

## **BAB III METODOLOGI PENELITIAN**

### **3.1 Subyek dan Objek Penelitian**

Subjek pada penelitian ini adalah taksonomi hewan. Adapun objek dalam penelitian ini menggunakan gambar 90 jenis hewan yang nantinya dijadikan sebagai *dataset* pada penelitian dalam pembuatan model yang dapat melakukan klasifikasi berdasarkan kelas yang ada serta memberikan informasi mengenai taksonomi genus.

### **3.2 Alat dan Bahan Penelitian**

#### **Alat:**

1. Laptop MSI (CPU : Intel I7-9750H, 2.6 GHz, RAM: 16.00 GB, OS: Windows 11-64 bit).
2. *Notebook* Google Collab.
3. Python *programming language*.
4. *Library* Tensorflow, Sklearn, Open-CV, Numpy, Matplotlib, Keras, Pandas, Microsoft Office, dan Numpy.

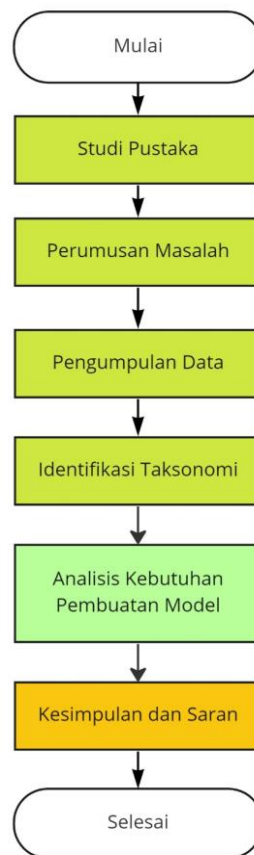
#### **Bahan:**

Pengambilan dataset dilakukan melalui cara mendownload dari situs Kaggle <https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals>, dengan jumlah gambar sebanyak 5400 dengan 90 jenis hewan, setiap jenis hewan ada 60 gambar.

### **3.3 Diagram Alir Penelitian**

Pada penelitian ini, variabel penelitian terdiri serangkaian langkah yang dilakukan dalam proses penelitian dari studi pustaka yang telah digunakan sebelumnya, pengumpulan dataset hewan dengan jenis berbeda dan klasifikasi taksonomi genus hewan, mengidentifikasi taksonomi sesuai dataset, proses

analisis kebutuhan pembuatan model, implementasi model, dan kesimpulan serta saran. Gambar 3.1 menunjukkan diagram alir penelitian. Tahap investigasi yang akan dilakukan untuk menghindari kesalahan yang dapat menyebabkan pada tahap selanjutnya:



Gambar 3. 1 Diagram Alir Penelitian

a. Studi Pustaka

Berdasarkan penelitian yang akan dilakukan, mengacu pada referensi dari beberapa penelitian untuk mengetahui hubungan relevan antara penelitian yang akan dilakukan dengan penelitian terdahulu, membantu untuk menghindari adanya suatu kesamaan dalam penelitian yang akan dilakukan.

b. Perumusan Masalah

Penelitian ini dilakukan dengan permasalahan utama tentang keterbatasan kemampuan manusia dalam menghafal dan mempelajari segala

hal, terutama mengidentifikasi taksonomi hewan. Dengan memperhatikan aspek batasan masalah, tujuan penelitian dan manfaat penelitian.

c. Pengumpulan Data

Pengumpulan dataset diambil melalui situs Kaggle.

d. Identifikasi Taksonomi

Proses pengumpulan informasi taksonomi mulai dari kingdom, filum, kelas, ordo, famili, dan genus. Informasi dilakukan sesuai dengan jenis 90 hewan yang ada pada dataset Kaggle.

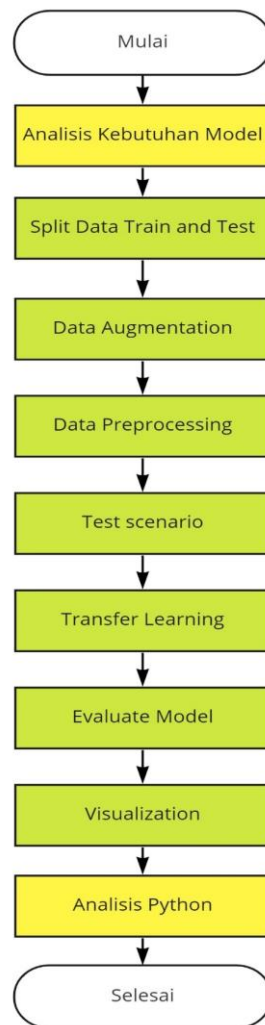
e. Analisis Kebutuhan Pembuatan Model

Proses analisis performa model mulai dari diagnosa permasalahan, analisis kebutuhan model, bagaimana membuat model yang optimal, dan proses analisis *clean code*.

f. Kesimpulan dan Saran

Kesimpulan dibuat setelah dilakukan penelitian sehingga hasil merupakan objektif dari hasil yang telah diteliti, lalu selanjutnya saran terhadap kendala dan kekurangan penelitian ini, sehingga perlu adanya perbaikan untuk pengembangan dimasa yang akan datang.

Setelah mengetahui tahapan diagram alir dilanjutkan tahapan pembuatan model yang ditunjukkan pada Gambar 3.2. Tahap-tahap ini meliputi analisis kebutuhan dalam pembuatan model dalam penelitian:



Gambar 3. 2 Diagram Alir Pembuatan Model

1. Analisis kebutuhan Model

A. *Split Data Train and Test*

Pembagian dataset menjadi *training* untuk pelatihan dan test untuk validasi akurasi model, penerapan akan mengimplementasikan (80% *training* dan 20% *testing*) yang berarti data pelatihan sebanyak 4320 citra dan data validasi sebanyak 1080

B. Data Augmentation

Augmentasi data adalah proses pengolahan data citra digital, sedangkan augmentasi merupakan proses mengubah atau memodifikasi suatu citra sehingga komputer mendeteksi bahwa citra yang diubah merupakan citra

yang berbeda seperti mengubah kemiringan, dan memutar gambar secara horizontal.

#### C. Data Preprocessing

Data preprocessing adalah teknik yang digunakan mengumpulkan data mentah dalam format, tempat, format warna, dan pengubahan ukuran gambar agar berguna serta efisien untuk pelatihan.

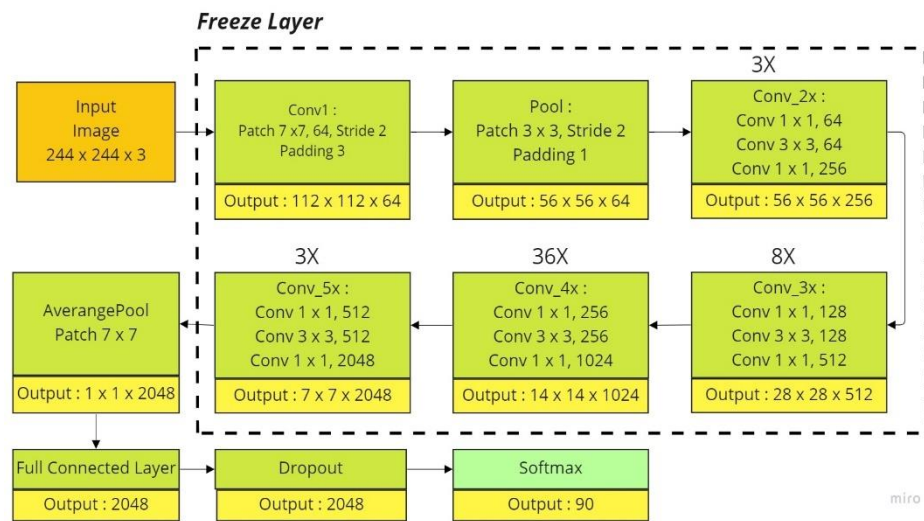
#### D. Test Scenario

Serangkaian beraga uji coba dalam *hyper-parameter* terutama pada *batch\_size* dan *dropout* yang telah ditetapkan untuk melakukan training model sehingga hasilnya dapat melakukan kalsifikasi taksonomi genus hewan.

#### E. Transfer Learning

*Pre-Trained ResNet 152*, menggunakan model yang pernah dilatih kemudian diterapkan untuk *training* pada objek klasifikasi baru sehingga memiliki pengetahuan lebih mendalam dengan tujuan menghindari penggunaan *layer* berlebihan dan *undefitting*. Contoh Perhitungan dalam Transfer Learning dari *inputan* lalu masuk ke dalam arsitektur ResNet 152 dengan arsitektur dalamnya *conv\_1* kernel 7 x 7 filter 64 *stride 2 Padding 3* dengan *input* 224 x 224 x 3 keluaran 112 x 112, lalu dilanjutkan dilanjutkan masuk ke *conv\_2x* dengan *max pooling 3 x 3 stride 2 padding 1*, dengan *residual block* 3 kali keluaran 56 x 56 dimensi, lalu masuk ke *conv\_3x* dengan *residual block* 8 kali keluaran 28 x 28 dimensi, lalu masuk ke *conv\_4x* dengan *residual block* 36 kali keluaran 14 x 14 dimensi, selanjutnya masuk ke *conv\_5x* dengan *residual block* 3 kali keluaran 7 x 7 dimensi, kemudian *Average Pooling 2D*, lalu *Dropout layer*, lalu *Fully connected layer*, dan terakhir masuk *dense layer* 90 (sesuai target 90 kelas ) dengan *softmax activation layer*. Dalam skema penelitian akan digunakan:

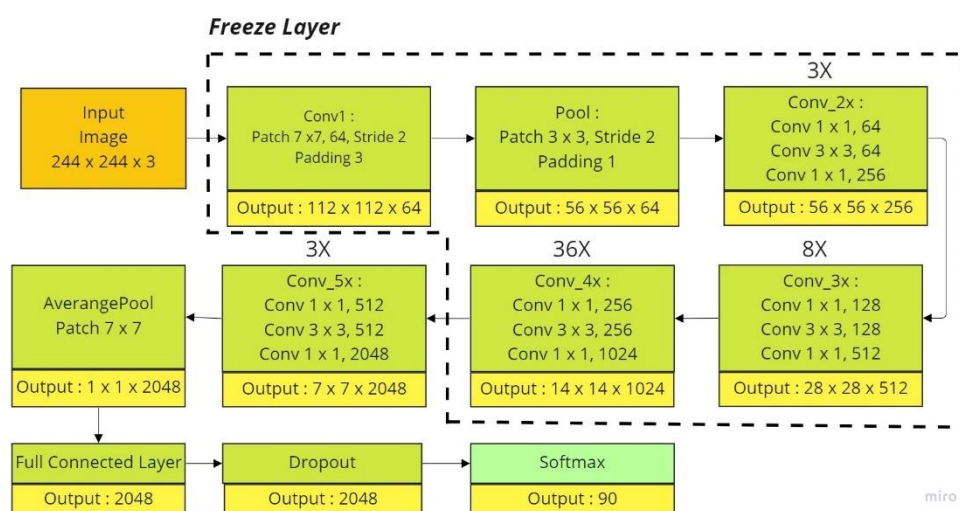
## a. Skema Model A:



Gambar 3. 3 Skema Model A Freeze Layer

Berdasarkan Gambar 3.3 skema arsitektur model A akan menggunakan *freeze layer* seluruhnya mulai dari Conv1, Conv\_2x, Conv\_3x, Conv\_4x, dan Conv\_5x untuk arsitektur pelatihan sebagai implementasi Transfer Learning ResNet 152. *Freeze layer* bertujuan menyimpan nilai *weights* sehingga mempertahankan representasi fitur yang sudah dipelajari dalam model pre-trained untuk digunakan pada objek baru.

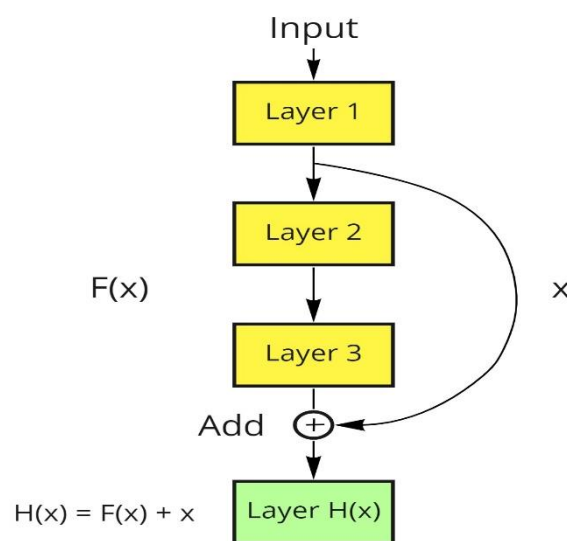
## b. Skema Model B:



Gambar 3. 4 Skema Model B Freeze Layer

Berdasarkan Gambar 3.4 skema arsitektur model B akan menggunakan *freeze layer* seluruhnya mulai dari Conv1, Conv\_2x, Conv\_3x, dan Conv\_4x kecuali block 3x terakhir (Conv\_5x), maka Conv\_5x dalam mode *trainable* yang berarti memperbaiki *weights* sehingga memungkinkan model memperbaiki representasi fitur sesuai dengan data baru pada *dataset*.

Skema arsitektur A dan B sebagai implementasi Transfer Learning ResNet 152. Adapun rumusan matematika yang dibangun diatas ResNet diperlihatkan pada Gambar 3.5.



Gambar 3. 5 Residual Network

- Residual Network, oleh peneliti [19], dengan persamaan dengan (3.1)
 
$$H(x) = F(x) + x \quad (3.1)$$
 dengan  $H(x)$  adalah *Weight layer* yang diharapkan, hasil dari  $F(x) + x$ .  $F(x)$  adalah *Weight layer* dan  $x$  adalah *Identity*.
- Pooling layer* adalah pengurangan ukuran matriks *input* dengan cara mengurangi jumlah parameter, dengan 3 jenis *min pooling*, *max pooling*, dan *average pooling* [13].
- Convolution layer* merupakan lapisan *inputan* pengaktifan ekstrasi fitur pada sebuah gambar dengan perhitungan konvolusi *filter* [28], dengan persamaan(3.2)

$$h(x) = f(x) * g(x) \quad (3.2)$$

dimana  $h(x)$  merupakan *weight parameter* dari perhitungan konvolusi  $f(x)$  dan  $g(x)$ ,  $f(x)$  merupakan *inputan* citra dan  $g(x)$  adalah *filter*.

- d. *Batch normalization* merupakan metode konversi *output layer* dengan normalisasi, teknik ini mempercepat proses belajar karena tidak adanya nilai *activation* yang terlalu tinggi ataupun terlalu rendah, berperan sebagai *regularizer* atau *Dropout* [13], *layer* 4 dimensi *tensor*, dengan perumuan  $b$  adalah *Batch*,  $c$  adalah *channel*, dan 2 spasial dimensi adalah  $x$  dan  $y$  [31], dengan formula (3,3)

$$o_{b,c,x,y} = \gamma \frac{I_{b,c,x,y} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}} + \beta_c \quad (3.3)$$

Dengan turunan aktivasi rata-rata dari *Batch normalization*, persamaan (3.4)

$$\mu_c = \frac{1}{|\beta|} \sum_{b,x,y} I_{b,x,y} \quad (3.4)$$

Dari seluruh inputan BN membagi aktivasi terpusat dengan standar deviasi, selama pengujian rata-rata dan varians digunakan, normalisasi akan ditransformasikan kedalam parameter  $\gamma_c$ ,  $\beta_c$  selama pelatihan.

- e. *Rectified Linear Unit* (ReLU) sebagai *activation layer* dengan fungsi memetakan nilai *negative* ke 0 dan mempertahankan nilai *positive* [28], dengan persamaan (3.5)

$$f(x) = \max(0, x) \quad (3.5)$$

Memiliki nilai terendah 0, maka *activation layer* jika  $x < 0$  nilai 0 atau jika  $x > 0$  nilai tetap  $x$

- f. *Fully connected layer* adalah *layer* penghubung seluruh *layer* yang disebut *feature map* kemudian dimasukkan ke *FC-layer* [28], dirumuskan dengan persamaan (3.6)

$$h(x) = g(b + \sum_i w_i x_i) \quad (3.6)$$

$g$  merupakan *activation layer*,  $b$  adalah bias,  $w_i$  sebagai nilai *input*, dan  $x_i$  nilai bobot

- g. *Dropout layer* adalah metode regulisasi dengan memilih *random neuron* pada *neural networks* agar *neuron* tidak dipakai selama



pelatihan, maka *neuron* yang dipilih tersebut akan diberhentikan dan *weights terupdate* tidak diterapkan pada *neuron* pada saat melakukan Backpropagation dengan tujuan menghindari *overfitting* pada saat pelatihan [16]. Proses Feedforward dilakukan dengan persamaan (3.7)

$$\begin{aligned} y^{\sim 1} &= r_j^l * y^l \\ z_i^{l+1} &= W_i^{(l+1)} y^l + b_i^{(l+1)} \\ y_i^{l+1} &= f(z_i^{(l+1)}) \end{aligned} \quad (3.7)$$

variabel  $y^l$  adalah nilai *output layer l*, variable  $z^l$  adalah nilai input *layer l*, variabel  $W^l$  dan  $b^l$  merupakan *weights layer l*, memakai unit  $i$  dengan fungsi aktivasi  $f$  dan  $r^l$  vector sampai  $j$  menyimpan nilai yang diperoleh dari distribusi Bernoulli.

- h. *Softmax activation layer* menghasilkan nilai yang ditransformasikan kedalam *probabilitas* belum dinormalisasikan sehingga nilai tiap kelas dilakukan fungsi *softmax* untuk mendapatkan hasil klasifikasi [32], dengan persamaan (3.8)

$$y_{ijk} = \frac{e^{x_{ijk}}}{\sum_{t=1}^D e^{x_{ijt}}} \quad (3.8)$$

$y_{ijk}$  merupakan *vector* nilai antar 0 sampai 1 dan  $x$  adalah *vector* berisi *FC-layer* terakhir.

Dari skema diatas akan digunakan dengan *peng-compile model*.

- i. Optimizer Adam yang merupakan gabungan RMSprop dan Momentum,

$$\theta_{t+1} = \theta_t - \frac{\partial}{\sqrt{\hat{v}_t + \varepsilon}} \cdot \hat{m}_t \quad (3.9)$$

$\theta_{t+1}$  hasil dari parameter baru,  $\theta_t$  parameter sebelum diperbarui,  $\partial$  adalah *learning rate*,  $\hat{m}_t$  *gradient* kuadrat orde 1,  $\hat{v}_t$  *gradient* orde 2, dan  $\varepsilon$  merupakan skala kecil pencegah pembagian dengan nilai 0, perhitungan error bergantung langsung dengan *gradient* orde 1 dan 2.

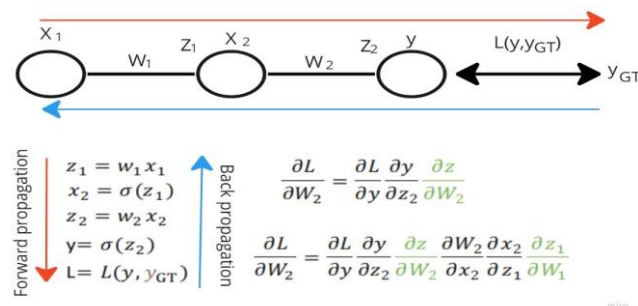
- j. *Loss* merupakan parameter untuk menghitung data yang tidak terdeteksi selama pelatihan dan pengujian, Penelitian ini menggunakan

*Categorical Cross Entropy* sebagai optimasi nilai *loss*, dengan formula (3.10)

$$H(p, q) = -\sum_{i=1}^N p_i \log(q_i) \quad (3.10)$$

$H$  merupakan nilai *loss*,  $p$  nilai asli,  $q$  nilai perkiraan,  $p_i$  merupakan nilai dari hasil probabilitas asli (kelas) dan  $q_i$  nilai probabilitas perkiraan (softmax) [33].

- k. Gradient Descent adalah teknik pengoptimalan menggunakan gradien yang dihitung dari hasil Backpropagation untuk memperbarui *weight* dan bias dengan tujuan meminimalkan *loss*.



Gambar 3. 6 Cara Kerja Gradient Descent

Dengan menemukan titik minimum dari fungsi dengan mengikuti gradient negatif secara berulang, persamaan (3.11)

$$W = W - \eta \frac{\sigma L}{\sigma W} \quad (3.11)$$

Dimana nilai  $W$  hasil update,  $\eta$  *learning rate*, dan  $\frac{\sigma L}{\sigma W}$  nilai *loss gradient* [34].

1. *Metrics accuracy* merupakan metrik yang digunakan untuk mendapatkan *nilai loss validation* dan *acc validation* setiap iterasi *training* dari data test, sehingga dapat diketahui performa model ketika visualisasi apakah model underfitting atau overfitting.

#### F. *Evaluation Model*

Setelah model dibuat dilakukan evaluasi menggunakan *data test* untuk mendapatkan *accuracy validation* dengan tujuan pengoptimalan akurasi model lebih baik dan efektif ketika digunakan untuk prediksi objek baru.

Dengan formula akurasi uji kebenaran, persamaan (3.12)

$$A = \frac{b}{n} \times 100\% \quad (3.12)$$

$A$  adalah nilai presentasi kebenaran,  $b$  adalah jumlah data benar, dan  $n$  merupakan jumlah seluruh data

#### G. *Visualization*

Dilakukan visualisasi dengan melakukan *line plotting* nilai dari *train loss*, *train accuracy* dan *validation, loss validation accuracy* dari model agar dapat mudah dianalisis untuk keperluan evaluasi seluruh kinerja model pada penelitian ini.

## 2. Analisa Python

Analisis Python merupakan proses *clean code* algoritma mulai dari memproses *dataset* sebagai pengujian dan pelatihan sampai *evaluasi* model sehingga kode efisien, efektif, dan mudah dibaca.