

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Rancangan Alur Penelitian

Untuk memfasilitasi penelitian, disusun beberapa tahapan untuk menerapkan pembelajaran mendalam VGG16 dengan filter tajam pada klasifikasi kanker kulit. Gambar 3.1 di bawah menunjukkan alur tahapan metode yang digunakan.



*Gambar 3.1 Tahapan Penelitian*

#### 3.2 Dataset Collection

Dataset yang digunakan untuk studi ini merupakan dataset yang tersedia pada laman *Kaggle* dengan judul “*Melanoma Skin Cancer Dataset of 10000 Images*”[9]. Berdasarkan Gambar 4 dan gambar 5, terdapat dua kategori gambar, yaitu *benign* dan *malignant*. Dimana seluruh dataset berjumlah 10615 data dengan ukuran 224 x 224 piksel.

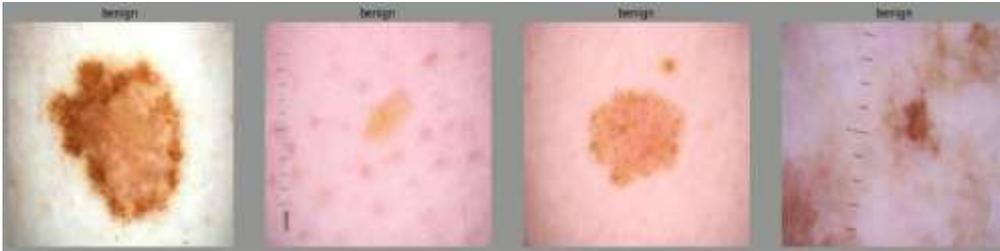
#### 3.3 Data Preparation

Setelah mendapatkan dataset yang akan digunakan, selanjutnya adalah melakukan persiapan pada dataset untuk digunakan pada pelatihan model.

##### **Pembagian Dataset**

Pada data train, kelas *benign* terdiri dari 5000 gambar sedangkan kelas *malignant* terdiri dari 4605 gambar. Jumlah gambar yang tidak seimbang dapat memengaruhi performa model machine learning. Berikut menampilkan gambar dari masing - masing kelas.

## Benign



Gambar 3.2 Sample hasil benign

## Malignant



Gambar 3.3 Sample hasil malignant

```
# Number of images for each disease
nums_train = {}
nums_val = {}

for s in skin:
    nums_train[s] = len(os.listdir(train_dir
+ '/' + s))

img_per_class_train =
pd.DataFrame(nums_train.values(),
index=nums_train.keys(), columns=["no.
of images"])

print("Train data distribution :")

img_per_class_train
```

nums\_train = {} dan nums\_val = {}: Membuat dua kamus kosong (nums\_train dan nums\_val) yang akan digunakan untuk menyimpan jumlah gambar per kelas di dataset pelatihan dan validasi.

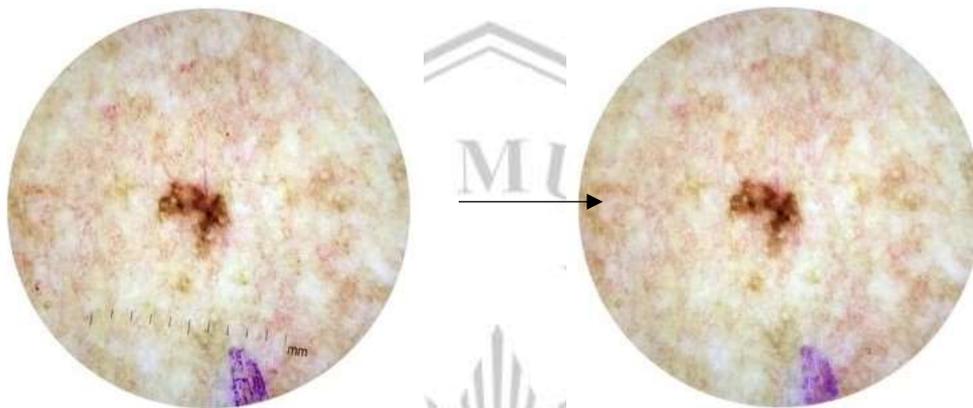
for s in skin:: Loop melalui setiap kelas dalam dataset. skin adalah iterabel yang berisi nama-nama kelas (misalnya, jenis kulit pada dataset).

img\_per\_class\_train = pd.DataFrame(columns=["no.of images"], index=nums\_train.keys(), values()): Membuat DataFrame pandas untuk menampilkan distribusi jumlah gambar per kelas dalam dataset pelatihan. DataFrame ini memiliki satu kolom bernama "no. of images" yang berisi jumlah gambar dan indeksinya adalah nama kelas.

img\_per\_class\_train: Menampilkan DataFrame yang berisi distribusi jumlah gambar per kelas dalam dataset pelatihan

### 3.4 Image PreProcessing

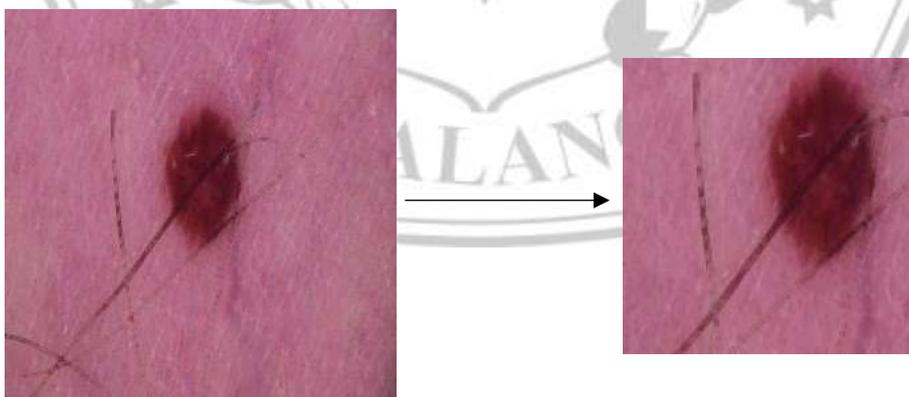
Pada penelitian ini diterapkan metode dull razor filtering. Dull Razor filtering berperan dalam mengurangi noise pada citra input yang disebabkan oleh rambut halus, dan sekaligus menormalisasi ukuran citra sebagai masukan.



*Gambar 3.4 Visualisasi dull razor filtering*

### 3.5 Segmentasi

Segmentasi pada gambar melakukan pemisahan atau pengelompokan piksel dalam suatu gambar menjadi bagian-bagian atau wilayah dengan karakteristik atau atribut tertentu. Tujuan utama segmentasi citra adalah untuk memudahkan analisis atau ekstraksi informasi yang lebih spesifik dari citra. Pada penelitian ini, gambar asli berukuran piksel 224 x 224 dan hasil segmentasi berukuran piksel 120 x 120.



*Gambar 3.5 Visualisasi resize segmentasi*

### 3.6 Arsitektur Model

Tiga model diusulkan oleh peneliti untuk dibandingkan. Penelitian ini menggunakan Convolutional Neural Network (CNN) yang memiliki arsitektur model 8 lapisan, CNN dengan model VGG 16, dan CNN dengan *dull razor filtering*. Secara umum, teknik CNN terdiri dari lapisan klasifikasi dan ekstraksi fitur. Lapisan ekstraksi fitur mengirimkan gambar ke lapisan convolutional, lapisan aktivasi Relu, dan lapisan pooling. Lapisan klasifikasi juga mencakup lapisan yang penuh terhubung dan lapisan aktivasi softmax. Gambar 3.6 menunjukkan pelatihan CNN menggunakan model arsitektur delapan lapisan.

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
dropout (Dropout)	(None, 222, 222, 32)	0
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 224)	64736
dropout_1 (Dropout)	(None, 109, 109, 224)	0
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 224)	0
flatten (Flatten)	(None, 653184)	0
dense (Dense)	(None, 128)	83607680
dense_1 (Dense)	(None, 2)	258

```
=====  
Total params: 83673570 (319.19 MB)  
Trainable params: 83673570 (319.19 MB)  
Non-trainable params: 0 (0.00 Byte)
```

Gambar 3.6 Summary Convolutional Layer pada arsitektur 8 Layer

Tahap pelatihan model *dull razor* terdiri dari dua lapisan convolutional, dua Max\_pooling2d, flatten, dense, dan aktivasi softmax. Pelatihan CNN dengan model VGG 16 ditunjukkan pada Gambar 3.7, yang menunjukkan input gambar berukuran 224 x 224 piksel dengan tiga channel RGB.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_2 (Dense)	(None, 1)	25089

Total params: 14739777 (56.23 MB)
Trainable params: 14739777 (56.23 MB)
Non-trainable params: 0 (0.00 Byte)

Gambar 3.7 Summary Convolutional Layer pada Model VGG16

Dalam model VGG-16, tahap pelatihan model terdiri dari tiga belas lapisan convolutional, empat Max\_Pooling2D, flatten, dense, dan aktivasi softmax. Input gambar berukuran 224 x 224 piksel melalui tiga channel RGB. Gambar 3.8 menunjukkan pelatihan CNN dengan filter *dull razor filtering*.

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 222, 222, 32)	896
dropout_2 (Dropout)	(None, 222, 222, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_3 (Conv2D)	(None, 109, 109, 224)	64736
dropout_3 (Dropout)	(None, 109, 109, 224)	0
max_pooling2d_3 (MaxPooling2D)	(None, 54, 54, 224)	0
flatten_1 (Flatten)	(None, 653184)	0
dense_2 (Dense)	(None, 128)	83607680
dense_3 (Dense)	(None, 2)	258

=====  
Total params: 83673570 (319.19 MB)  
Trainable params: 83673570 (319.19 MB)  
Non-trainable params: 0 (0.00 Byte)  
=====

Gambar 3.8 Summary Convolutional Layer pada dull razor

Tahap pelatihan model dull razor terdiri dari dua lapisan convolutional, dua Max\_pooling2d, flatten, dense, dan aktivasi softmax. Input gambar dengan dimensi 224 x 224 piksel melalui tiga channel RGB.

### 3.7 Augmentasi Data

Seluruh gambar di dalam dataset akan dilakukan proses augmentasi Sebagian secara acak dengan tujuan agar system tetap dapat mengenali berbagai bentuk gambar sebagai data citra gambar kanker kulit saat akan dilakukan klasifikasi. Proses augmentasi tersebut menggunakan fungsi Image Data Generator dengan beberapa parameter yang diaktifkan sesuai dengan table II yang akan merubah bentuk masing – masing gambar yang ada di dalam dataset.

Table II. Jenis Paramater yang diaktifkan pada proses augmentasi

Parameter	Server
Rescale	1.0/255
Rotation _range	20
Horizontal _flip	True
Validation_split	0.25

### 3.8 Skenario Pengujian

Circa yang dipelajari terbagi ke dalam dua folder, train dan uji, di mana dari seluruh dataset “*Melanoma Skin Cancer Dataset of 10000 Images*” yang sebelumnya berjumlah 10615 hanya di ambil dari data train dengan jumlah data 9605. Berikutnya, dataset diterbagi Kembali menjadi dua kategori, yakni benign dan malignant. Untuk proses pelatihan dan pengujian model, data set dibagi menjadi data validasi dan data latih.

Jumlah data terangkum pada tabel 3 dengan pembagian data train dan validation masing – masing sebesar 75% dan 25% dari keseluruhan data pada masing – masing kelas. Jumlah dataset sebelum dan sesudah augmentasi adalah sama, hal tersebut dikarenakan proses augmentasi yang tidak menambahkan variasi gambar.

Table III. Jumlah pembagian data train dan validation

Number of training data	7204
Number of validation samples	2401
Total Images	9605

Selanjutnya, masing-masing model diuji menggunakan MaxPooling2D sesuai dengan Tabel IV.

Table IV. Skenario Pengujian

Model	Jumlah <i>Hidden Layer</i>	Skenario <i>Pooling</i>
Model 1	2	MaxPooling2D
Model 2	4	MaxPooling2D
Model 3	2	MaxPooling2D

