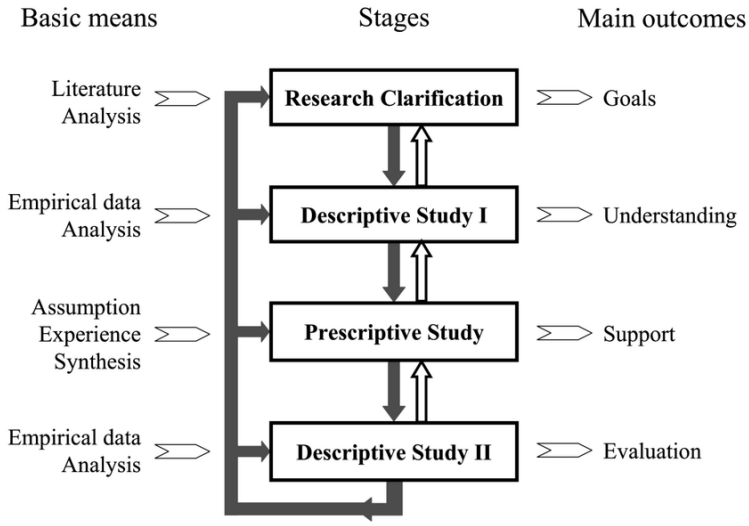
BAB III
METODE PENELITIAN

3.1 Desain Penelitian

Penelitian ini berfokus pada upaya pengoptimalan model *deep learning* untuk perangkat seluler, menggunakan data gambar media sosial X sebagai dataset, dan mengobservasi sejauh mana model dapat berkinerja dengan dataset yang lebih beragam. Model yang paling optimal di antaranya akan divalidasi secara empiris menggunakan dua *testing set* yang mengandung kumpulan gambar tidak terlihat oleh model sebelumnya; *testing set* simulasi dan *testing set* emulasi. Perbedaannya terletak pada *testing set* emulasi, di mana proses koleksi gambar tidak ditentukan proporsinya terlebih dahulu. Dengan pendekatan *Design Research Methodology* (DRM) sebagai kerangka kerja, prosedur penelitian meliputi empat tahap: Klarifikasi penelitian; Studi deskriptif I; Studi preskriptif; dan Studi deskriptif II (Blessing dan Chakrabati, 2009). Gambar 3.1.1 menggambarkan *pipeline* dari pendekatan DRM.



Gambar 3.1.1

Pipeline Kerangka Kerja *Design Research Methodology* (DRM)

Sumber: https://www.researchgate.net/figure/Design-Research-Methodology-Framework-Blessing-and-Chakrabarti-2009_fig4_325893060

3.1.1 Klarifikasi Penelitian

Untuk memperjelas konteks penelitian, masalah utama yang ingin diselesaikan merujuk pada metode dan temuan-temuan pada penelitian terdahulu (Bab 2.2). Sebagaimana terdapat pada Bab 1, tujuan dari penelitian ini menaungi rumusan masalah, meneliti bagaimana hasil dari proses *fine tuning* dan *hyperparameter tuning* masing-masing model memengaruhi performa model dengan memanfaatkan data gambar media sosial X. Turut diperhatikan bahwa subjek yang diteliti hanya berspesies kucing domestik (*felis catus*). Diagnosis perkembangan model sebelum dievaluasi pada *testing set* dibantu dengan sejumlah metode seperti *scatterplot, confusion matrix,* histogram, dan Grad-CAM untuk keterjelasan di balik mengenai keputusan yang diambil oleh model. Dari itu, literatur dibalik isu dan solusi hipotetis pada Bab 2 berfungsi sebagai landasan untuk memastikan bahwa solusi yang dikembangkan pragmatis dan dapat menjawab masalah yang diurai dengan valid. Adapun mengenai lanskap cakupan kemampuan model menyandingkan *testing set* simulasi dengan *testing set* emulasi. Akan tetapi, tetap menggunakan *testing set* simulasi untuk memberi perbandingan hasil yang setara dengan referensi ketiga penelitian terdahulu (Feighelstein, dkk., 2023; Namboonlue dan Sa-ing, 2024).

3.1.2 Studi Deskriptif I

Hasil kajian pemahaman terhadap teori-teori dan referensi-referensi penelitian pada Bab 2 dapat digunakan untuk menganalisis masalah desain model *deep learning* yang menggunakan dataset kecil sebagai input. Faktor-faktor tersebut diidentifikasikan sebagai keterbatasan dalam syarat kelayakan gambar, berakibat pada kuantitas dan kualitas dataset yang tidak cukup serta mengalami ketimpangan, arsitektur model yang kurang memadai, yaitu terlalu kompleks untuk jumlah dataset yang sangat kecil dibandingkan dengan dataset ImageNet. Sehingga memengaruhi hasil dari tindakan lanjut yang diambil, seperti teknik augmentasi yang kurang efektif. Kesenjangan antara kondisi yang diharapkan dan kondisi aktual disebabkan oleh perlakuan yang tidak sepadan terhadap pendekatan *deep learning* dibanding *machine learning*

(Feighelstein, dkk., 2023) dan teknik yang kurang memadai pada pelaksanaan *model training* (Feighelstein, dkk., 2023; Namboonlue dan Sa-ing, 2024). Hal ini dirinci secara sederhana pada Gambar 2.5 dan Gambar 2.6.

3.1.3 Studi Preskriptif

Desain solusi ditentukan berdasarkan hasil analisis Studi Deskriptif I, di mana dalam proses pengumpulan data hingga perancangan arsitektur model mengacu pada preferensi pengaturan model pada penelitian terdahulu sebagai titik awal, yang akan melewati penyesuaian secara sistematis mengikuti kerangka kerja yang telah dibentuk pada Bab 2.3. Dituju untuk mengatasi masalah desain tertera pada Studi Deskriptif I dalam proses menjawab rumusan masalah yang telah dirancang.

3.1.3.1 Pengumpulan Data

Proses pengumpulan data diotomatisasi dengan Jupyter Notebook, menggunakan API ntscraper untuk terhubung dengan X lalu mengikis data yang diperlukan. Environment REPL (*Read, Evaluate, Print, and Loop*) seperti format notebook digunakan untuk mengatasi masalah pemborosan daya komputasi jika menggunakan format *script*. Di mana saat inisasi objek pengikis (*scraper*), ntscraper mengecek library *instances*-nya terlebih dahulu untuk memitigasi error saat proses pengikisan. Hal ini membuat prosesnya iteratif dan sangat memakan waktu karena bersifat I/O bound yang bergantung pada stabilitas koneksi internet. Kualitas dan kuantitas data yang diperoleh tergantung pada kata kunci pencarian yang digunakan. Oleh karena itu, dilakukan studi semantik terlebih dahulu pada setiap kata kunci untuk menyaring kata kunci yang menimpa kata kunci lain serta bersifat ambigu. Hal ini mempermudah proses pengikisan data. Perlu diperhatikan bahwa kata-kunci yang digunakan mengikuti sifat platform media sosial X yang intuitif dan humoris.

Berdasarkan hasil yang ditunjukkan pada Tabel 3.1.3.1.1, kata kunci yang digunakan untuk kategori *no pain* terdiri dari "majikan", "mbabu", dan "muka". Sedangkan kata kunci yang digunakan untuk kategori *pain* hanya

"nyeri", dan "luka". Hal ini disebabkan oleh kata kunci seperti "tolong" dan "obat" sangat sering diikuti dengan kata kunci seperti "nyeri" dan "luka", di mana yang pilihan pertama lebih layak karena ketepatan relevansinya, sementara kata kunci "tolong" terkadang digunakan sebagai lelucon, contohnya "...tolong pap kucing kalian yang lucu itu". Kata kunci "pulih" dan "sembuh" dikecualikan karena bersifat ambigu yang dapat membahas topik proses kesembuhan dan rasa syukur atas kesembuhan.

Tabel 3.1.3.1.1

Kategori Kata Kunci Pencarian

Kata Kunci	Kategori	Keterangan
Majikan	*No pain*	Terkait dengan lelucon, sudut pandang pemilik kucing
Mbabu		Pelesetan dari "babu", sudut pandang kucing
Muka		Wajah pada umumnya
Bantu, donasi, obat, tolong	*Pain*	Cenderung ke arah lelucon dan permintaan bantuan
Pulih, sembuh		Cenderung ke konteks proses pemulihan dan sudah pulih
Luka, nyeri		Berkaitan langsung dengan rasa nyeri

Proses *scraping* data tidak terlepas dari peristiwa *rate limited*, maka dari itu, mekanisme sederhana dalam memitigasi terkena *rate limit* adalah menunda eksekusi *script* berikutnya atau dengan memperhatikan *server load* pada website *nitter instance health*. Proses *scraping* dilanjut dengan tanggal setelah tanggal terakhir data yang didapatkan untuk mencegah penimpaan data. Hasil *scraping* pada Tabel 3.1.3.1.2 diformat kedalam file CSV untuk digabungkan

menjadi satu berdasarkan target kategori (*no pain* dan *pain*) dan memperbaiki barisan data yang tidak konsisten atau tidak lengkap.

Tabel 3.1.3.1.2

Hasil *Tweet Scraping* Berdasarkan Kata Kunci

Kata Kunci	**Jumlah**	**Sumber (@akun)**
Majikan	1.460	kochengfs
	806	petsfess
Mbabu	1.185	kochengfs
	398	petsfess
Muka	744	kochengfs
	281	petsfess
Total kategori *no pain*	**4.874**	
Luka	608	kochengfs
	234	petsfess
Nyeri	773	kochengfs
	307	petsfess
Total kategori *pain*	**1.922**	
Total	**6.796**	

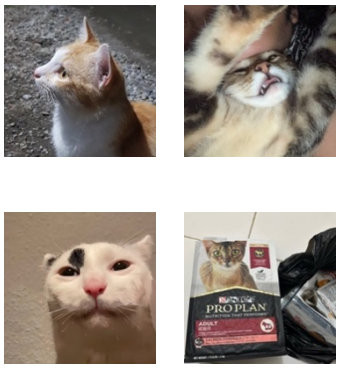
Catatan: Data dari 1 Januari 2022 sampai dengan 31 Agustus 2024.

Sebelum data diintegrasi, data dari setiap kata kunci dibersihkan dengan mengecualikan setiap baris yang mengandung kata-kata kunci kontradiktif lainnya, yaitu "luka", "muka", "mbabu", dan "majikan", serta kata-kata yang bersifat ambigu, misalnya "kayanya", "hilang", dan "meninggal". Selanjutnya data digabungkan dan diselaraskan formatnya (misalnya menghapus karakter spesial emoji, tanda kurung siku, dan tanggal waktu), menghapus data duplikat, dan mengidentifikasi kemungkinan anomali data. Dataset lalu divalidasi secara manual menggunakan Google Sheets, untuk memastikan kesesuaian data dengan ketentuan yang berlaku. Proses ini memverifikasikan integritas data, memastikan setiap nilai data berada dalam rentang dan format yang diharapkan, dan memeriksa setiap sampel untuk akurasinya. Adapun

ketentuan kelayakan gambar sebagai berikut, dengan contoh gambar yang tidak layak pada Gambar 3.1.3.1.1:

1. Gambar memiliki setidaknya satu kucing pada rentang usia dan jenis apapun, dengan penampakan wajah setidaknya tidak lebih dari satu *facial action unit* (FAU) dari FGS yang tidak dapat dinilai.
2. Kucing sedang tidak tidur
3. Gambar tidak merupakan jenis media meme dan gambar sampul
4. Tweet tidak mengandung kata-kata ambigu penyebab tidak adanya korelasi langsung dengan rasa nyeri kucing, seperti "kayanya kucingku sedang nyeri.." dan "tangan sender nyeri abis digigit".

Proses terkahir dalam pengumpulan data adalah memetakan setiap data dengan label yang berkaitan, 0 untuk mengindikasikan tidak nyeri (*no pain*) dan 1 untuk mengindikasikan nyeri (*pain*). Penentuan jumlah data yang akan digunakan melihat kategori *pain* sebagai tolak ukur karena berfungsi sebagai pembeda kategori.



Gambar 3.1.3.1.1
Contoh Gambar Tidak Layak

Gambar-gambar yang telah disaring lalu melalui validasi oleh seorang dokter hewan menggunakan FGS. Ketentuan yang berlaku mengikuti panduan resmi FGS, di mana jika ada lebih dari satu *action unit* yang tidak dapat dinilai karena tertutup atau tidak lolos ketentuan kelayakan lingkungan penelitian, gambar didiskualifikasi. Sedangkan *threshold* kategori *no pain* ialah yang

memiliki skor rata-rata di bawah 3.9 dan kategori *pain* memiliki skor lebih dari atau sama dengan 3.9 (Evangelista dkk., 2019).

3.1.3.2 *Data Preprocessing*

Wajah kucing pada gambar yang akan digunakan melalui *preproccessing* terlebih dahulu untuk memastikan kemudahan model dalam mempelajari fitur-fiturnya dengan optimal. Teknik *preprocessing* meliputi penyelarasan wajah dan *downsampling* resolusi saat di*crop* ke resolusi 224 x 224 pixel. Hal ini untuk menghindari *upsampling* gambar beresolusi kecil sehingga menimbulkan *noise* yang menghancurkan fitur gambar. Penyelarasan wajah memanfaatkan algoritma *Haar Feature-based Cascade Classifier* untuk mendeteksi wajah kucing dan memotongnya secara otomatis. Sehingga gambar hasilnya tetap mencakup ciri data bawaan media sosial, dalam hal ini, wajah dengan rotasi kepala yang beragam. Namun, model *Cascade Classifier* yang tersedia tidak cukup optimal, menyebabkan beberapa gambar tidak terdeteksi wajahnya dan target wajah meleset. Sehingga proses dilanjut dengan menyejajarkan dan memotongnya secara manual.

Dataset *training* dibagi secara acak dan proporsional dengan rasio 4:1:1 (*training*, *validation*, dan *testing*), lima jumlah lipatan (k = 5) pada teknik *k-fold cross-validation* yang digunakan, 80% dalam *training set,* dan 10% lainnya dalam *validation set* dan *testing set*. Masing-masing subjek *training set* tidak saling menimpa dan ditempatkan pada posisi dan batch yang berbeda di setiap lipatannya (*fold*). Jumlah *fold* dua kali lebih kecil dibanding metode Feighelstein dkk. untuk mempercepat dan menghemat daya komputasi berbasis *cloud*. Hal ini tidak mempengaruhi nilai-nilai *weights* dan *bias* performa akhir model. Teknik *k-fold cross-validation* membantu dalam mengevaluasi metrik performa secara kokoh dan memahami kinerja model, memberi gambaran bagaimana model akan bekerja pada pengaturan yang realistis. Tabel 3.1.3.2.1 menjelaskan pembagian distribusi data.

Tabel 3.1.3.2.1

Distribusi Jumlah Gambar pada *Training Set*, *Validation Set*, dan *Testing Set*

Set	Proporsi (*no pain* / *pain*)	Jumlah Gambar
Training	57 / 57	114
Validation	7 / 7	14
Testing	7 / 7	14
Total	**142**	

Adapun dataset untuk menguji coba model pada lingkungan yang nyata, menggunakan dataset wajah kucing dari archive.org, mengambil sejumlah gambar yang tidak mempunyai label nyeri atau tidak. Gambar-gambar tersebut dipilah khususnya untuk mengeleminasi gambar yang tidak dapat dinilai wajahnya, lalu kumpulan gambar-gambar tersebut diberi label sesuai dengan skala nyerinya. Tabel 3.1.3.2.2 menunjukkan dataset yang dipakai untuk menguji coba model terbaik, tanpa dan dengan augmentasi, pada lingkungan yang nyata dengan ciri-ciri yang serupa, mengikuti ketentuan kelayakan gambar penelitian.

Tabel 3.1.3.2.2

Distribusi Jumlah Gambar pada Dataset Emulasi Dunia Nyata

Kategori	Jumlah Gambar
No pain	95
Pain	5
Total	**100**

Sumber: https://archive.org/details/CAT_DATASET

Teknik augmentasi yang digunakan mengikuti teknik Feighelstein dan Namboonlue, yaitu *Random Resized crop, Random Horizontal Flip,* dan *Random Rotation*; sedangkan teknik Namboonlue meliputi *Random Horizontal Flip, Random Rotation,* dan *Random Autocontrast.* Perlu dicatat, karena sifat gambar dari media sosial X bersifat sembarang, yakni ukuran resolusi yang berbeda, harus memuat fungsi untuk menangani peristiwa ini, yaitu menjaga aspek rasio asli gambar untuk menghindari *upscaling* pada

gambar beresolusi kecil agar tidak menimbulkan *noise.* Gambar 3.1.3.2.1 menunjukkan contoh sampel yang telah diaugmentasi dengan dua teknik yang berbeda.

Gambar 3.1.3.2.1

Contoh Sampel Data yang Diaugmentasi

Catatan: atas = teknik Namboonlue; bawah = teknik Feighelstein

Tabel 3.1.3.2.3 merangkum perbedaan yang terdapat dalam metode akuisisi data, khususya, pada sandingan tiga penelitian lain yang mencakup *domain* serta *subjek* yang sama. Yaitu klasifikasi nyeri pada kucing domestik.

Tabel 3.1.3.2.3

Perbandingan Metode Akuisisi Data

Penelitian	Sumber	Deskripsi	Jumlah	SP:JP	Jumlah Final
Feighelstein, dkk., 2022	Universitas	Seragam, terkontrol	464	MCPS:1	Simulasi dunia nyata
Feighelstein, dkk., 2023			84	CMPS:>1	
Namboonlue dan Sa-ing, 2024	Klinik hewan		114	CSU-FAPS:>1	
Ayyash, M. Y., 2025	**Media sosial X**	**Beragam, acak**	**142**	**FGS:1**	**Simulasi dan emulasi dunia nyata**

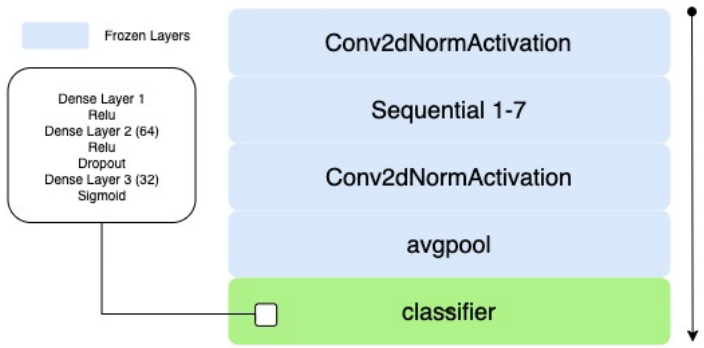
Catatan: SP:JP = sistem penilaian : jumlah penilai; jumlah gambar pada penelitian Namboonlue dan Sa-ing (2024) sebelum augmentasi

3.1.3.3 Arsitektur Model

Pemilihan jenis model selain ResNet50 mempertimbangkan ukuran dataset yang digunakan, menyediakan kapasitas yang efisien, namun panda dan akurat. Sehingga, dalam penelitian ini menggunakan dua jenis model, EfficientNet B3 dan ResNet18. Pemilihan tiga model tersebut bertujuan untuk dapat melakukan perbandingan yang sebanding. ResNet18 (~12 juta parameter) sebagai model yang lebih kecil dibanding ResNet50, setelahnya, EfficientNet B3 (~12 juta parameter) sebagai model yang lebih kecil dari pada EfficientNet B7 (~66 juta parameter) dan memiliki jumlah parameter yang kurang lebih setara dengan ResNet18. Model yang lebih sederhana secara eksponensial dibanding EfficientNet B0 (~5.3 juta parameter), untuk melihat kemungkinan apakah jika menggunakan model EfficientNet varian B yang lebih kecil berkemungknan menghasilkan akurasi yang lebih baik dengan

kapasitas kemampuan yang lebih besar, sekaligus melihat kapabilitas varian EfficientNet yang lebih sederhana.

Secara teori, EfficientNet B3 akan lebih berkinerja lebih bagus untuk dataset ini, mampu mempelajari data lebih rinci dengan cepat. ResNet18 turut serta memiliki kapabilitas yang bagus, akan tetapi, kedalaman *layer* tidak cukup untuk mendukung kapabilitas yang lebih canggih. Ketiga model yang telah ditentukan akan memiliki kerentanan terhadap terjadinya *overfitting* lebih kecil dan proses *fine tuning* yang lebih sedikit dibandingkan model-model yang digunakan pada penelitian terdahulu. Metode *compound scaling* dan infrastruktur yang efisien pada EfficientNet mengalahkan kapabilitas ResNet18, melalui *compound scaling* yang membantunya mencapai — tidak hanya jumlah kedalaman *layer* yang lebih efektif — akan tetapi juga jumlah *channel* dan input — spesifiknya, struktur *depthwise seperable convolutions* dan *inverted bottleneck*, dibandingkan ResNet18 yang hanya mengandalkan kedalaman *layer*-nya, untuk tugas yang meliputi pengenalan perbedaan yang sangat halus antara kucing yang sehat dan yang sedang nyeri. Model EfficientNet B3 yang akan digunakan sebagai tahap inisial *fine tuning* di antara ketiga model, mengikuti pengaturan arsitektur dan *hyperparameter* terbaik yang diungkapkan oleh Namboonlue dan Sa-ing (2024), seperti pada diagram di gambar 3.1.3.3.1.

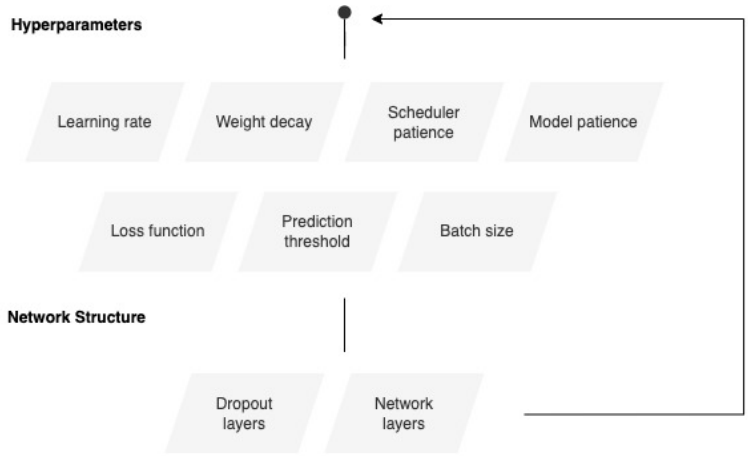


Gambar 3.1.3.3.1

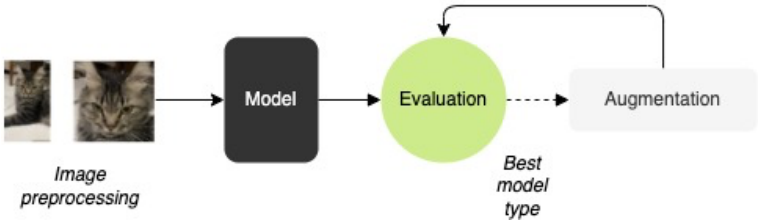
Pengaturan Inisial Model EfficientNet B3

3.1.3.4 *Fine Tuning*

Proses *fine tuning* model dilakukan pada struktur model dan *hyperparameter* yang terlibat secara empiris dan iteratif, seperti yang ditunjukkan pada Gambar 3.1.3.4.1 dan 3.1.3.4.2.



Gambar 3.1.3.4.1
Diagram Alur *Fine Tuning*



Gambar 3.1.3.4.2
Diagram Alur *Model Training*

Pengaturan dasar uji coba *fine tuning* merujuk pada yang digunakan oleh Namboonlue seperti yang telah dikaji pada Bab 2.2.3. Adapun proses *training* dibagi menjadi dua fase, yaitu tanpa augmentasi data dan dengan augmentasi data. Awal mula, dipastikan lingkungan pengembangan bersifat *deterministic,* mampu mereproduksi hasil yang konsisten pada *runtime* yang berbeda dengan menentukan jumlah *seed*. Alur proses *training* model mengikuti skema pada gambar 3.1.3.4.2. Model mempelajari data gambar yang telah di-*preprocess,*

mengevaluasi kinerjanya, lalu hanya satu jenis model yang terbaik di antara setiap jenisnya akan diaugmentasi. Model teraugmentasi pada akhirnya dibandingkan dengan model tanpa augmentasi. Dalam proses *fine tuning,* terlepas dari nilai *loss* dan akurasi terbaik, model yang berhasil ke tahap selanjutnya dan diperbaiki lagi setelannya adalah model dengan kurva loss yang menunjukkan adanya indikasi *convergence*, mewakili gambaran ideal bagaimana progres yang baik seharusnya. Tindakan selanjutnya berdasarkan perbaikan *convergence* kurva *loss*, mengindikasi perbaikan kemampuan model dalam mempelajari data, yang akan menjamin performa akurasi yang lebih baik.

Hal penting yang turut diperhatikan setelah itu ialah nilai *threshold* akurasi. Model tidak akan selalu yakin sepenuhnya dengan pasti mengenai prediksinya, sehingga nilai *logits* probabilitas dapat mengambang pada interval tertentu. Nilai *threshold* ditetapkan dengan mengambil nilai *mean* dari distribusi nilai *logits* klasifikasi hasil model, jika penetapan dengan nilai *mean* tidak memberi nilai *precision* dan *recall* yang relatif sama besar, maka akan diiterasi ulang, menyesuaikan nilai *threshold* dengan menambahkannya berdasarkan *standard deviation* dari distribusi *logits*-nya. Hal ini berpengaruh terhadap graf metrik akurasi, *precision,* dan *recall*. Jika tidak disesuaikan dengan benar maka graf pun akan memproyeksi performa yang tidak sesuai. Statistik nilai akurasi terbaik diambil dari tahap validasi, relatif dengan titik terendah nilai *loss*. Reproduktifitas model diimplementasikan dengan menetapkan modul *deterministic* dan *benchmark* dari library *deep neural network* NVIDIA Pytorch pada *True* dan *False*. Hal ini memaksa model untuk menggunakan algoritma proses *machine learning* yang sama di berbagai *runtime* dan *hardware* yang berbeda. Tabel 3.1.3.4.1 merangkum perbedaan metode-metode yang terlibat pada masing-masing pendekatan dalam proses pelatihan model.

Tabel 3.1.3.4.1

Perbandingan Metode *Model Training*

Penelitian	*Optimizer*	*Scheduler*	Teknik Validasi	Teknik Augmentasi	Model
Feighelstein dkk., 2022	Adam	–	LOSO-CV	*Crop, resize, horizontal flip, rotation*	ResNet50
Feighelstein dkk., 2023			10 Fold CV		
Namboonlue dan Sa-ing, 2024	SGD, RMSProp, Adam		*Validation set*	*Crop, resize, horizontal flip, rotate, contrast*	EfficientNet B7, ResNet50
Ayyash, M. Y., 2025	**SGD, Adam, AdamW**	**CyclicLR**	**5 Fold CV**	***Crop, resize, horizontal flip, rotate, contrast***	**EfficientNet B3, ResNet18**

3.1.3.5 *Training*

Progres *training* dilanjut dengan menggunakan pengaturan *benchmark* yang telah ditentukan saat *fine tuning* untuk perbandingan kinerja antarmodel pada langkah-langkah selanjutnya. Dalam fungsi pelatihan model, teknik *early stopping* diimplementasi guna memitigasi terjadinya overfitting berkelanjutan dan menghemat daya komputasi. Hal ini dilakukan dengan memantau berapa lama loss validasi tidak membaik selama sejumlah *epoch* yang telah ditentukan. Teknik lainnya yang digunakan untuk memitigasi *overfitting* yaitu melibatkan *layer dropout*, *weight decay* pada *optimizer*, dan *learning rate scheduler* pada *optimizer* yang mengeksplor *learning rate* pada interval tertentu. Jenis *learning rate scheduler* akan bereksperimen seputar

ReduceLROnPlateau dan CyclicLR. Saat progres pelatihan sudah mencapai batas *epoch* optimalnya, *model state*, *optimizer state, dan scheduler state* disimpan untuk dilanjut dengan data yang telah diaugmentasi. Hal ini membolehkan model untuk melanjutkan proses *training* dengan memanfaatkan *weights* yang telah dipelajari.

3.1.4 Studi Deskriptif II

Model dengan kinerja terbaik secara keseluruhan ditentukan oleh yang melalui teknik augmentasi dan mengalami perbaikan pada evaluasi menggunakan *testing set* emulasi, tentunya telah melewati evaluasi dengan *testing set* simulasi. Hal tersebut menguatkan performa yang tergambar oleh teknik *k-fold cross validation* saat *training*, memberi interpretasi kinerja model yang lebih memadai.

3.1.4.1 Evaluasi

Model terbaik dari ketentuan sebelumnya lalu dievaluasi keandalannya dalam emulasi lingkungan yang nyata, mengklasifikasi dataset gambar kucing di mana label diberikan setelah prediksi. Karena sedikitnya gambar yang layak pada *dataset,* jumlah gambar pada dataset ini mengambil nilai yang lebih besar di antara penelitian-penelitian rujukan dengan *domain* yang serupa, agar variabilitias interval *confidence* (keyakinan) lebih sempit. Dari itu, jumlah *test set* pada penelitian Namboonlue dan Sa-ing (2024), sebanyak 100 gambar wajah kucing digunakan agar dapat mencapai hasil yang sebanding. Hal ini dapat dilihat pada Tabel 3.1.4.1.1 di mana proporsi sesuai dengan *training-validation-test split* yang dipakai oleh masing-masing lingkungan penelitian.

Tabel 3.1.4.1.1

Perbandingan Jumlah Gambar pada *Test Set*

Penelitian	Jumlah Gambar
Feighelstein, dkk., 2022	18 dari 464
Feighelstein, dkk., 2023	8 dari 84
Namboonlue dan Sa-ing, 2024	**100 dari 1140**

3.2 Instrumen Penelitian

Dalam tahap pengumpulan data, integritas data yang selaras dengan ketentuan penelitian ini dibutuhkan untuk memvalidasi bahwa data yang diperoleh bersifat sah, tidak bertentangan dengan praktik penelitian pada lazimnya, yang mengakibatkan negasi hasil penelitian hingga ke tahap awal. Proses tersebut menggunakan teknik studi semantik pada hasil *tweets* yang diperoleh dari masing-masing kata kunci menggunakan *script* python dan Google Sheets untuk pemeriksaan kembali. Pemeriksaan ulang kelayakan data dilakukan secara manual, walaupun memakan waktu, memastikan kredibilitas data dengan pasti. Nilai kebenaran data yang telah dikategorikan diverifikasi ulang dengan mmelibatkan seorang dokter hewan untuk menilai skala nyeri masing-masing subjek dataset dengan FGS. Data yang diketahui memiliki nilai FGS yang tidak sesuai dengan kategorinya didiskualifikasi atau dipindahkan ke dalam kategori sesungguhnya. Adapun spesifikasi software beserta *library* python yang digunakan dalam keseluruhan proses pengumpulan data tertera pada Tabel 3.2.1.

Tabel 3.2.1

Spesifikasi Lingkungan Pengembangan dalam Pengumpulan Data

Formal Language	***Version***
Python	3.11
Software	***Version***
Google Sheets	2024
Zsh Terminal	5.9
Python Library	***Version***
concurrent.features	3.10
ntscraper	0.1.7
Numpy	1.25.3
os	3.10
Pandas	2.0.3
PyTorch	2.6.0
requests	2.30.0

3.2.1 Alat dan Bahan Penelitian

Seluruh pelaksanaan penelitian dilakukan menggunakan perangkat pribadi dan pada beberapa layanan *cloud computing*. Alur pengembangan menganut pada model *Iterative Workflow,* di mana fitur pada setiap fungsi yang digunakan untuk melatih model berdasarkan skala dataset dan kebutuhan analisis untuk perbaikan nilai *hyperparameter* dan struktur model, dengan mengalokasikan kegunaannya secara dinamis dan berdasarkan observasi. Seluruh komponen *hardware* dan *software* terlibat dalam optimasi model tertera pada Tabel 3.2.1.1.

Tabel 3.2.1.1

Spesifikasi *Hardware* dan *Software* Lingkungan Penelitian

Hardware	***Specification***
Google Colab Pro GPU	NVIDIA Tesla 4, 16GB GDDR6
Laptop Macbook Pro 13 2017	2,3 GHz Dual-Core Intel Core i5 8 GB RAM 2133 MHz LPDDR3 macOS Ventura 13.7.1
Software	***Version***
Google Colab	2024

3.3 Metode Analisis Data

Tahap awal penelitian melibatkan metode eksplorasi data dan *data preprocessing* untuk menangani nilai yang kosong, *outlier*, dan inkonsistensi dalam data dengan memastikan data berada dalam format yang benar untuk analisis. Contohnya, *label encoding,* melakukan *mapping* dari representasi kategori format string ke format numerik. Proses *fine tuning* dibantu oleh evaluasi model menggunakan visualisasi data dari metrik *loss,* akurasi, *precision, recall*, *F1-score, PR curve*, dan histogram dari seluruh *fold*. Tahap akhir, *testing*, memanfaatkan metode Grad-CAM untuk menganalisis area fitur wajah yang digunakan oleh model untuk pertimbangannya dari model yang terbaik. Hal ini lalu dapat dijadikan bahan referensi untuk penelitian selanjutnya.