabil. Mekanisme ini membuat ResNet50 banyak digunakan sebagai model dasar dalam berbagai tugas klasifikasi citra [11].

Dalam penelitian ini, ResNet50 digunakan karena memiliki kemampuan representasi fitur yang kuat dan stabil pada berbagai permasalahan *computer vision*. Pada kasus deteksi *Lumpy Skin Disease*, ResNet50 berpotensi mengenali pola visual lesi dan tekstur kulit melalui jaringan konvolusional yang dalam. Selain itu, ResNet juga sering digunakan sebagai model pembanding dalam penelitian klasifikasi citra penyakit, sehingga relevan untuk dimasukkan dalam studi komparatif bersama DenseNet121, EfficientNetV2S, InceptionV3, dan Xception [10], [11].

2.3.7 Teori Model Xception

Xception merupakan arsitektur CNN yang mengembangkan konsep *depthwise separable convolution*. Berbeda dengan konvolusi standar yang memproses informasi spasial dan antar-kanal secara bersamaan, *depthwise separable convolution* memisahkan proses ekstraksi fitur spasial dan kombinasi fitur antar-kanal. Pendekatan ini membuat proses ekstraksi fitur menjadi lebih efisien dan dapat mengurangi kompleksitas komputasi [11].

Dalam konteks klasifikasi citra *Lumpy Skin Disease*, Xception relevan karena citra penyakit memiliki pola visual yang kompleks, seperti variasi tekstur kulit, bentuk nodul, dan distribusi lesi. Dengan mekanisme *depthwise separable convolution*, Xception berpotensi mempelajari pola visual tersebut secara efisien. Beberapa penelitian terdahulu juga menggunakan Xception sebagai salah satu arsitektur *pretrained* dalam deteksi LSD berbasis citra [10],

[11]. Oleh karena itu, Xception digunakan dalam penelitian ini sebagai model pembandingan yang merepresentasikan pendekatan ekstraksi fitur efisien berbasis *depthwise separable convolution*.

2.4 Tools/software yang digunakan

Bagian ini menjelaskan perangkat lunak yang digunakan dalam proses pengolahan data, pelatihan model, dan evaluasi hasil penelitian. Pemilihan *tools/software* disesuaikan dengan kebutuhan penelitian berbasis *computer vision* dan *deep learning*, terutama untuk proses *preprocessing* citra, implementasi model *transfer learning*, visualisasi hasil, serta evaluasi performa model.

2.4.1 Python

Python merupakan bahasa pemrograman yang banyak digunakan dalam penelitian *machine learning* dan *deep learning* karena memiliki sintaks yang relatif sederhana, fleksibel, serta didukung oleh ekosistem pustaka ilmiah yang luas. Dalam penelitian berbasis data, Python sering digunakan untuk proses pengolahan data, analisis numerik, visualisasi, serta implementasi model pembelajaran mesin. Dukungan berbagai pustaka seperti NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, dan Keras menjadikan Python sesuai untuk penelitian yang melibatkan eksperimen model secara berulang [38].

Pada penelitian ini, Python digunakan sebagai bahasa pemrograman utama untuk menjalankan seluruh tahapan eksperimen. Tahapan tersebut mencakup pembacaan dataset citra, *preprocessing*, pembagian data, penerapan *data augmentation*, pelatihan model *transfer learning*, evaluasi performa model, serta pembuatan visualisasi seperti grafik *training*, grafik *validation*, dan *confusion matrix*. Penggunaan *Python* juga mendukung reproduktibilitas eksperimen karena setiap tahapan pemrosesan dan pelatihan model dapat terdokumentasi dalam bentuk kode program yang dapat dijalankan kembali [39].

2.4.2 Jupyter Notebook

Jupyter Notebook merupakan lingkungan kerja interaktif yang memungkinkan pengguna menjalankan kode, menampilkan hasil, membuat visualisasi, dan menulis dokumentasi dalam satu dokumen yang terintegrasi. Lingkungan ini banyak digunakan dalam penelitian *data science* dan *machine learning* karena mendukung proses eksplorasi data, eksperimen model, serta dokumentasi alur kerja secara bertahap [40]. Selain itu, *Jupyter Notebook* mendukung prinsip reproduktibilitas karena proses analisis dapat disusun dalam bentuk sel kode dan teks yang menjelaskan tahapan eksperimen [41].

Dalam penelitian ini, *Jupyter Notebook* digunakan untuk menjalankan proses eksperimen secara bertahap, mulai dari eksplorasi dataset, *preprocessing* citra, pelatihan model, hingga evaluasi hasil. Penggunaan *Jupyter Notebook* memudahkan proses pemantauan hasil pelatihan karena grafik *accuracy*, *loss*, dan *confusion matrix* dapat ditampilkan langsung setelah kode dijalankan. Selain itu, format interaktif pada *Jupyter Notebook* membantu proses pengujian beberapa arsitektur *transfer learning* secara lebih sistematis [40].

Eksperimen dalam *Jupyter Notebook* didukung oleh beberapa pustaka utama. *TensorFlow* dan *Keras* digunakan untuk membangun, melatih, dan mengevaluasi model *deep learning*, termasuk arsitektur DenseNet121, EfficientNetV2S, InceptionV3, ResNet50, dan Xception [11]. NumPy digunakan untuk komputasi numerik dan manipulasi array, sedangkan Pandas digunakan untuk pengolahan data tabular dan penyusunan hasil evaluasi. Matplotlib digunakan untuk membuat visualisasi grafik pelatihan, grafik validasi, dan *confusion matrix*. Scikit-learn digunakan untuk menghitung metrik evaluasi seperti *accuracy*, *precision*, *recall*, *F1-score*, serta membuat *classification report* [42].