

BAB III

METODE PENELITIAN

3.1 Subjek dan Objek Penelitian

Subjek pada penelitian yang dilakukan penulis yaitu klasifikasi ikan laut menggunakan algoritma *Convolutional Neural Network* dan model VGG16. Objek pada penelitian yang dilakukan penulis adalah dataset yang merupakan data citra jenis ikan laut.

3.2 Alat dan Bahan

Dalam penelitian yang dilakukan, penulis memanfaatkan perangkat keras dan lunak untuk merancang dan membangun model algoritma. Berikut adalah daftar perangkat yang digunakan :

3.2.1 Perangkat Keras

Perangkat keras pada penelitian ini memiliki spesifikasi sebagai berikut:

- HP Notebook 14-R201TX
- Intel® Core™ i5-3317U CPU @ 1.70GHz
- NVIDIA GeForce 820M 1GB DDR3
- RAM 4GB DDR3

3.2.2 Perangkat Lunak

Perangkat lunak pada penelitian ini meliputi:

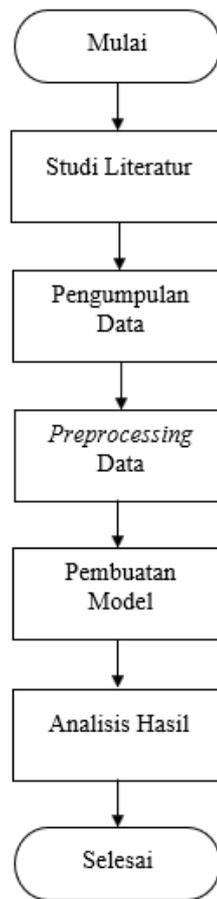
- Google Collab

3.2.3 Bahan

Bahan yang digunakan pada penelitian ini terdiri dari sekumpulan gambar berbagai jenis ikan laut, yang diperoleh dari platform Kaggle dan Google.

3.3 Diagram Alur Penelitian

Dalam penelitian ini, terdapat serangkaian tahapan yang telah penulis lalui. Berikut adalah urutan proses penelitian yang dijalankan:



Gambar 3. 1 Tahapan Penelitian

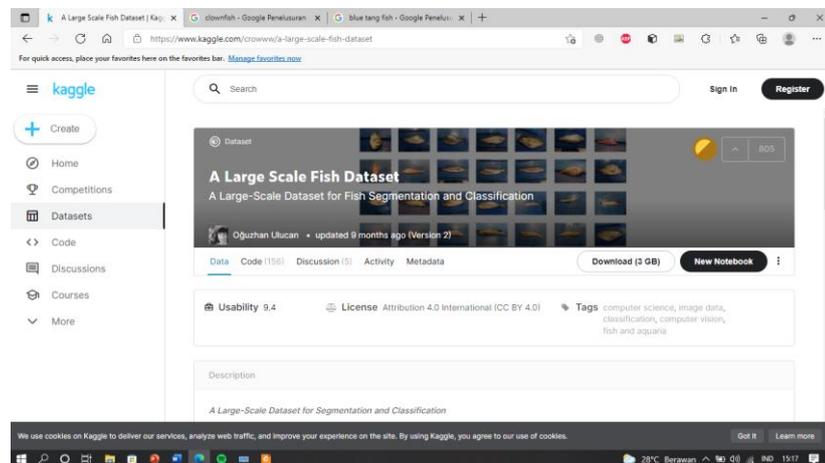
3.3.1 Studi Literatur

Pada penelitian ini, dilakukan analisis literatur. Penulis melakukan analisis literatur dengan mengumpulkan dan menelusuri referensi dari berbagai sumber seperti jurnal, artikel, dan situs web. Dari tahapan studi literatur, penulis mendapatkan sebuah permasalahan untuk penelitian ini. Selain itu, dengan melakukan studi literatur penulis memperoleh pengetahuan mengenai penelitian terkait seperti teori-teori, metode yang digunakan sehingga dapat mempermudah dalam penulisan penelitian.

3.3.2 Pengumpulan Data

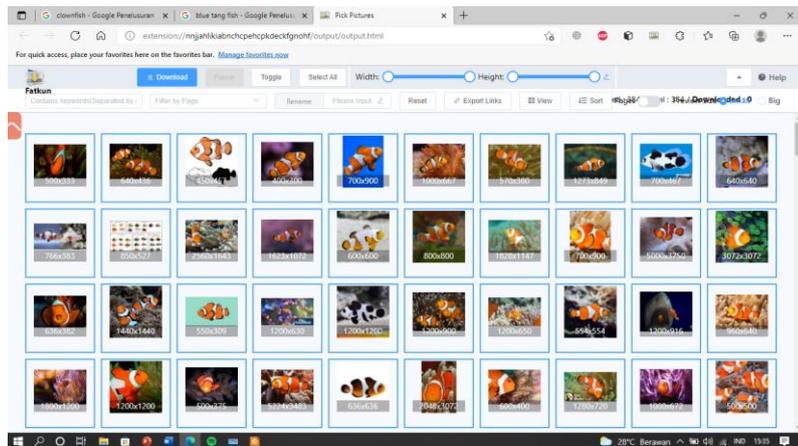
Dataset yang digunakan pada penelitian ini merupakan data gambar berbagai jenis ikan laut. Data diperoleh dari google dan *website* kaggle dengan format .jpg. Data gambar yang diperoleh dibagi menjadi 4 kelas dimana kelas tersebut merupakan jenis ikan laut. Masing - masing kelas memiliki 1500 data gambar dengan gambar asli berjumlah 30 gambar dan 50 gambar adalah hasil *augmentasi* dari gambar asli.

Sumber dataset gambar ikan laut diperoleh dari *website* kaggle. Kaggle merupakan *website* komunitas *online* yang membahas dan mempelajari *data science*, *machine learning* dan ilmu-ilmu lain yang terkait. Pada *website* tersebut juga menyediakan dataset yang diberikan oleh peneliti-peneliti pada *website* tersebut secara gratis. Sumber dataset pada penelitian ini diperoleh dari link : <https://www.kaggle.com/crowww/a-large-scale-fish-dataset>



Gambar 3. 2 Sumber dataset yang berasal dari *website* kaggle

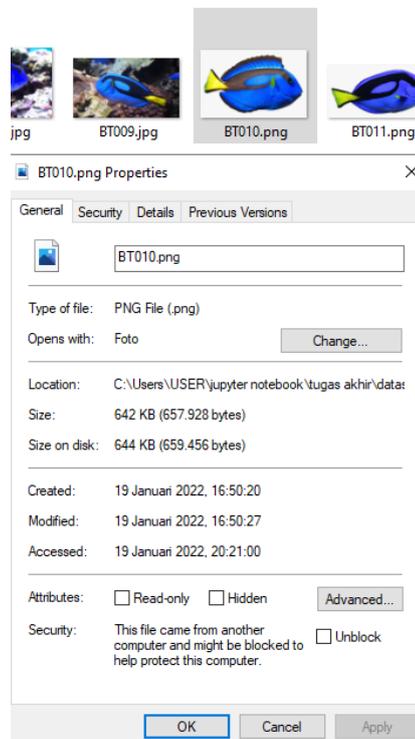
Pengumpulan data gambar pada google menggunakan ekstensi tambahan pada *web browser* yaitu Fatkun Batch Download Image. Dengan bantuan ekstensi tersebut, data gambar akan di *download* secara bersama dalam satu waktu sehingga mempersingkat waktu pengumpulan data.



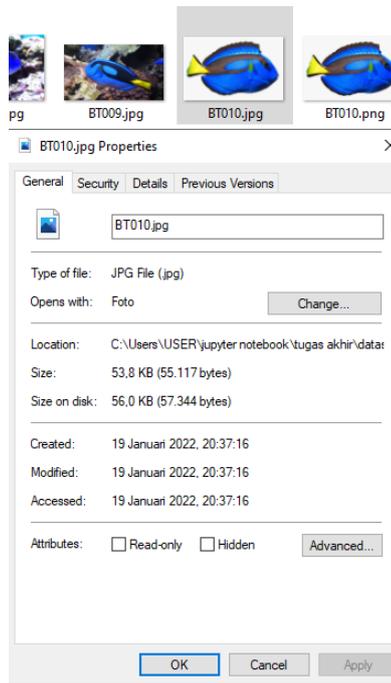
Gambar 3. 3 Proses pengumpulan data menggunakan Fatkun Batch Download Image

3.3.3 Preprocessing Data

Data yang sudah diperoleh selanjutnya akan masuk ke tahapan *preprocessing*. Pada tahap ini, data yang berformat selain .jpg (.jpeg dan .png) akan diubah ke format .jpg untuk memudahkan dalam proses *resize* dan augmentasi.



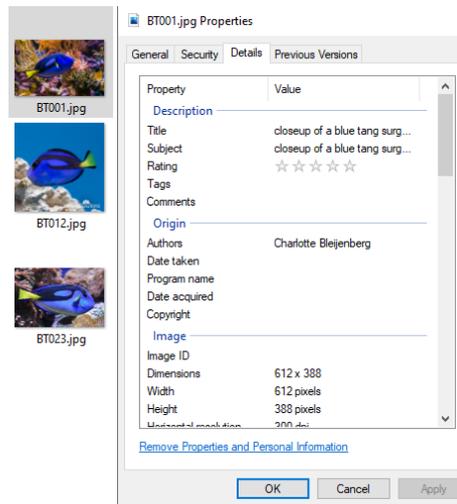
Gambar 3. 4 Sampel gambar sebelum di *convert*



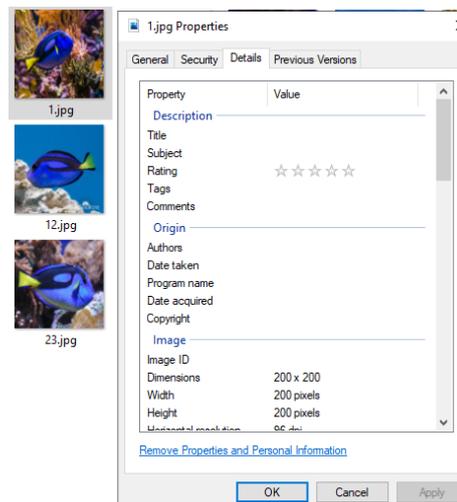
Gambar 3. 5 Sampel gambar setelah di *convert*

Gambar 3.4 menunjukkan sampel gambar yang sebelumnya memiliki format *.png*. Setelah dilakukan proses *convert* dengan menggunakan kode program, seperti yang ditunjukkan pada Gambar 3.5 format sampel gambar berubah menjadi *.jpg*. Proses ini juga diterapkan pada semua sampel gambar dengan format *.png* atau yang lainnya.

Setelah dataset gambar melalui proses konversi format, selanjutnya data gambar memasuki proses *resize* yang akan diterapkan ke semua dataset. Proses *resize* bertujuan agar saat proses pengolahan citra menjadi seragam dan mencegah *training loss*.



Gambar 3. 6 Sampel gambar sebelum di *resize*



Gambar 3. 7 Sampel gambar setelah di *resize*

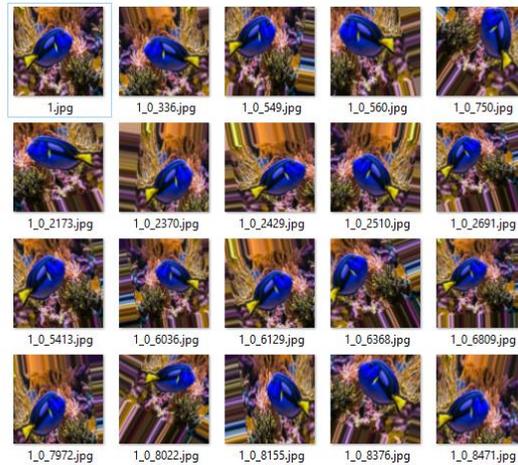
Gambar 3.6 menunjukkan sampel data gambar yang memiliki resolusi 612 x 388 piksel. Setelah sampel data melalui proses *resizing*, sampel data gambar berubah resolusinya menjadi 200 x 200 piksel seperti yang ditunjukkan pada Gambar 3.7. Proses ini juga akan diterapkan pada sampel data yang lain.

Tahapan proses *preprocessing* yang terakhir yaitu proses augmentasi. Proses augmentasi bertujuan untuk memperbanyak sampel data yang digunakan sebagai data latih dan uji agar model dapat mengenali suatu citra dengan berbagai pola dan bentuk. Pada

penelitian ini, proses augmentasi dilakukan pada dataset dengan masing-masing kelas sebanyak 50 kali dari 30 sampel data asli dengan variasi *rotate* dan *zoom* sehingga hasil dari masing-masing kelas memiliki 1500 data citra.



Gambar 3. 8 Sampel data sebelum di augmentasi



Gambar 3. 9 Sampel gambar setelah di augmentasi

Gambar 3.8 menunjukkan sampel gambar yang belum di augmentasi. Setelah masuk ke proses augmentasi dengan variasi rotation sebesar 40 derajat dan zoom sebesar 20%, sampel gambar dimodifikasi dengan jumlah 50 termasuk gambar asli seperti yang ditunjukkan pada Gambar 3.9.

3.3.4 Pembuatan Model

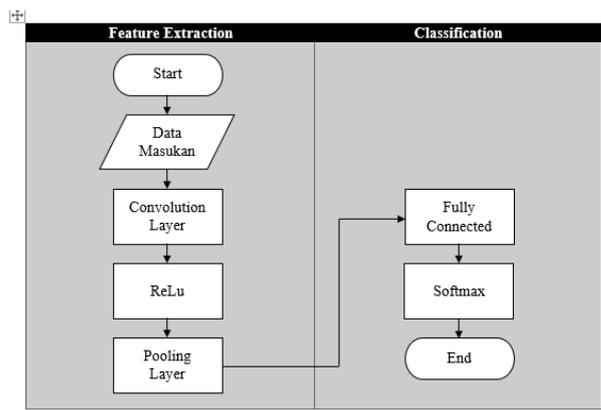
3.3.4.1 Convolutional Neural Network

Pada model dengan algoritma CNN, data yang sudah melalui proses *preprocessing* akan diolah untuk

menghasilkan klasifikasi jenis ikan laut. Data latih yang telah diberi label akan diproses dengan perulangan yang diperlukan untuk menghasilkan hasil yang akurat. Data latih akan digunakan ke dalam model yang nantinya akan diuji apakah model tersebut mampu memprediksi hasil secara baik atau tidak.

Umumnya CNN memiliki 2 tahapan dalam mengolah data citra. Tahapan pertama yaitu tahap *feature extraction* dimana pada tahap ini akan dilakukan ekstraksi ciri atau fitur penting dari data citra yang nantinya akan dipelajari oleh model sehingga model tersebut dapat mengklasifikasikan data citra masukan. Tahap *feature extraction* melibatkan *convolution layer*, fungsi aktivasi ReLu, dan *pooling layer*.

Tahapan yang kedua yaitu tahap *classification* dimana pada tahap ini hasil dari tahap *feature extraction* akan dilakukan prediksi berdasarkan label yang telah ditentukan. Pada tahapan ini melibatkan *fully connected layer* dan fungsi aktivasi *softmax*. Gambar 3.10 menunjukkan diagram alir untuk model CNN.



Gambar 3. 10 Diagram alir model CNN

- *Convolution Layer*

Berikut merupakan bagaimana cara kerja *convolution layer*. Diketahui data citra memiliki matriks berukuran 4x4 dan kernel dengan matriks 2x2.

6	3	5	1
4	7	2	8
1	2	1	5
3	2	4	1

1	0
0	1

Langkah pertama yaitu dengan menempatkan kernel ke bagian kiri atas data citra lalu mengalikannya seperti dibawah ini.

6	3	5	1
4	7	2	8
1	2	1	5
3	2	4	1

1	0
0	1

13		

Citra baru = citra *input* x kernel

$$\text{Citra baru} = (6 \times 1) + (3 \times 0) + (4 \times 0) + (7 \times 1)$$

$$\text{Citra baru} = 13$$

Langkah selanjutnya geser kernel sejauh 1 *stride* ke kanan dan kalikan dengan piksel selanjutnya pada data citra *input*.

6	3	5	1
4	7	2	8
1	2	1	5
3	2	4	1

1	0
0	1

13	5	

Citra baru = citra *input* x kernel

$$\text{Citra baru} = (3 \times 1) + (5 \times 0) + (7 \times 0) + (2 \times 1)$$

$$\text{Citra baru} = 5$$

Lakukan perhitungan hingga pergeseran kernel mencapai bagian kanan bawah dari citra *input*.

6	3	5	1
4	7	2	8
1	2	1	5
3	2	4	1

1	0
0	1

13	5	13
6	7	7
2	6	1

Citra baru = citra *input* x kernel

$$\text{Citra baru} = (1 \times 1) + (5 \times 0) + (4 \times 0) + (1 \times 1)$$

$$\text{Citra baru} = 1$$

- Fungsi Aktivasi ReLU

Hasil dari data citra yang telah diproses pada *convolution layer* akan melalui proses normalisasi menggunakan fungsi aktivasi ReLU dengan mengubah nilai negatif pada hasil *output convolution layer* menjadi nilai 0. Berikut merupakan hasil dari *convolution layer*.

13	5	13
6	7	7
2	6	1

Dari hasil *output convolution layer*, diketahui bahwa seluruh nilai dalam matriks 3x3 bernilai positif. Oleh karena itu hasil perhitungan ReLU akan menghasilkan hasil matriks yang sama dengan hasil *output convolution layer*.

- *Pooling Layer*

Hasil dari perhitungan fungsi aktivasi ReLU selanjutnya akan diproses dalam *pooling layer*. *Pooling layer* yang digunakan pada penelitian ini yaitu *max*

pooling layer. Berikut merupakan bagaimana cara kerja *max pooling layer*. Diketahui *filter* dengan matriks 2x2 sebagai berikut:

13	5	13
6	7	7
2	6	1

13	

Langkah pertama yaitu dengan menempatkan *filter* pada kiri atas citra, lalu mengambil nilai tertinggi dari matriks tersebut.

13	5	13
6	7	7
2	6	1

13	13

Lakukan pergeseran *filter* sebanyak 1 *stride* ke arah kanan, lalu ambil nilai tertinggi.

13	5	13
6	7	7
2	6	1

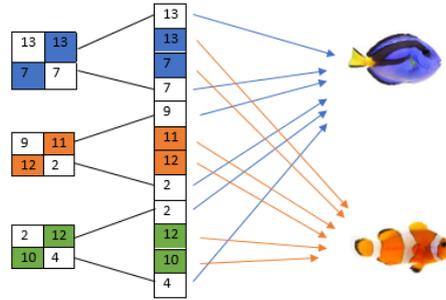
13	13
7	7

Lakukan pergeseran *filter* hingga mencapai bagian kanan bawah matriks data citra dan ambil nilai tertinggi.

- *Fully Connected Layer*

Hasil dari *pooling layer* merupakan *features* yang akan diproses pada *fully connected layer*. Pada *fully connected layer*, hasil dari *pooling layer* akan ditumpuk menjadi satu untuk menentukan suatu nilai

mendefinisikan suatu kelas. Berikut bagaimana cara kerja *fully connected layer* dapat dilihat pada gambar dibawah.

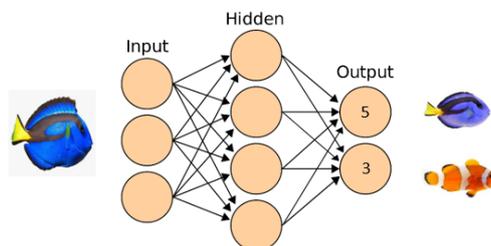


Gambar 3. 11 *Fully Connected Layer*

Dari Gambar 3.11 diketahui 3 buah *features* dengan matriks 2x2. Selanjutnya seluruh *features* akan ditumpuk menjadi satu. Dari *features* yang sudah ditumpuk terdapat beberapa piksel yang menjadi ciri khusus dari kelas.

- Fungsi Aktivasi Softmax

Fungsi aktivasi softmax digunakan untuk menghitung probabilitas setiap kelas yang dihasilkan oleh lapisan *fully connected*. Jumlah output dari fungsi aktivasi softmax harus sama dengan jumlah kelas yang digunakan dalam model jaringan saraf.



Gambar 3. 12 Fungsi Aktivasi Softmax

Diketahui *output* yang dihasilkan model *neural network* dengan nilai 5 untuk kelas *blue tang fish* dan 3

untuk kelas *clown fish*. Berikut merupakan perhitungan probabilitas fungsi aktivasi softmax.

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

Probabilitas kelas *blue tang fish* :

$$y_{ijk} = \frac{e^5}{e^5 + e^3}$$

$$y_{ijk} = \frac{148.41}{148.41 + 20.08}$$

$$y_{ijk} = 0.88$$

Probabilitas kelas *clown fish* :

$$y_{ijk} = \frac{e^3}{e^5 + e^3}$$

$$y_{ijk} = \frac{20.08}{148.41 + 20.08}$$

$$y_{ijk} = 0.12$$

- Model Arsitektur CNN

Model CNN yang dibangun untuk melakukan pelatihan dataset ikan menggunakan 4 layer. Pada layer konvolusi pertama data masukan yang berukuran 200 x 200 piksel akan dilakukan operasi konvolusi dengan 64 fitur dan dimensi gambar berubah menjadi 198 x 198 x 64. Selanjutnya citra akan masuk ke dalam *pooling layer* dan dilakukan pengurangan dimensi menjadi 99 x 99 x 64 piksel. Kemudian citra masuk kembali ke dalam layer konvolusi dengan 32 fitur dan dimensi gambar berubah menjadi 97 x 97 x 32 piksel. Selanjutnya citra mengalami reduksi data pada *pooling layer* menjadi 48 x 48 x 32

piksel. Kemudian citra masuk kembali ke dalam layer konvolusi dengan 16 fitur dan dimensi gambar berubah menjadi 46 x 46 x 16 piksel. Selanjutnya citra mengalami reduksi data pada *pooling layer* menjadi 23 x 23 x 16 piksel. Kemudian citra masuk kembali ke dalam layer konvolusi dengan 8 fitur dan dimensi gambar berubah menjadi 21 x 21 x 8 piksel. Selanjutnya citra mengalami reduksi data pada *pooling layer* menjadi 10 x 10 x 8 piksel. Dari proses konvolusi dan *max pooling* akan membentuk 1 dimensi *array* yang digunakan untuk menghubungkan ke *fully connected layer*. Dari *fully connected layer* akan menghasilkan *neuron* yang saling terhubung dengan objek yang telah diberi label. Arsitektur model CNN dapat dilihat pada Tabel 3.1.

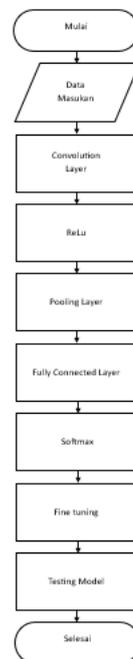
Layer(type)	Output Shape	Param #
conv2d (Conv2D)	(None, 198, 198, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 99, 99, 64)	0
conv2d_1 (Conv2D)	(None, 97, 97, 32)	18464
max_pooling2d_1 (MaxPooling2D)	(None, 48, 48, 32)	0
conv2d_2 (Conv2D)	(None, 46, 46, 16)	4624
max_pooling2d_2 (MaxPooling2D)	(None, 23, 23, 16)	0
conv2d_3 (Conv2D)	(None, 21, 21, 16)	1160

max_pooling2d_3 (MaxPooling2D)	(None, 10, 10, 16)	0
flatten (Flatten)	(None, 800)	0
dropout (Dropout)	(None, 800)	0
dense (Dense)	(None, 8)	6408
Dense_1 (Dense)	(None, 4)	36
Total params : 32.484		
Trainable params : 32.484		
Non-trainable params : 0		

Tabel 3. 1 Arsitektur Model CNN

3.3.4.2 VGG16

Tahapan pembuatan model VGG16 pada penelitian ini ditunjukkan pada Gambar 3.13 dibawah ini.



Gambar 3. 13 Diagram alir model VGG16

- Tahapan Model VGG16

Tahapan awal pembuatan model VGG16 yaitu diawali dengan menginput dataset yang sebelumnya sudah memasuki tahapan *preprocessing* ke dalam model VGG16. Setelah dataset di input pada model VGG16, dilakukan proses *fine tuning* pada model. *Fine tuning* dilakukan untuk membekukan beberapa lapisan *layer convolutional* dan *pooling layer* yang bertanggung jawab dalam proses ekstraksi fitur pada gambar yang akan diklasifikasikan. Dengan membekukan beberapa lapisan *layer*, fitur-fitur yang sudah dipelajari model VGG16 akan tetap bertahan. Langkah selanjutnya yaitu menambahkan *layer* klasifikasi baru dengan tujuan memperoleh *output* berdasarkan kelas yang sesuai dengan fitur-fitur dari hasil *pre trained model* yang sebelumnya dibekukan.

Proses selanjutnya yaitu melatih atau melakukan *fine tuning* pada model menggunakan dataset baru. Hal tersebut dilakukan dengan tujuan menyesuaikan fitur-fitur yang sudah ada pada model awal dengan fitur-fitur baru yang diperoleh dari hasil *training* model.