BAB II
TINJAUAN PUSTAKA

2.1. Studi Literatur

Penelitian sebelumnya yang dilakukan oleh Ali *et al.*[4] berfokus pada penerapan *deep learning* untuk klasifikasi penyakit *mpox* dengan membandingkan performa beberapa *model pre-trained* yang telah banyak digunakan dalam tugas klasifikasi citra medis. Dalam penelitian ini, empat arsitektur *deep learning* digunakan, yaitu *VGG16*, *ResNet50*, *InceptionV3*, dan *Ensemble Learning*, yang diterapkan untuk membedakan antara citra lesi kulit *mpox* dan jenis lesi kulit lainnya. Hasil eksperimen menunjukkan bahwa masing-masing *model* memiliki tingkat akurasi yang bervariasi, dengan *VGG16* memperoleh akurasi sebesar 81,48%, *ResNet50* sebesar 82,96%, *InceptionV3* sebesar 74,07%, dan metode *Ensemble* mencapai akurasi 79,26% pada citra yang belum pernah dikenali sebelumnya. *Dataset* yang digunakan dalam penelitian ini adalah *Monkeypox Skin Lesion Dataset (MSLD)*, yang merupakan *Dataset* citra lesi kulit *mpox* yang banyak digunakan dalam berbagai penelitian terkait.

Selain itu, penelitian yang dilakukan oleh Haque *et al.*[5] mengeksplorasi pendekatan *deep transfer learning* dengan menambahkan *Convolutional Block Attention Module (CBAM)* sebagai mekanisme perhatian untuk meningkatkan fokus *model* terhadap area penting dalam citra lesi kulit *mpox*. Lima *model deep learning* yang diterapkan dalam penelitian ini meliputi *VGG19*, *Xception*, *DenseNet121*, *EfficientNetB3*, dan *MobileNetV2*. Dari hasil pengujian, *model Xception* dengan implementasi *CBAM* menunjukkan performa tertinggi dengan akurasi mencapai 83,89%, mengungguli *model* lainnya. *Dataset* yang digunakan dalam penelitian ini sama dengan yang digunakan dalam penelitian Ali *et al.*, yaitu *Monkeypox Skin Lesion Dataset (MSLD)*.

Lalu, penelitian yang dilakukan oleh Sitaula *et al.*[6] mengkaji performa tiga belas *model deep learning* yang berbeda dengan menerapkan teknik *fine-tuning* pada setiap *model* untuk meningkatkan akurasinya dalam mendeteksi lesi kulit *mpox*. Setiap *model* dalam penelitian ini disesuaikan dengan *custom layers* untuk menyesuaikan arsitektur dengan karakteristik data yang digunakan. Hasil eksperimen menunjukkan bahwa rata-rata akurasi yang diperoleh dari ketiga belas *model* ini mencapai 85,44%, yang menunjukkan peningkatan performa dibandingkan penelitian sebelumnya. *Dataset* yang digunakan tetap *MSLD*, yang telah menjadi *dataset* acuan dalam berbagai studi terkait klasifikasi citra lesi kulit *mpox*.

Terakhir, penelitian yang dilakukan oleh Irmak *et al.*[7] juga menggunakan pendekatan *pre-trained CNN models* untuk melakukan klasifikasi citra lesi kulit *mpox*. *Model* yang diimplementasikan dalam penelitian ini adalah *MobileNetV2*, *VGG16*, dan *VGG19*, yang diuji pada *dataset Monkeypox Skin Lesion Dataset (MSLD)*. Hasil penelitian menunjukkan bahwa *model MobileNetV2* memiliki performa tertinggi dengan akurasi sebesar 91,38%, presisi sebesar 90,5%, *recall* sebesar 86,75%, dan *f1-score* sebesar 88,25%. Sementara itu, *model VGG16* memperoleh akurasi 83,62%, dan *model VGG19* mencapai akurasi 78,45%. Ringkasan dari penelitian-penelitian ini dapat dilihat pada Tabel 1 berikut.

Tabel 1. *Dataset* proses *train*, *validation,* dan *test*

No	Peneliti	Implementasi	Hasil Akurasi
1	Ali, *et al.*	Memanfaatkan 4 *pre-trained model* yaitu *VGG16*, *ResNet50*, *Inception V3*, dan *Ensemble* dengan melatih *top-layer* nya untuk klasifikasi lesi kulit *mpox*.	81.48% (*VGG16*), 82.96% (*ResNet50*), 74.07% (*Inception V3*), dan 79.26% (*Ensemble*)
2	Haque, *et al.*	Menambahkan *Convolutional Block Attention Module (CBAM)* untuk meningkatkan fokus *model* terhadap area penting dalam citra lesi kulit	*Model* terbaik *Xception* mencapai 83.89%

No	Peneliti	Implementasi	Hasil Akurasi
		mpox. Lima *model deep learning* yang diterapkan dalam penelitian ini meliputi *VGG19*, *Xception*, *DenseNet121*, *EfficientNetB3*, dan *MobileNetV2*. Penelitian juga melakukan *freeze* pada *bottom layer* dari setiap *model*.	
3	Sitaula, *et al.*	Melakukan *fine-tuning* pada *top-layer* pada 13 *pre-trained model* untuk meningkatnya performa akurasinya.	Dari ke tiga belas *model* menghasilkan rata-rata di 85.44%
4	Irmak *et al.*	Mengimplementasikan *fine-tuning* pada *top-layer* dari *pre-trained model* *MobileNetV2*, *VGG16*, dan *VGG19*.	91.38% (*MobileNetV2*), 83.62% (*VGG16*), dan 78,45% (*VGG19*)

Berdasarkan ringkasan dari Tabel 1, terlihat bahwa penggunaan *model pre-trained* dapat memberikan performa yang cukup baik dalam klasifikasi citra lesi kulit *mpox*. Namun, penelitian masih mengandalkan *transfer learning* dengan bobot yang diperoleh dari *dataset ImageNet*. *Dataset ImageNet* memiliki cakupan domain yang lebih luas dan tidak secara khusus dilatih untuk mendeteksi lesi kulit *mpox*. Oleh karena itu, penelitian ini mengusulkan pendekatan yang berbeda dengan merancang arsitektur *CNN* khusus yang dilatih sepenuhnya dari awal menggunakan *dataset Monkeypox Skin Lesion Dataset*, sehingga diharapkan dapat lebih optimal dalam mengenali karakteristik unik dari lesi kulit *mpox*.

2.2. Virus *Mpox*

Virus *mpox* merupakan virus DNA beruntai ganda linier dengan ukuran sekitar 200–250 nm. Dibandingkan dengan virus RNA, virus ini memiliki

kestabilan yang lebih tinggi serta lebih efektif dalam mendeteksi dan menangani lesi yang berpotensi menyebabkan mutasi. Penyakit *mpox* disebabkan oleh infeksi virus *monkeypox*, yang dapat menular melalui kontak dengan hewan, lingkungan yang terkontaminasi, maupun antar manusia. Virus ini memiliki dua varian utama, yaitu varian Afrika Tengah (*Congo Basin*) dan varian Afrika Barat. Dari kedua varian tersebut, varian Afrika Tengah lebih umum ditemukan serta memiliki tingkat penularan antarmanusia yang lebih tinggi. Selain itu, varian ini juga dikaitkan dengan gejala yang lebih parah, tingkat kesakitan dan kematian yang lebih tinggi, serta *viremia* yang lebih berat[10].

2.3. Lesi Kulit *Mpox*

Lesi Kulit *Mpox* (*Monkeypox*) termasuk dalam kategori penyakit *zoonosis* yang sedang muncul (*emerging zoonosis*) dan disebabkan oleh virus *Monkeypox*. Penyakit ini umumnya bersifat ringan dengan gejala yang berlangsung sekitar 2–4 minggu, tetapi dalam beberapa kasus dapat berkembang menjadi infeksi yang lebih parah hingga menyebabkan kematian, dengan *Case Fatality Rate (CFR)* berkisar antara 3–6%.

Penularan virus ini terjadi melalui kontak langsung dengan individu atau hewan yang terinfeksi, serta paparan terhadap benda atau permukaan yang telah terkontaminasi virus. *Mpox* memiliki beragam gejala klinis, yang meliputi demam, sakit kepala hebat, pembengkakan kelenjar getah bening, sakit tenggorokan, hidung tersumbat, dan batuk. Gejala yang paling khas dari penyakit ini adalah munculnya ruam atau lesi pada kulit, yang dapat menyebar ke berbagai bagian tubuh dan berkembang dalam beberapa tahap sebelum akhirnya mengering dan mengelupas[11].

2.4. *Deep Learning*

Deep learning merupakan salah satu cabang dari *machine learning* yang memanfaatkan jaringan saraf tiruan (*neural networks*) untuk menyelesaikan permasalahan kompleks. *Model deep learning* terinspirasi dari struktur dan fungsi

otak manusia, memungkinkan sistem untuk belajar dari data yang tidak terstruktur dan tidak berlabel.

Salah satu keunggulan utama *deep learning* adalah kemampuannya untuk mempelajari pola dari data secara otomatis, tanpa memerlukan pemrograman eksplisit. Hal ini menjadikannya sangat efektif dalam berbagai aplikasi yang melibatkan jumlah data besar, seperti pengenalan citra dan suara, pemrosesan bahasa alami (*Natural Language Processing*), serta sistem rekomendasi[12].

2.5. *Convolutional Neural Networks (CNN)*

Convolutional Neural Networks (CNN) merupakan salah satu jenis utama jaringan saraf tiruan yang digunakan untuk pengenalan dan klasifikasi citra. *CNN* memiliki berbagai aplikasi, termasuk pengenalan objek, pemrosesan citra, visi komputer (*computer vision*), dan pengenalan wajah.

Pada *CNN*, citra digunakan sebagai input, dan jaringan ini mampu mempelajari hierarki fitur secara otomatis, tanpa perlu rekayasa fitur secara manual. Untuk mencapai hal ini, CNN membangun peta fitur (*feature maps*) secara hierarkis dengan cara melakukan konvolusi pada citra masukan menggunakan filter yang telah dipelajari. Proses ini memungkinkan *CNN* untuk mengekstrak karakteristik penting dari citra, yang kemudian digunakan dalam proses klasifikasi[13].

2.6. *Transfer Learning*

Transfer Learning bertujuan untuk meningkatkan pemahaman terhadap suatu tugas dengan menghubungkannya dengan tugas lain yang telah dilakukan sebelumnya dalam domain sumber yang terkait. Pendekatan ini memungkinkan pembelajaran yang lebih logis, cepat, dan efisien dengan memanfaatkan pengalaman dari tugas sebelumnya untuk menyelesaikan tugas yang sedang dihadapi.

Transfer Learning sangat berguna dalam situasi di mana jumlah data pelatihan untuk tugas target terbatas, karena *model* dapat memperoleh pengetahuan dari domain sumber yang memiliki lebih banyak data. Namun, hubungan antara

tugas sumber dan tugas target tidak selalu kompatibel. Jika data uji dan pelatihan tidak sesuai dalam proses transfer, maka dapat menyebabkan penurunan performa *model* pada tugas target, yang dikenal sebagai *negative transfer*. Sebaliknya, jika transfer dilakukan dengan tepat, performa *model* akan meningkat secara signifikan[14].

2.7. *Pre-Trained Model*

Pre-trained Model adalah jaringan saraf yang telah dilatih pada *dataset* berskala besar dan dapat digunakan kembali untuk tugas lain yang lebih spesifik. Konsep *pre-training* berkaitan erat dengan *transfer learning*, di mana pengetahuan yang diperoleh dari satu atau lebih tugas sebelumnya diterapkan pada tugas baru[15].

Salah satu *dataset* yang paling umum digunakan dalam *pre-training* adalah *ImageNet*, yang digunakan dalam berbagai *model deep learning* untuk tugas klasifikasi gambar. Beberapa *model* populer yang telah dilatih menggunakan *dataset* ini antara lain *ResNet50*, *DenseNet*, *InceptionV3*, dan *VGG16*[16].

2.8. *VGG16* dan *VGG19 Model*

Arsitektur *VGG16* pertama kali diperkenalkan oleh K. Simonyan dan A. Zisserman dari *University of Oxford* melalui *paper* yang berjudul *"Very Deep Convolutional Networks for Large-Scale Image Recognition"*. *Model* ini mampu mencapai akurasi sebesar 92,77% pada pengujian data menggunakan *dataset ImageNet[17]*.

VGG16 dirancang untuk menerima *input* gambar berukuran 224×224 piksel dengan 3 *channel* warna (*RGB*). Arsitekturnya terdiri dengan dua *layer convolutional* pertama yang dilanjutkan dengan satu *layer maxpooling*. Setelah itu, terdapat dua *layer convolutional* dan satu *layer maxpooling* lagi. Lalu, diikuti oleh tiga *layer convolutional*, satu *layer maxpooling*, tiga *layer convolutional*, satu *layer maxpooling*, serta tiga *layer convolutional* terakhir dan satu *maxpooling*. Pada bagian akhir arsitektur, terdiri dari tiga *fully connected layer* yang dihubungkan dengan fungsi aktivasi *ReLU[17]*.

Sementara itu, arsitektur *VGG19* memiliki struktur yang lebih dalam, yaitu terdiri dari dua *layer convolutional*, satu *layer maxpooling*, dua *layer convolutional*, satu *layer maxpooling*, empat *layer convolutional*, satu *layer maxpooling*, empat *layer convolutional*, satu *layer maxpooling*, serta empat *layer convolutional* dan satu *layer maxpooling* terakhir. Arsitektur ini ditutup dengan tiga *fully connected layer* dengan fungsi aktivasi *ReLU* juga[17]. Ringkasan dari arsitektur *VGG16* dan *VGG19* ini dapat dilihat pada Tabel 2 dibawah ini.

Tabel 2. Arsitektur *VGG16* dan *VGG19*

No.	***VGG16***	***VGG19***
1	*Image (Input Layer)*	*Image (Input Layer)*
2	*2 * Convolutional Layer*	*2 * Convolutional Layer*
3	*Maxpooling Layer*	*Maxpooling Layer*
4	*2 * Convolutional Layer*	*2 * Convolutional Layer*
5	*Maxpooling Layer*	*Maxpooling Layer*
6	*3 * Convolutional Layer*	*4 * Convolutional Layer*
7	*Maxpooling Layer*	*Maxpooling Layer*
8	*3 * Convolutional Layer*	*4 * Convolutional Layer*
9	*Maxpooling Layer*	*Maxpooling Layer*
10	*3 * Convolutional Layer*	*4 * Convolutional Layer*
11	*Maxpooling Layer*	*Maxpooling Layer*
12	*3 * Fully Connected Layer*	*3 * Fully Connected Layer*
13	*Output Layer*	*Output Layer*

2.9. *ResNet50 Model*

ResNet50 merupakan *model* berbasis *Convolutional Neural Network* (*CNN*) yang terdiri dari 48 *layer convolutional*, serta dilengkapi dengan 1 *layer maxpooling* dan 1 *layer average pooling*. Arsitektur *ResNet50* dirancang untuk memiliki *CNN* yang jauh lebih dalam dibandingkan arsitektur konvensional sebelumnya[17].

Pada umumnya, semakin banyak jumlah *layer* yang ditambahkan pada jaringan *neural* cenderung menyebabkan peningkatan *error* saat *training*. Namun, *ResNet* berhasil mengatasi permasalahan ini melalui konsep *residual learning*. Dalam arsitektur ini, ditambahkan *shortcut connection* antar *layer* yang memungkinkan informasi dapat melewati beberapa *layer* sekaligus tanpa melalui proses transformasi yang kompleks. Penggunaan *residual block* ini membantu jaringan dalam mempertahankan akurasi meskipun terdiri dari banyak *layer*[17]. Ringkasan dari Arsitektur ResNet50 ini dapat dilihat pada Tabel 3 berikut.

Tabel 3. Arsitektur *ResNet50*

No.	**Arsitektur *ResNet50***
1	*Image (Input Layer)*
2	*7x7 Convolutional Layer, 64*
3	*Maxpooling Layer*
4	*6 * (3x3 Convolutional Layer, 64)*
5	*6 * (3x3 Convolutional Layer, 128)*
6	*6 * (3x3 Convolutional Layer, 256)*
7	*6 * (3x3 Convolutional Layer, 512)*
8	*6 * (3x3 Convolutional Layer, 1024)*
9	*6 * (3x3 Convolutional Layer, 2048)*
10	*6 * (3x3 Convolutional Layer, 4096)*
11	*6 * (3x3 Convolutional Layer, 8192)*
12	*Averagepooling Layer*
13	*Fully Connected Layer*
14	*Output Layer*

2.10. *InceptionV3 Model*

InceptionV3 merupakan *model CNN* yang cukup umum digunakan dan telah terbukti mampu mencapai tingkat akurasi lebih dari 78,1% pada *dataset ImageNet*. Secara arsitektur, *InceptionV3* terdiri dari *deep neural network* yang memiliki

kedalaman hingga 42 layer. *Model* ini dibangun menggunakan kombinasi blok-blok simetris dan asimetris yang mencakup *convolutional layer*, *maxpooling layer*, *averagepooling layer*, *dropout*, serta *fully connected layer*[18]. Ringkasan dari Arsitektur *InceptionV3* ini dapat dilihat pada Tabel 4 dibawah ini.

Tabel 4. Arsitektur *InceptionV3*

No.	Arsitektur ***InceptionV3***
1	*Image (Input Layer)*
2	*3 * (3x3 Convolutional Layer)*
3	*Maxpooling Layer*
4	*1x1 Convolutional Layer*
5	*3x3 Convolutional Layer*
6	*Maxpooling Layer*
7	*Inception Module A*
8	*Reduction A*
9	*Inception Module B*
10	*Reduction B*
11	*Inception Module C*
12	*Averagepooling Layer*
13	*Dropout*
14	*Fully Connected Layer*
15	*Output Layer*

2.11.*Xception Model*

Xception (*Extreme Inception*) merupakan arsitektur *CNN* yang terdiri dari 71 layer. Keunggulan utama dari arsitektur ini terletak pada penggunaan *deeply separable convolution*, yaitu teknik konvolusi yang memisahkan *Depthwise Convolution* dan *Pointwise Convolution*, sehingga lebih efisien dibandingkan metode Convolution biasa. Pendekatan ini menjadi lompatan besar dibandingkan arsitektur pendahulunya yaitu *InceptionV3*[19].

Inovasi utama *deeply separable convolution* ini tidak hanya mengurangi kompleksitas komputasi, tetapi juga tetap menjaga kemampuan representasi *model* secara optimal. Selain itu, efisiensi *model Xception* juga terlihat dari jumlah parameter yang sebanding dengan *InceptionV3*, namun performa yang dihasilkan jauh lebih unggul[19]. Ringkasan dari Arsitektur *Xception* ini dapat dilihat pada Tabel 5 dibawah ini.

Tabel 5. Arsitektur *Xception*

No.	**Arsitektur *Xception***
1	*Image (Input Layer)*
2	*2 * (3x3 Convolutional Layer)*
3	*Maxpooling Layer*
4	*Residual Blocks 1*
5	*Residual Blocks 2*
6	*Residual Blocks 3*
7	*8 * (Residual Blocks 4)*
8	*Residual Blocks 5*
9	*Residual Blocks 6*
10	*3x3 Seperable Convolutional Layer*
11	*3x3 Seperable Convolutional Layer*
12	*GlobalAveragepooling Layer*
13	*Dropout*
14	*Fully Connected Layer*
15	*Output Layer*

2.12. *MobileNetV2 Model*

MobileNetV2, dikenal sebagai *feature extractor* yang sangat efisien untuk tugas deteksi dan segmentasi objek. Salah satu bagian penting dari arsitektur ini adalah *bottleneck*, yang menjadi titik pemrosesan antara *input* dan *output model*.

Sementara itu, lapisan-lapisan dalam (*inner layers*) bertanggung jawab untuk mentransformasikan *lower-level concept input* contohnya piksel, menjadi *higher-level descriptor* contohnya kategori gambar. Seperti halnya pada *CNN* konvensional yang menggunakan *residual connection*, *MobileNetV2* juga menerapkan *shortcut* antar *bottleneck* untuk mempercepat proses pelatihan sekaligus meningkatkan akurasi *model*[20]. Ringkasan dari Arsitektur *MobileNetV2* ini dapat dilihat pada Tabel 6 dibawah ini.

Tabel 6. Arsitektur *MobileNetV2*

No.	**Arsitektur *MobileNetV2***	
1	*Image (Input Layer)*	
2	*3x3 Convolutional Layer*	
3	*7 * (Bottleneck Blocks)*	*Convolutional Layer*
4		*Batch Normalization*
		ReLU6
		Depthwise Convolution
		Batch Normalization
		ReLU6
		Linear Convolutional Layer
		Batch Normalization
5	*1x1 Convolutional Layer*	
6	*GlobalAveragepooling Layer*	
7	*Fully Connected Layer*	
8	*Output Layer*	

2.13. *Fine-Tuning*

Pelatihan ulang *pre-trained model* memerlukan *resource* dan waktu yang sangat besar. Meskipun mampu mencapai performa yang sangat baik pada *ImageNet*, melatihnya kembali dari awal cukup memberatkan karena keterbatasan komputasi. *Model CNN* umumnya dilatih selama beberapa minggu atau lebih

dengan menggunakan sejumlah *GPU* untuk mendapatkan hasil optimal pada *dataset* yang kompleks dan beragam.

Dalam *fine-tuning*, tujuan utamanya adalah mempertahankan atau membekukan fitur tingkat rendah (*bottom layer*) yang telah dipelajari sebelumnya, sementara hanya fitur tingkat tinggi (*top layer*) yang dilatih ulang agar sesuai dengan permasalahan klasifikasi gambar yang baru. Pendekatan ini memungkinkan transfer pengetahuan yang lebih efisien tanpa harus melatih seluruh *model* dari awal[21].

2.14. Evaluasi *Model* (*Accuracy, Precision, Recall, F1-score*)

Metrik evaluasi sangat penting dalam menilai kinerja *model* klasifikasi. Akurasi mengukur proporsi prediksi yang benar (baik *true positive* maupun *true negative*) terhadap seluruh prediksi, sehingga memberikan gambaran umum tentang performa *model*. Namun, pada *dataset* yang tidak seimbang, metrik ini dapat memberikan gambaran yang menyesatkan[22].

Presisi didefinisikan sebagai rasio antara *true positive* dengan total prediksi positif, yang menunjukkan seberapa banyak prediksi positif yang benar. *Recall* atau sensitivitas, mengukur rasio *true positive* terhadap seluruh kasus positif sebenarnya, mencerminkan kemampuan *model* dalam mengidentifikasi seluruh *instance positif*[22].

F1-score, yang merupakan rata-rata harmonik dari *precision* dan *recall*, digunakan untuk menyeimbangkan kedua metrik tersebut, terutama dalam menangani *dataset* yang tidak seimbang. Metrik ini memberikan gambaran yang lebih adil mengenai kinerja *model* dalam mendeteksi kelas minoritas maupun mayoritas[22].

2.15. *Monkeypox Skin Lesion Dataset*

Pembuatan "*Monkeypox Image Lesion Dataset*" bertujuan utama untuk membedakan kasus *monkeypox* dari kasus *non-monkeypox* yang memiliki kemiripan. Oleh karena itu, selain kelas "*Monkeypox*", *dataset* ini juga mencakup

gambar lesi kulit dari penyakit "*Chickenpox*" dan "*Measles*", yang memiliki kesamaan dengan ruam serta *pustula* pada tahap awal infeksi *monkeypox*. Keduanya dimasukkan ke dalam kelas "*Others*" untuk memungkinkan klasifikasi biner[4]. *Dataset* ini terdiri dari tiga folder utama :

1. *Original Images*

   Folder ini berisi 228 gambar, dengan 102 gambar termasuk dalam kelas "*Monkeypox*", sementara 126 gambar lainnya mewakili kelas "*Others*" (kasus *non-monkeypox* seperti *chickenpox* dan *measles*).

2. *Augmented Images*

   Untuk meningkatkan performa klasifikasi, metode augmentasi data seperti rotasi, translasi, refleksi, *shear*, perubahan rona (*hue*), saturasi, kontras, kecerahan, penambahan *noise*, dan skala (*scaling*) diterapkan. Augmentasi dilakukan menggunakan *MATLAB R2020a*, meskipun metode ini juga dapat diterapkan melalui *ImageGenerator* atau *augmentor* gambar lainnya. Untuk memastikan *reproducibility*, hasil augmentasi disertakan dalam *folder* ini. Setelah augmentasi, jumlah gambar meningkat sekitar 14 kali lipat, dengan 1428 gambar pada kelas "*Monkeypox*" dan 1764 gambar pada kelas "*Others*".

3. *Fold1*

   Folder ini merupakan salah satu dari tiga set *cross-validation* dengan metode *three-fold cross-validation* untuk menghindari bias dalam pelatihan *model*. Gambar asli dibagi menjadi *training* (70%), *validation* (10%), dan *test set* (20%). Hanya data *training* dan *validation* yang mengalami augmentasi, sedangkan *test* tetap menggunakan gambar asli. *Dataset* ini dapat langsung digunakan atau diterapkan metode augmentasi lain sesuai kebutuhan.