Klasifikasi Batik Jawa Tengah dengan VGG-16, ResNet-12 dan MobileNet

Oleh:

Yunus Marsetio	C14190065
Peter Yudhistira	C14190067
Matthew Sutanto	C14190085
Francisco Allenxeon	C14190118
Vincent Darmawan	C14190162
Michael Halim	C14190119
Gregorius Tifanico	C14190135

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS KRISTEN PETRA

SURABAYA

2022

1. Deskripsi Proyek

Klasifikasi dua tingkat terhadap sebuah sampel gambar batik. Klasifikasi pertama merupakan klasifikasi biner, untuk menentukan apakah sampel merupakan batik yang berasal dari Jawa Tengah atau bukan. Jika batik adalah batik Jawa Tengah, akan dilakukan klasifikasi *multiclass* untuk menentukan apakah batik tersebut tergolong Batik Kawung, Parang, Ceplok, atau Sidomukti.

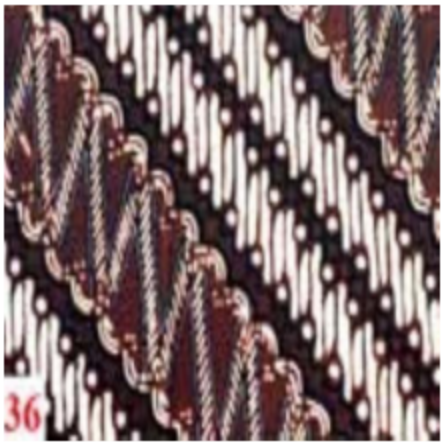
Klasifikasi multiclass dilakukan untuk membedakan batik-batik Jawa Tengah berdasarkan karakteristiknya.

2. Dataset

Dataset berjumlah 2002 gambar yang terdiri dari 4 *class* yaitu Kawung, Parang, Ceplok, dan Sidomukti. Dataset berukuran 250x250. Dataset akan di *split* dengan rasio 80 : 20 untuk *training* dan *testing*. Dataset akan diperkaya dengan augmentasi berupa flip.

- Batik Kawung.

- Parang.



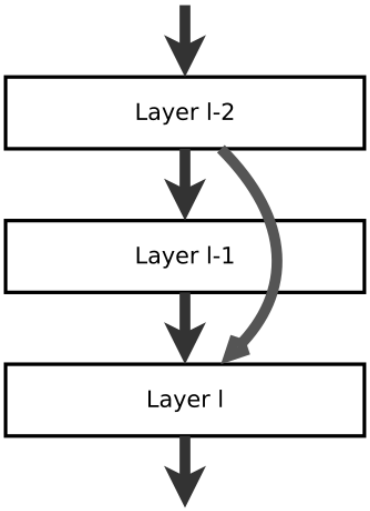
- Ceplok:

- Sidomukti:

3. Dasar Teori

a. ResNet

ResNet (Residual Neural Network) adalah salah satu jenis *artificial neural network* yang diperkenalkan oleh Kaiming He, Xiangyu Zhang, Shaoqing Ren dan Jian Sun dalam “Deep Residual Learning for Image Recognition” pada tahun 2015.



ResNet berusaha untuk menyelesaikan permasalahan *vanishing gradient* dengan cara menambahkan *skip connections* atau *shortcuts* yang melompati satu layer atau lebih sehingga gradient bisa dipertahankan.

Kelebihan ResNet:

- ResNet dapat memudahkan proses training dari jumlah layer yang banyak tanpa menambah persentase training error.
- ResNet membantu menyelesaikan *vanishing gradient problem*.

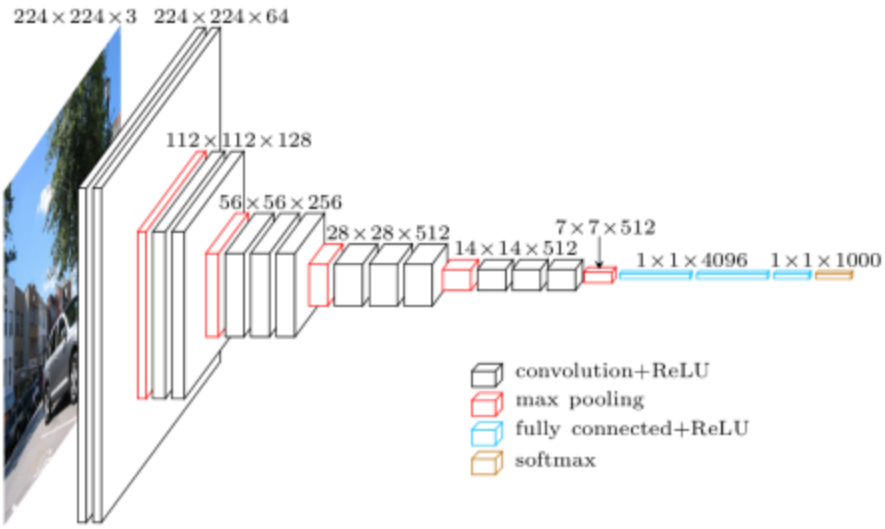
Kekurangan ResNet:

- Kompleksitas yang cukup tinggi karena adanya *skip connections*.

https://iq.opengenus.org/residual-neural-networks/
https://www.mygreatlearning.com/blog/resnet/
https://en.wikipedia.org/wiki/Residual_neural_network

b. VGG-16

VGG-16 adalah sebuah jenis *Convolutional Neural Network (CNN)* yang menggunakan filter 3x3 dengan stride 1 pixel. Filter ini lebih kecil dibandingkan dengan AlexNet yang menggunakan 11x11 dengan stride 4 pixel dan ZFNet yang menggunakan 7x7 dengan stride 2 pixel. Model ini menggunakan kedalaman 16 - 19 layer.



https://www.mygreatlearning.com/blog/introduction-to-vgg16/#VGG%20%E2%80%93%20The%20Idea

c. MobileNet (https://arxiv.org/pdf/1704.04861.pdf)

Pembaharuan yang diterapkan pada arsitektur MobileNet adalah penerapan depthwise separable convolution yaitu tensor diproses oleh depthwise convolution yang berisi 1 filter saja diikuti dengan 1x1 convolution dengan N filter untuk menggabungkannya.

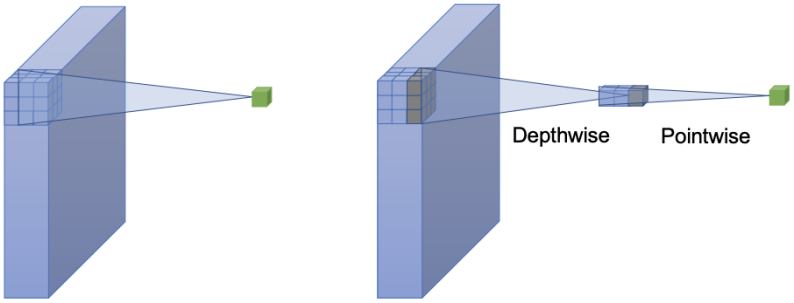
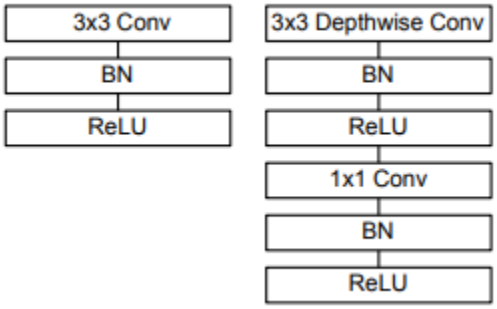


Figure 3: Standard convolution and depthwise separable convolution.

Hasil dari penggunaan Depthwise-separable convolution ini adalah berkurangnya biaya komputasi daripada konvolusi biasa.

Tiap Layer depthwise convolution diikuti oleh layer BatchNormalization dan layer ReLU yang diikuti lagi oleh konvolusi dengan filter berukuran 1x1 yang diikuti layer BatchNormalization dan layer ReLU. Berikut adalah ilustrasi dari layer konvolusi mobilenet.



Gambar 3. Perbandingan Layer Konvolusi biasa dengan layer Depthwise Separable Convolution

Berikut arsitektur dari Mobile Net

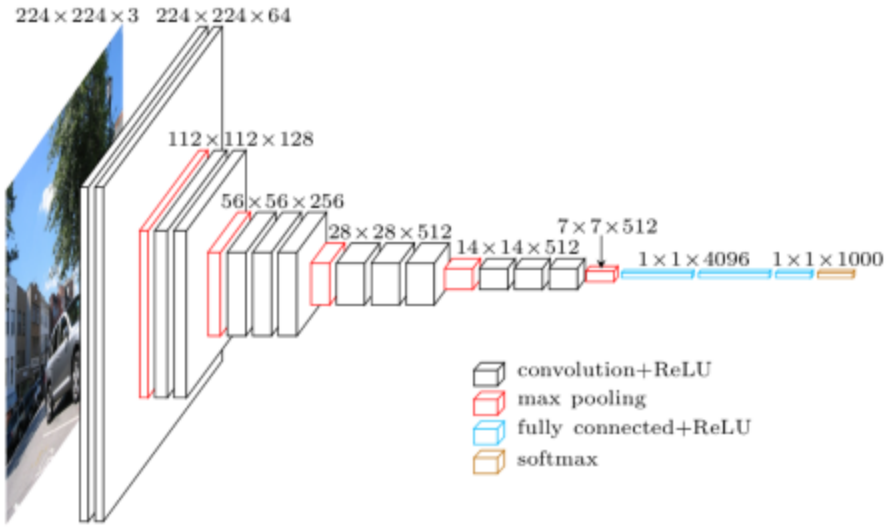
Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Gambar 3. Strukur MobileNet

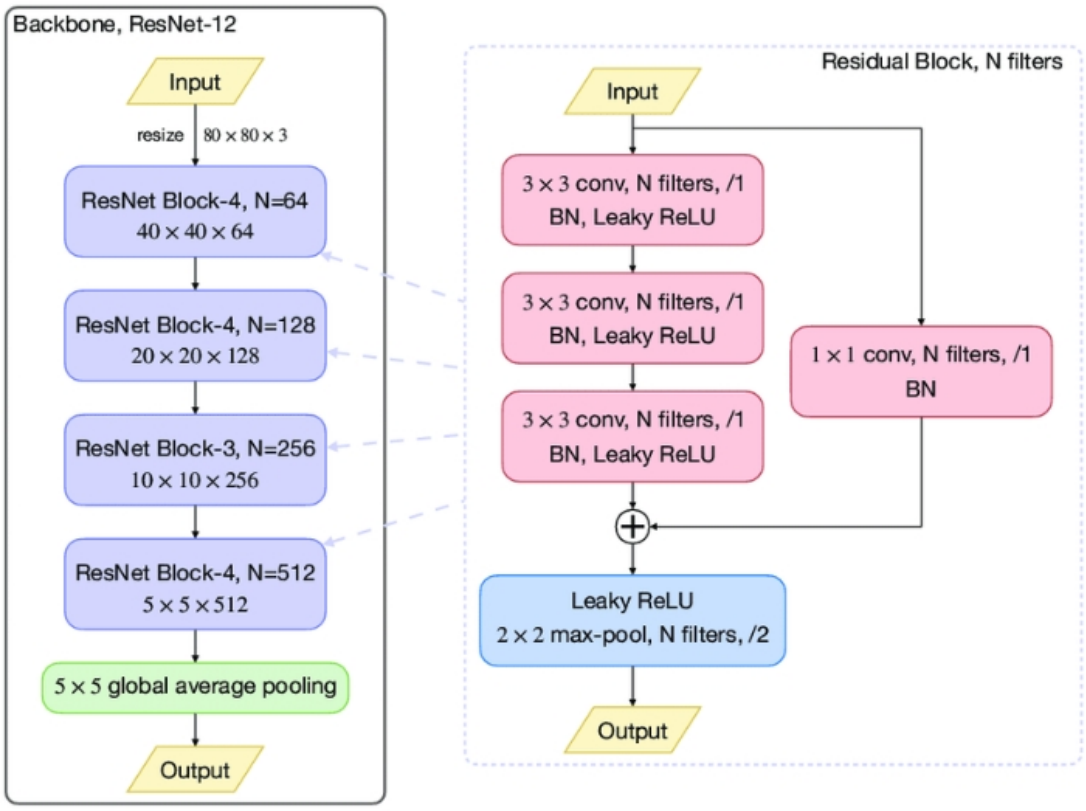
https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

Model yang digunakan merupakan VGG16, ResNet 12, dan MobileNet. Diantaranya, model VGG16 akan digunakan untuk melakukan *multiclass classification*. Klasifikasi ini menentukan apakah batik merupakan Batik Kawung, Parang, Ceplok, atau Sidomukti.



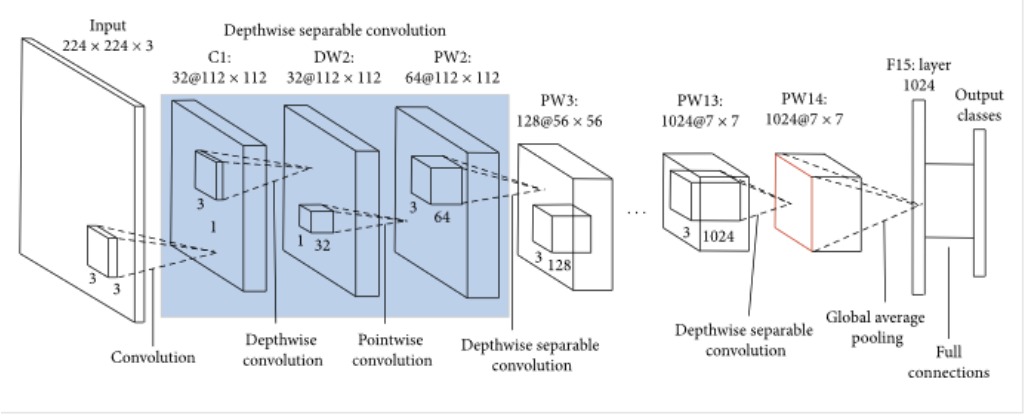
Gambar 1. Arsitektur VGG16

Model ResNet-12 digunakan untuk *binary classification*. Tujuannya agar dapat memprediksi batik yang dimasukkan merupakan batik jawa tengah atau bukan.



Gambar 2. Arsitektur Resnet-12

Model MobileNet digunakan untuk *binary classification*. Tujuannya agar dapat memprediksi batik yang dimasukkan merupakan batik jawa tengah atau bukan.



Gambar 3. Arsitektur MobileNet

4. Jadwal dan Pembagian kerja

	Minggu 1	Minggu 2	Minggu 3	Minggu 4
Pengumpulan Data				
Pembuatan Model				
Evaluasi				

Minggu 1:

1. Yunus : Preprocessing Data dan Membuat model ResNet-12
2. Peter : Membuat model ResNet-12 yang di modifikasi
3. Matthew : Preprocessing Data dan Membuat model VGG16
4. Allen : Membuat model VGG16 yang di modifikasi
5. Vincent : Preprocessing Data dan Membuat model MobileNet
6. Halim : Pengumpulan Data (Batik Kawung, Parang, Sidomukti)
7. Tifa : Pengumpulan Data (Batik Ceplok)

Minggu 2:

1. Yunus : Fit Model ResNet-12
2. Peter : Fit Model ResNet-12 yang di modifikasi

3. Matthew : Fit Model VGG16
4. Allen : Fit Model VGG16 yang di modifikasi
5. Vincent : Fit Model MobileNet

Minggu 3:

1. Yunus : Fit Model ResNet-12 dengan setting parameter
2. Peter : Fit Model ResNet-12 yang di modifikasi dengan setting parameter
3. Matthew : Fit Model VGG16 dengan setting parameter
4. Allen : Fit Model VGG16 yang di modifikasi dengan setting parameter
5. Vincent : Fit Model MobileNet dengan setting parameter

Minggu 4:

1. Yunus : Evaluasi dan Pembuatan laporan
2. Peter : Evaluasi dan Pembuatan laporan
3. Matthew : Evaluasi dan Pembuatan laporan
4. Allen : Evaluasi dan Pembuatan laporan
5. Vincent : Evaluasi dan Pembuatan laporan

5. Rencana Research

a. Akurasi berdasarkan data original
b. Akurasi berdasarkan data augmentasi
c. Akurasi berdasarkan model klasifikasi 4 kelas
d. Akurasi berdasarkan model klasifikasi 4 kelas + 1 kelas (yang lainnya)

6. Hasil Research

a. Binary Classification

VGG-16	Resnet-12	MobileNet
Accuracy Train: 0.829 Accuracy Test: 0.805 Epoch: 10	Accuracy Train: 0.9744 Accuracy Test: 0.9333 Epoch: 50	Accuracy Train: 0.939 Accuracy Test: 0.921 Epoch: 10

b. Multi class Classification (4 Class)

VGG-16	Resnet-12	MobileNet

Accuracy Train: 0.7577 Accuracy Test: 0.7481 Epoch: 10	Accuracy Train: 0.9269 Accuracy Test: 0.9077 Epoch: 50	Accuracy Train: 0.9737 Accuracy Test: 0.8753 Epoch: 10

c. Multi class Classification (4 Class Augmentasi)

VGG-16	Resnet-12	MobileNet
Accuracy Train: 0.7919 Accuracy Test: 0.7799 Epoch: 10	Accuracy Train: 0.9938 Accuracy Test: 0.8602 Epoch: 31 (dari 50)	Accuracy Train: 0.986 Accuracy Test: 0.957 Epoch: 20

d. Multi class Classification (5 Class)

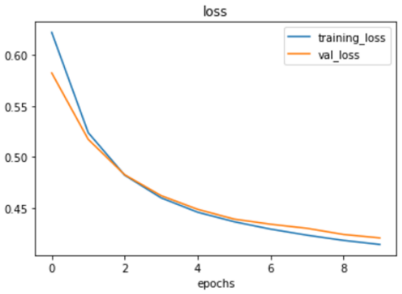
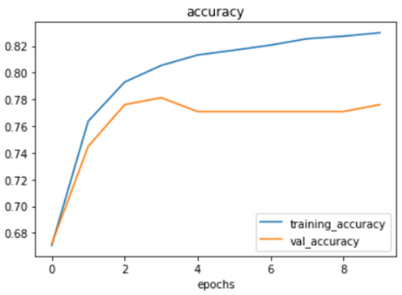
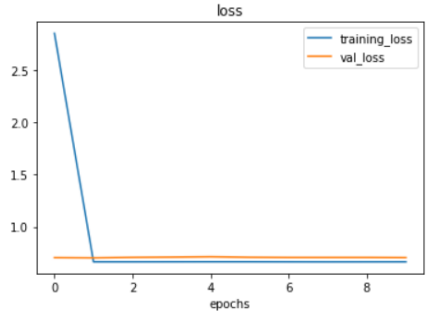
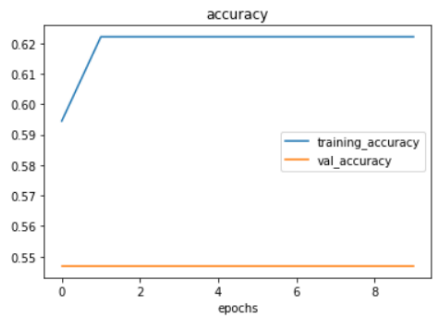
VGG-16	Resnet-12	MobileNet
Accuracy Train: 0.7913 Accuracy Test: 0.7724 Epoch: 10	Accuracy Train: 0.9633 Accuracy Test: 0.8776 Epoch: 30	Accuracy Train: 0.964 Accuracy Test: 0.872 Epoch: 20

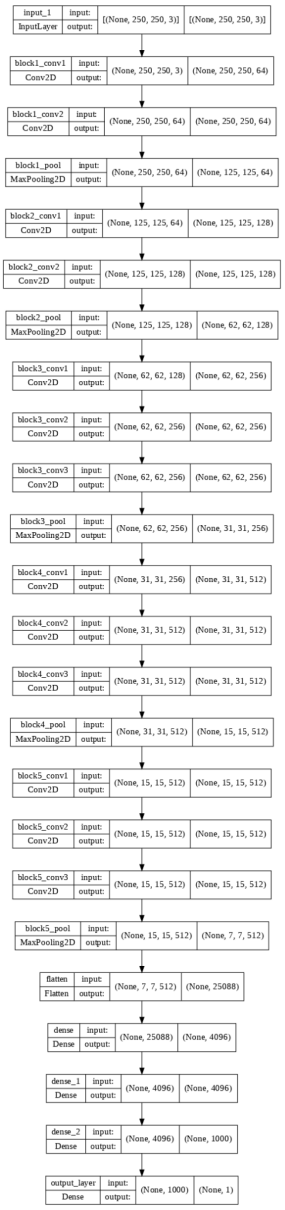
7. Progress Perubahan Parameter

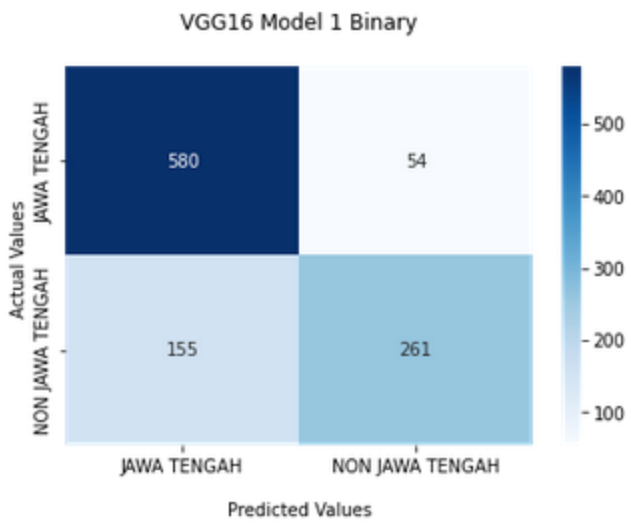
a. VGG-16

Layer input diubah dari (224, 244, 3) menjadi (250, 250, 3). 3 layer dense dihapus karena menyebabkan overfitting. Layer flatten diubah menjadi GlobalAverage untuk mengubah data menjadi 1 dimensi.

i. Binary Classification

Model 1: -Freeze layer Convolutional -Remove Dense 4096x4096x1000 -Output activity function softmax -epoch 10	  Accuracy Training : 0.8010
Model 2: Default	  Accuracy Training : 0.6221

	
Model 3: -Remove Dense 4096x4096x1000 -Add globalAverage layer -Output activity function softmax -epoch 10	Accuracy Training : 0.6221
Model 3: -Remove Dense 4096x4096x1000 -Add globalAverage layer -Output activity function softmax -epoch 50	Accuracy Training : 0.6221



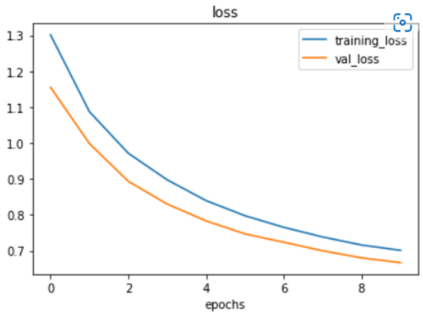
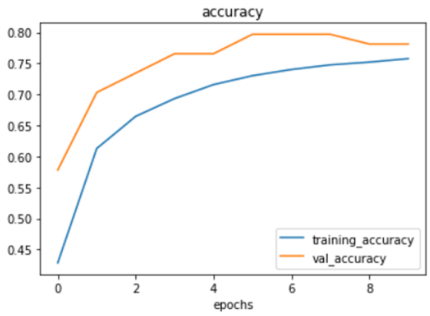
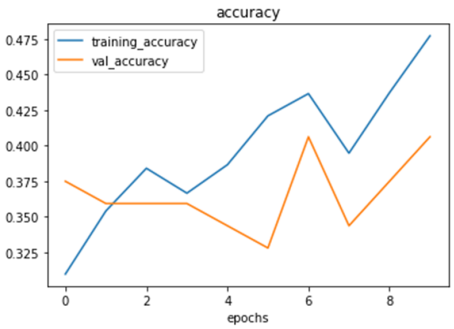
Skenario terbaik :

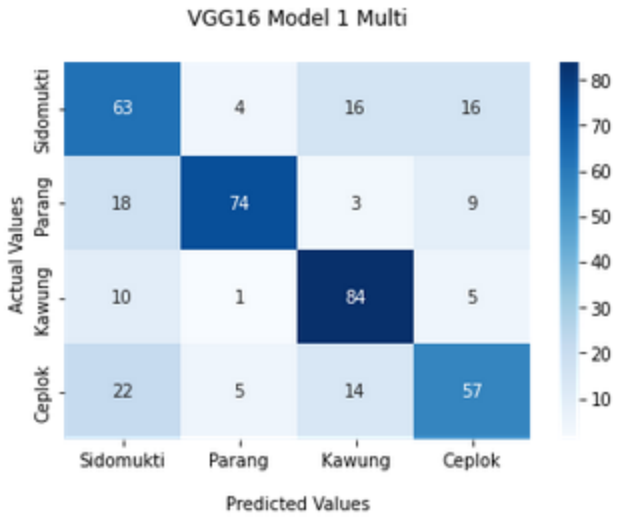
Label: JAWA TENGAH; Predicted: JAWA TENGAH

Skenario terburuk :

Label: JAWA TENGAH; Predicted: NON JAWA TENGAH

ii. Multi class Classification (4 Class)

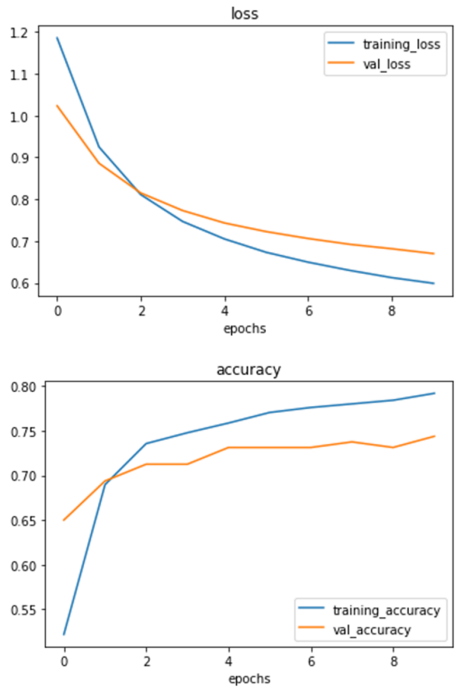
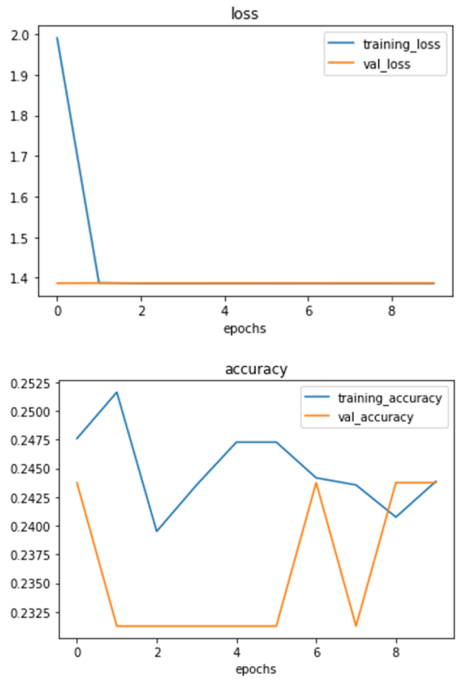
Model 1: -Freeze layer Convolutional -Remove Dense 4096x4096x1000 -Output activity function softmax -epoch 10	  Accuracy Training : 0.7577
Model 2: -Remove Dense 4096x4096x1000 -Output activity function softmax -epoch 10	  Accuracy Training : 0.4772

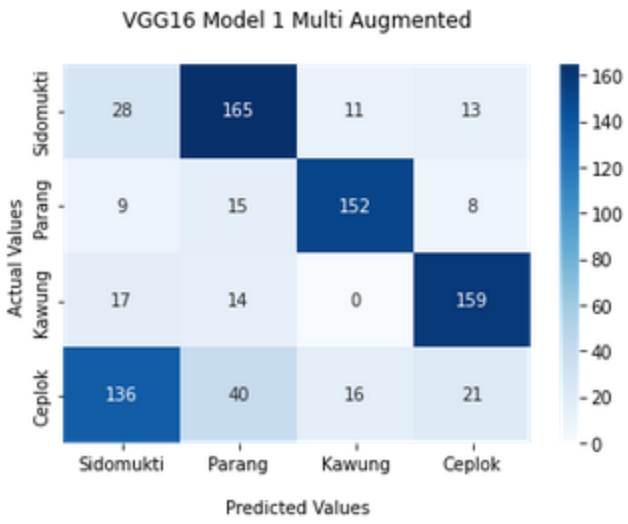


Skenario terbaik :

Skenario terburuk :

iii. Multi class Classification (4 Class Augmentasi)

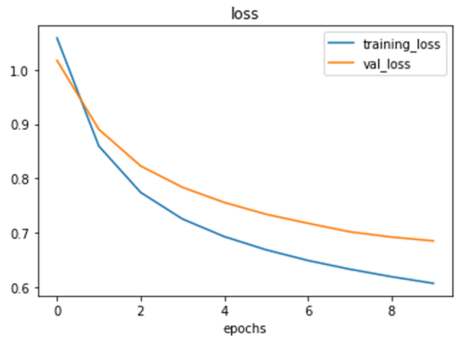
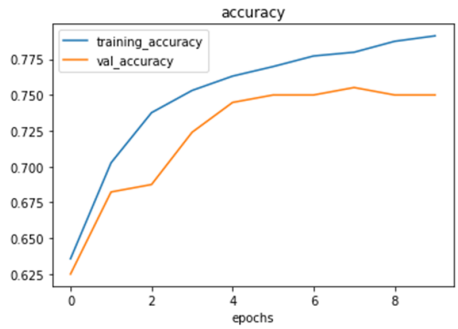
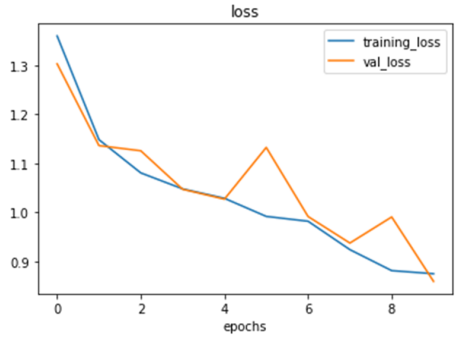
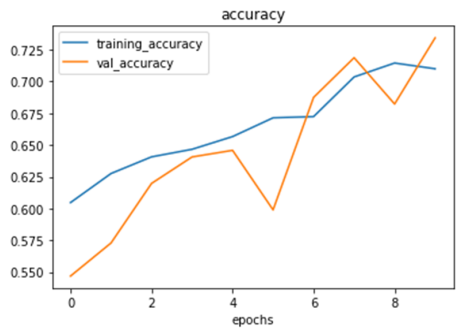
Model 1: -Freeze layer Convolutional -Remove Dense 4096x4096x1000 -Output activity function softmax -epoch 10	 Accuracy Training : 0.7919
Model 2: -Remove Dense 4096x4096x1000 -Output activity function softmax -epoch 10	 Accuracy Training : 0.2439

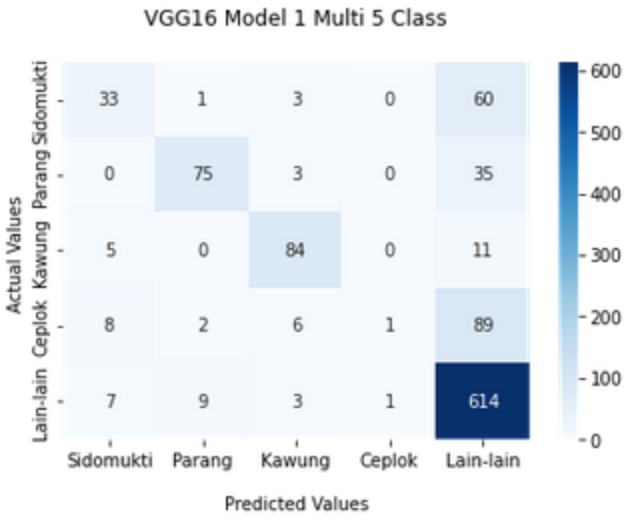


Skenario terbaik :

Skenario terburuk :

iv. Multi class Classification (5 Class)

Model 1: -Freeze layer Convolutional -Remove Dense 4096x4096x1000 -Output activity function softmax -epoch 10	  Accuracy Training : 0.7913
Model 2: -Remove Dense 4096x4096x1000 -Output activity function softmax -epoch 10	  Accuracy Training : 0.7100



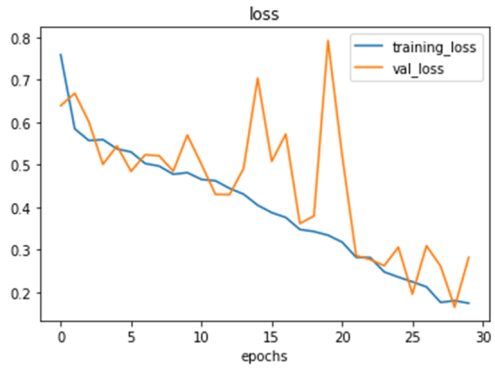
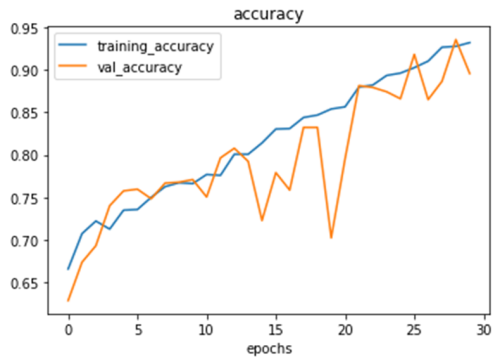
Skenario terbaik :

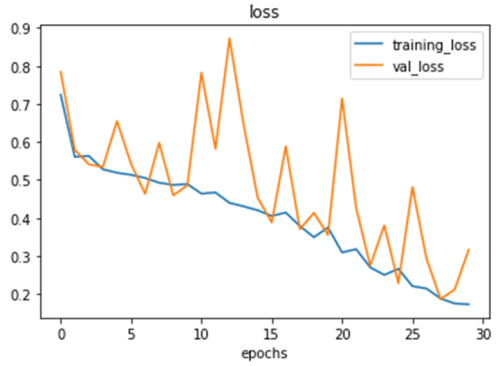
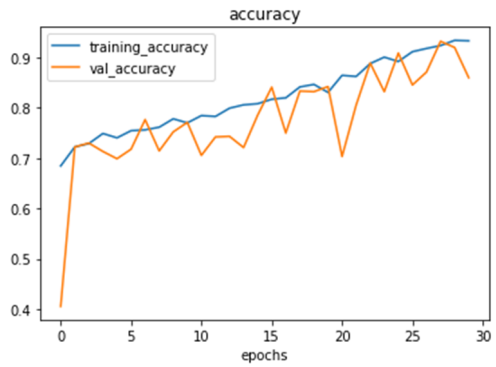
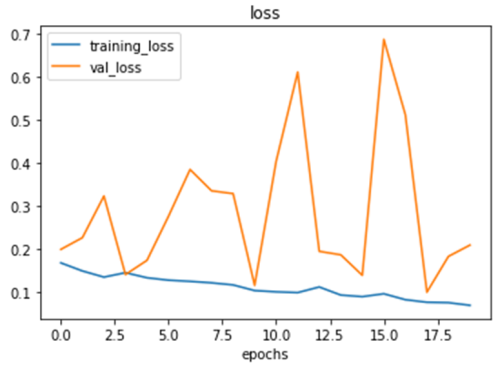
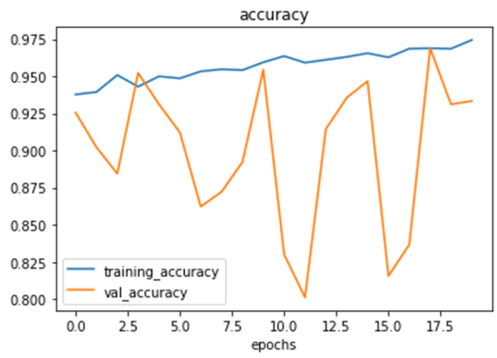
Skenario terburuk :

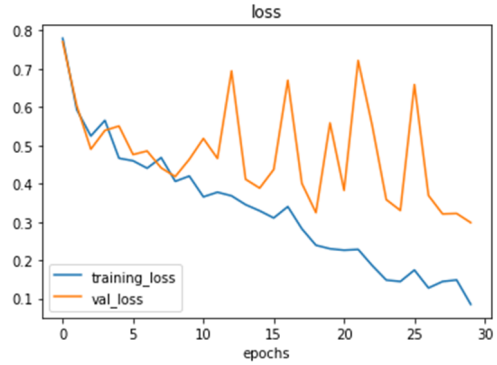
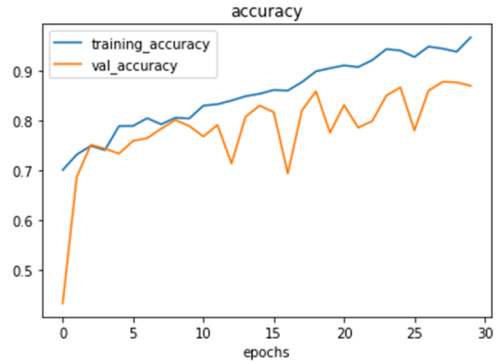
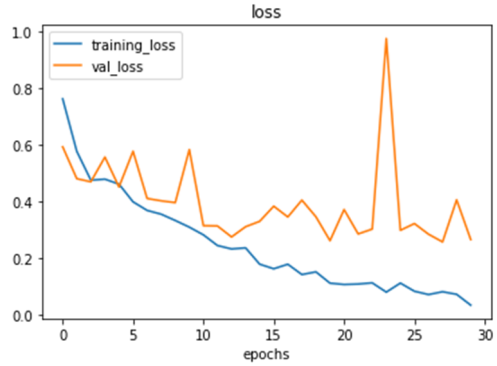
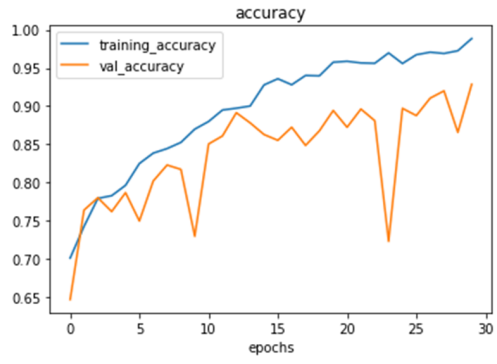
b. Resnet-12

Perubahan Parameter yang dilakukan adalah dengan merubah *Global Average Pooling* menjadi *Global Max Pooling* dengan alasan agar pembelajaran lebih cepat dan karena batik merupakan gambar bermotif yang memerlukan perhitungan ketajaman motif pada gambar. Lalu dari beberapa percobaan yang telah dilakukan, terlihat bahwa *Global Max Pooling* mampu mencapai akurasi *train* yang tinggi dengan lebih cepat dibandingkan dengan *Average Max Pooling*. Namun hasil akurasi *test* pada *Global Max Pooling* cukup rendah bila dibandingkan dengan *Average Max Pooling*. Hal ini bisa dikarenakan oleh model yang *overfitting*, sehingga kami mencoba untuk melakukan *training* dengan tambahan *Dropout*. Setelah dilakukan penambahan *dropout*, hasilnya beberapa model mampu memberikan akurasi *test* yang tinggi dengan *dropout* tersebut.

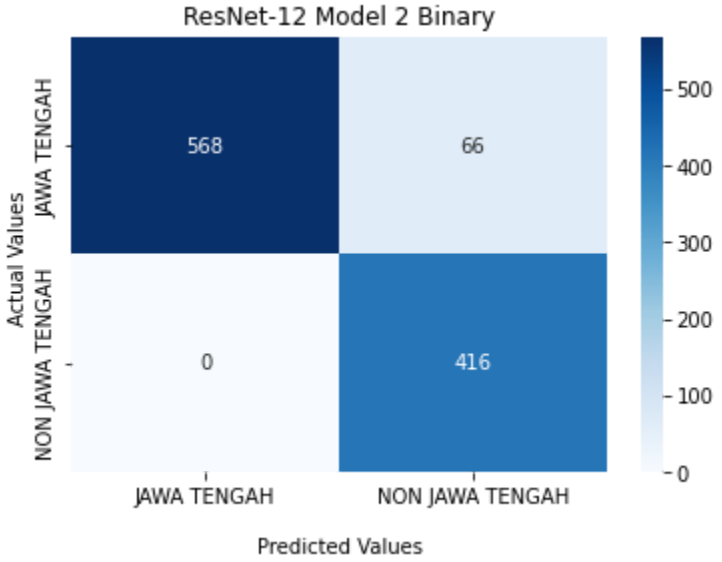
i. Binary Classification

Model 1: Default	  30 epoch, Accuracy Train = 0.9320, Accuracy Test = 0.8957

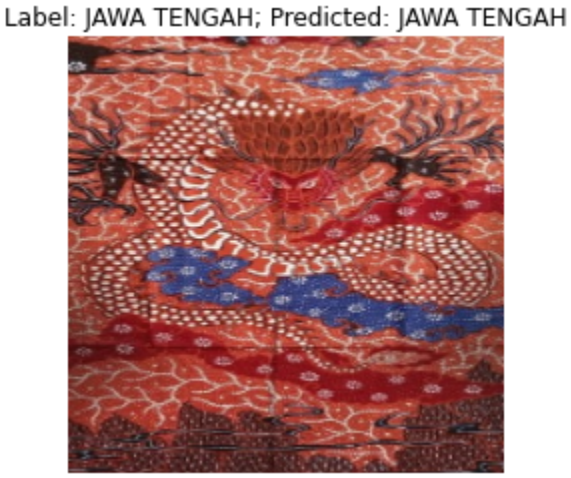
Model 2: Penambahan 1 Dropout (0.1)	    30 epoch, Accuracy Train = 0.9333, Accuracy Test = 0.8600 50 epoch, Accuracy Train = 0.9744, Accuracy Test = 0.9333

Model 3: Penambahan 1 Dropout (0.1), Merubah Global Avg Pool ke Global Max Pool	  30 epoch, Accuracy Train = 0.9672, Accuracy Test = 0.8700
Model 4: Merubah Global Avg Pool ke Global Max Pool	  30 epoch, Accuracy Train = 0.9886,

	Accuracy Test = 0.9286



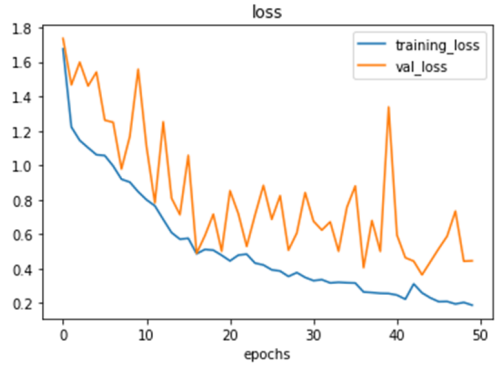
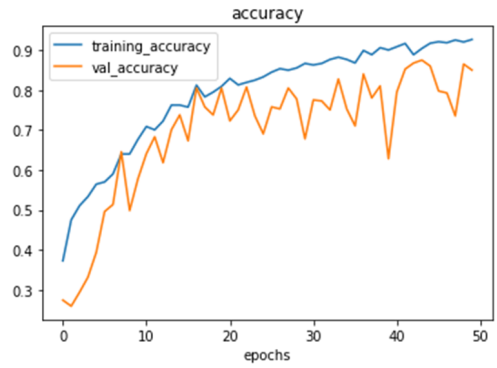
Skenario Terbaik:

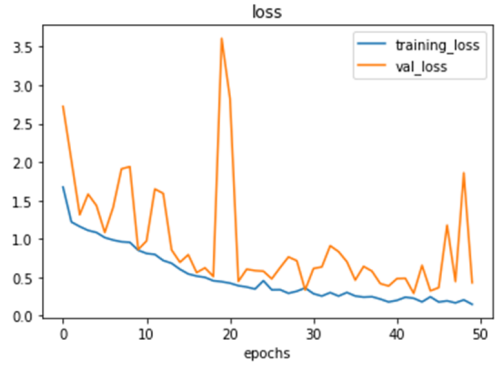
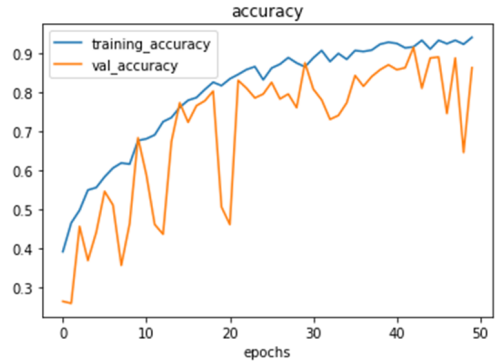
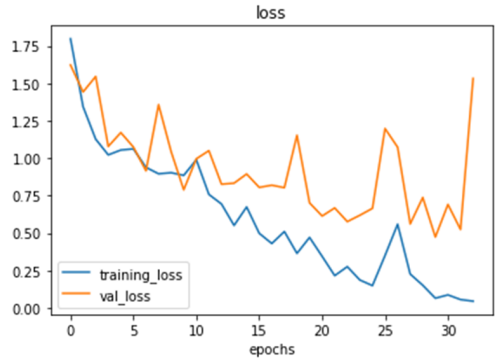


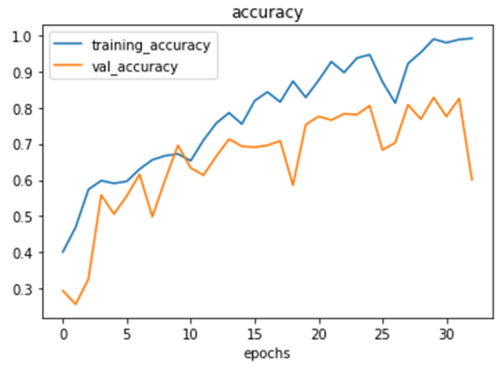
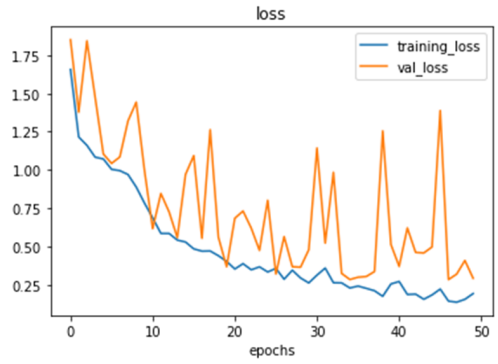
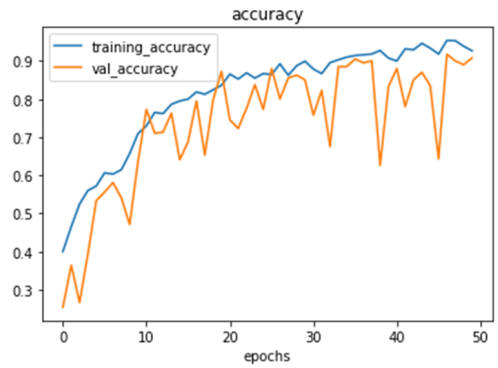
Skenario Terburuk:

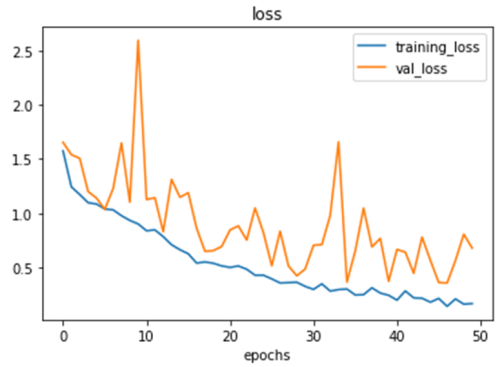
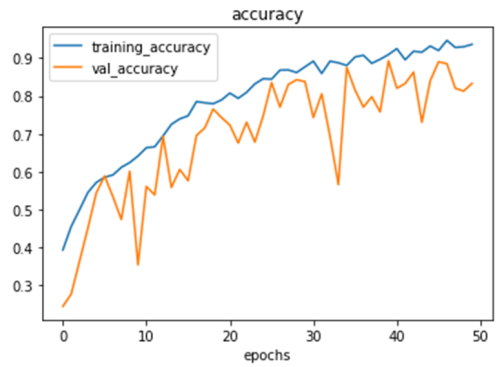
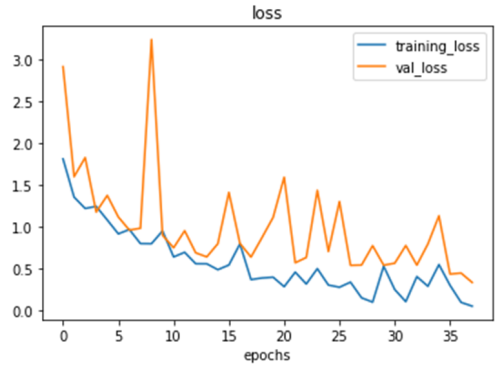
Label: JAWA TENGAH; Predicted: NON JAWA TENGAH

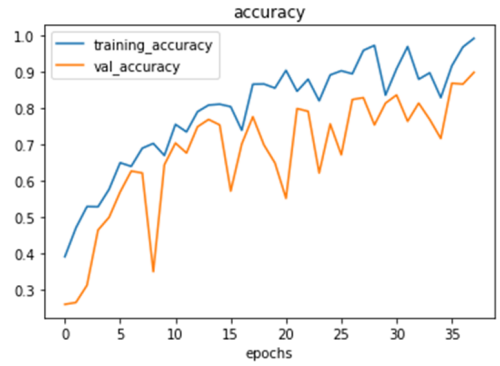
ii. Multi class Classification (4 Class)

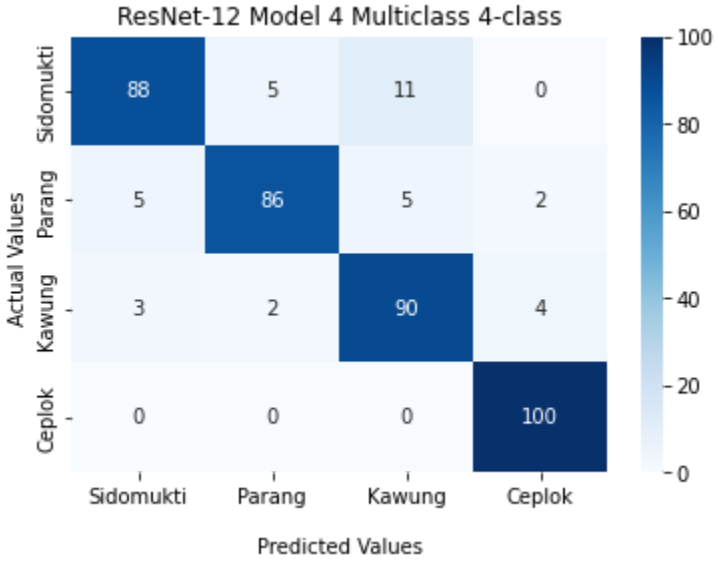
Model 1: Default	  30 epoch, Accuracy Train = 0.8670, Accuracy Test = 0.6783 50 epoch, Accuracy Train = 0.9269, Accuracy Test = 0.8504

Model 2: Penambahan 1 Dropout (0.1)	  30 epoch, Accuracy Train = 0.8651, Accuracy Test = 0.8753 50 epoch, Accuracy Train = 0.9407, Accuracy Test = 0.8628
Model 3: Penambahan 1 Dropout (0.1), Merubah Global Avg Pool ke Global Max Pool	

	 30 epoch, Accuracy Train = 0.9894, Accuracy Test = 0.8279 33 epoch, Accuracy Train = 0.9913, Accuracy Test = 0.6010
Model 4: Penambahan 2 Dropout (0.1)	  30 epoch, Accuracy Train = 0.8994, Accuracy Test = 0.8504 50 epoch, Accuracy Train = 0.9269, Accuracy Test = 0.9077

Model 5: Penambahan 2 Dropout (0.01)	  30 epoch, Accuracy Train = 0.8770, Accuracy Test = 0.8379 50 epoch, Accuracy Train = 0.9363, Accuracy Test = 0.8329
Model 6: Penambahan 2 Dropout (0.1), Merubah Global Avg Pool ke Global Max Pool	

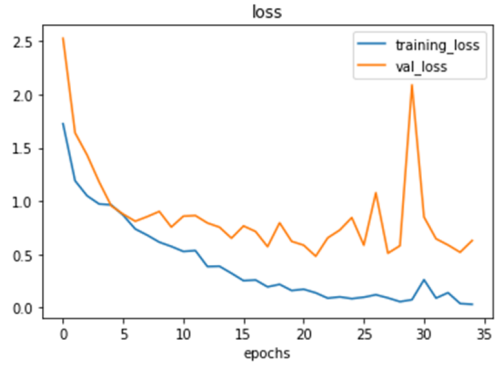
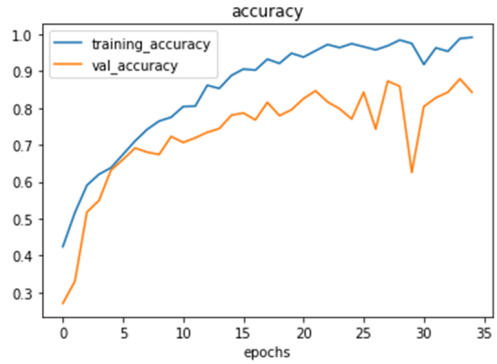
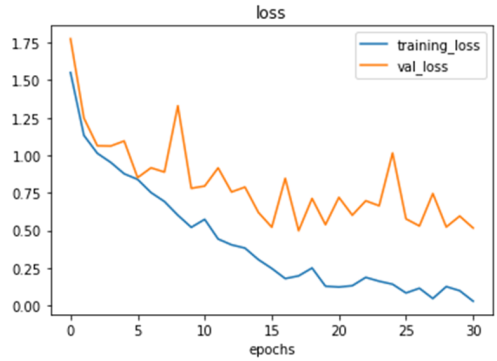
	 30 epoch, Accuracy Train = 0.8351, Accuracy Test = 0.8130 38 epoch, Accuracy Train = 0.9913, Accuracy Test = 0.8978

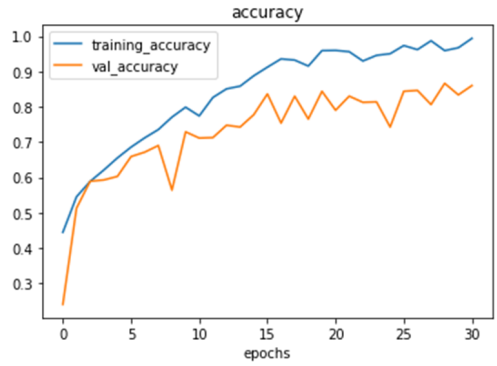


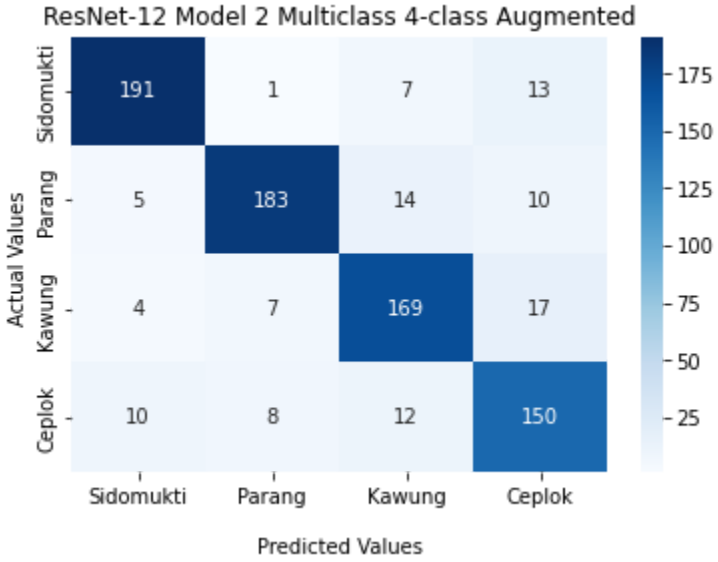
Skenario Terbaik:

Skenario Terburuk:

iii. Multi class Classification (4 Class Augmentasi)

Model 1: Penambahan 2 Dropout (0.1), Merubah Global Avg Pool ke Global Max Pool	  30 epoch, Accuracy Train = 0.9747, Accuracy Test = 0.6255 35 epoch, Accuracy Train = 0.9913, Accuracy Test = 0.8427
Model 2: Penambahan 3 Dropout (0.1), Merubah Global Avg Pool ke Global Max Pool	

	 30 epoch, Accuracy Train = 0.9672, Accuracy Test = 0.8340 31 epoch, Accuracy Train = 0.9938, Accuracy Test = 0.8602



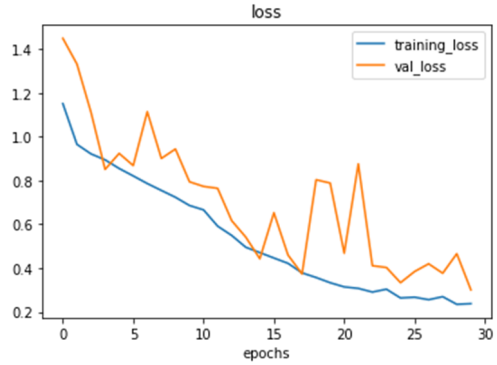
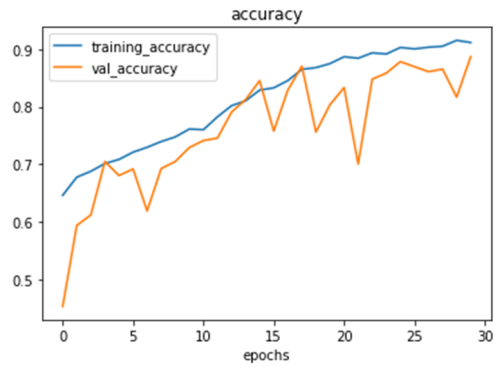
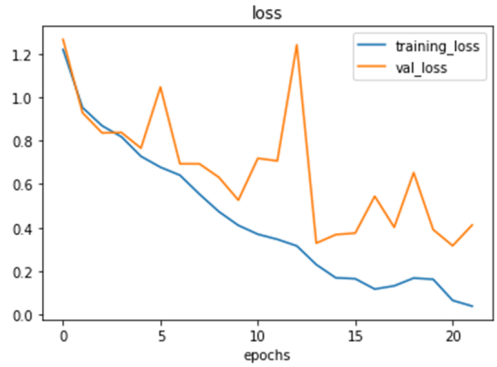
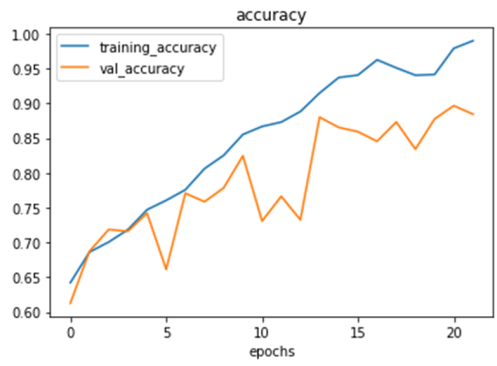
Skenario Terbaik:

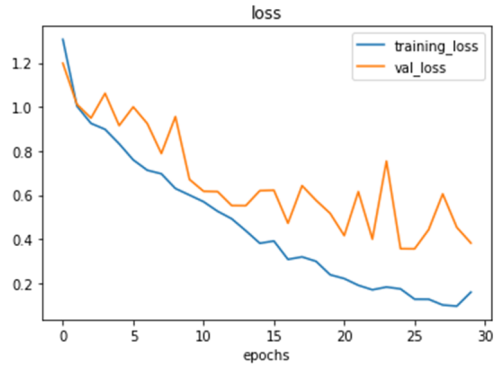
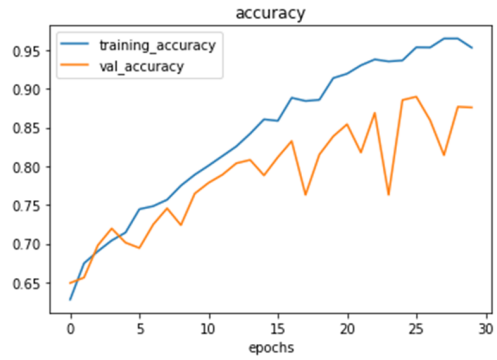
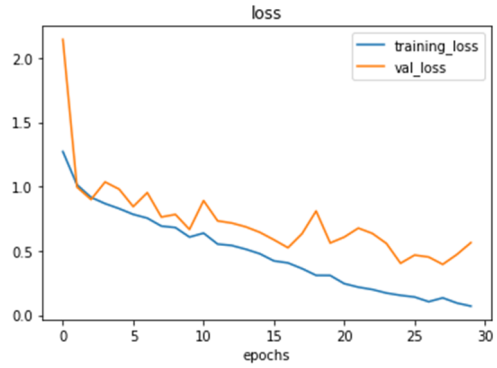
Label: Sidomukti; Predicted: Sidomukti
Confidence: 100%

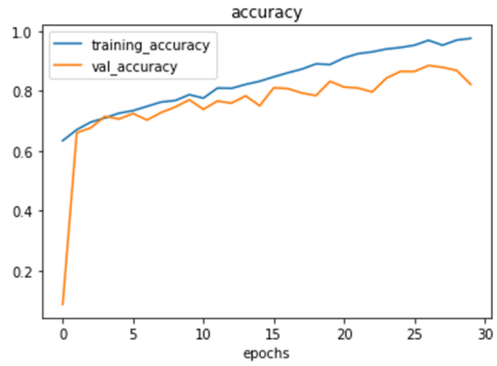
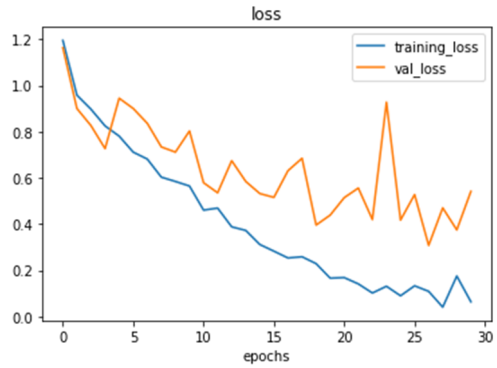
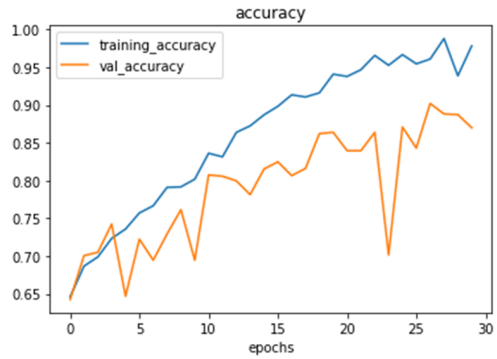
Skenario Terburuk:

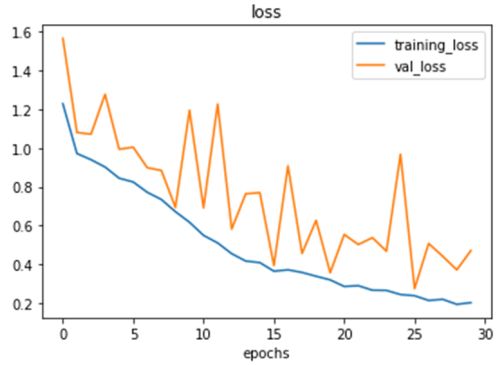
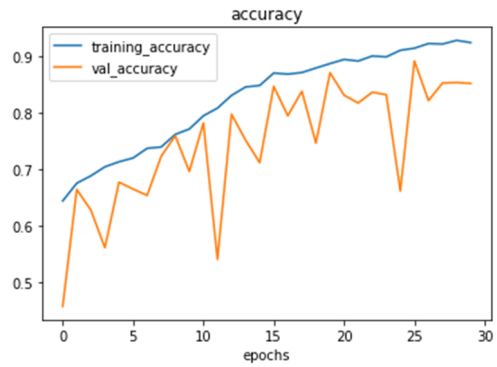
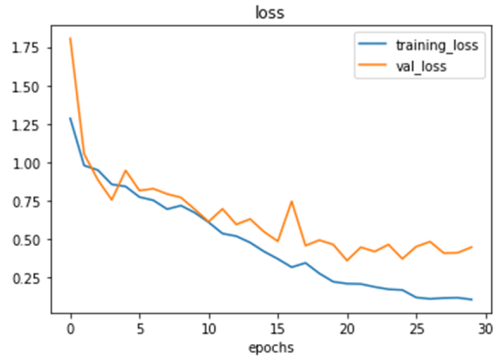
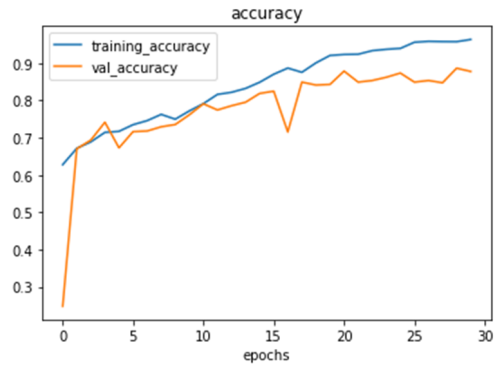
Label: Sidomukti; Predicted: Parang
Confidence: 100%

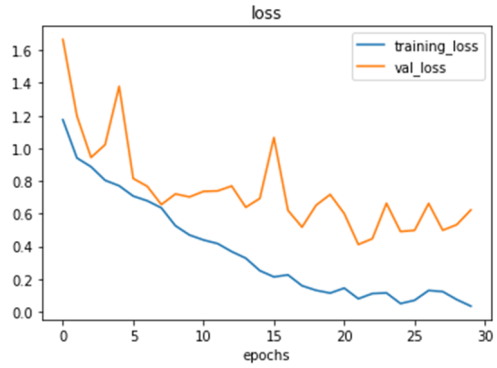
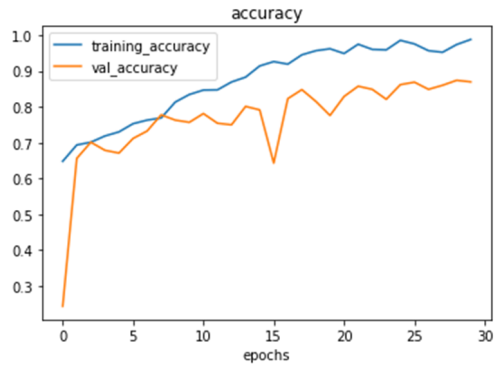
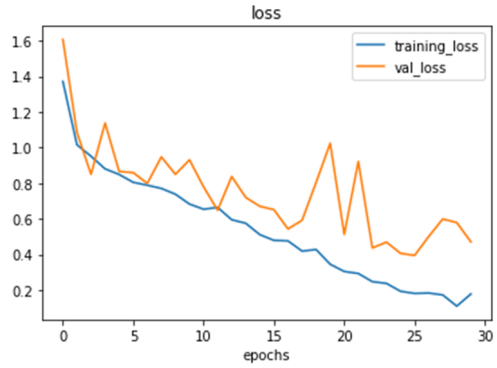
iv. Multi class Classification (5 Class)

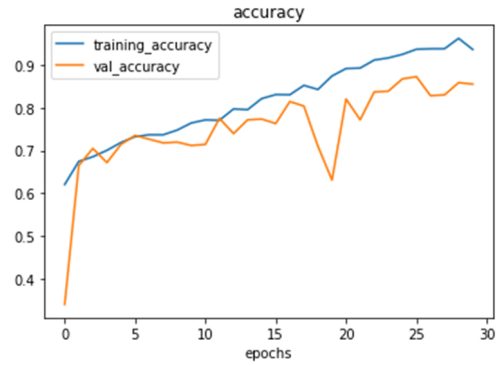
Model 1: Default	  30 epoch, Accuracy Train = 0.9118, Accuracy Test = 0.8872
Model 2: Merubah Global Avg Pool ke Global Max Pool	  22 epoch, Accuracy Train = 0.9900,

	Accuracy Test = 0.8845
Model 3: Penambahan 2 Dropout (0.1), Merubah Global Avg Pool ke Global Max Pool	  30 epoch, Accuracy Train = 0.9531, Accuracy Test = 0.8759
Model 3: Penambahan 3 Dropout (0.1), Merubah Global Avg Pool ke Global Max Pool	

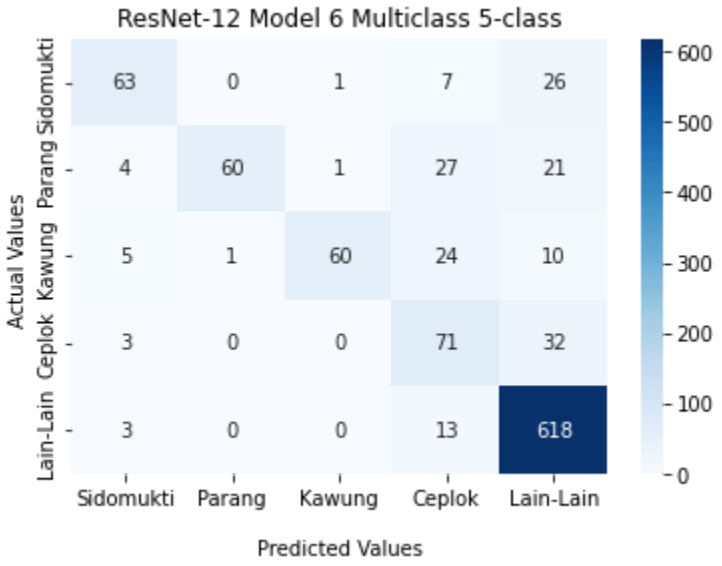
	 30 epoch, Accuracy Train = 0.9761, Accuracy Test = 0.8220
Model 4: Penambahan 1 Dropout (0.1), Merubah Global Avg Pool ke Global Max Pool	  30 epoch, Accuracy Train = 0.9781, Accuracy Test = 0.8698

Model 5: Penambahan 1 Dropout (0.1)	  30 epoch, Accuracy Train = 0.9251, Accuracy Test = 0.8524
Model 6: Penambahan 3 Dropout (0.15, 0.1, 0.1), Merubah Global Avg Pool ke Global Max Pool	  30 epoch, Accuracy Train = 0.9633,

	Accuracy Test = 0.8776
Model 7: Penambahan 1 Dropout (0.1), Merubah Global Avg Pool ke Global Max Pool	  30 epoch, Accuracy Train = 0.9881, Accuracy Test = 0.8698
Model 8: Penambahan 3 Dropout (0.15), Merubah Global Avg Pool ke Global Max Pool	



30 epoch, Accuracy Train = 0.9359, Accuracy Test = 0.8550



Skenario Terbaik:

Skenario Terburuk:

c. MobileNet

Layer Input MobileNet dari (224, 224, 3) diubah menjadi (250, 250, 3). Sesudah layer Mobilenet dan sebelum layer Classifier akan digunakan layer GlobalAveragePooling2D.

- Model A: Transfer Learning MobileNet dengan weight Imagenet dengan layer akhir (classifier) yang dimodifikasi
- Model B: Transfer Learning MobileNet dengan weight Imagenet lalu layer 73 - 85 mobilenet di unfrozen
- Model C: MobileNet training dari awal dengan penghapusan layer ZeroPadding2D

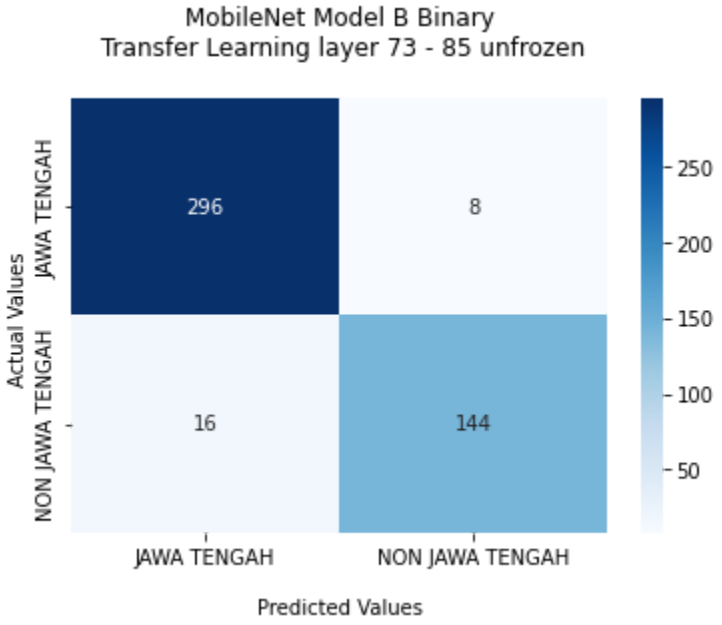
Pendataan diambil dari yang memiliki val_accuracy terbaik

i. Binary Classification

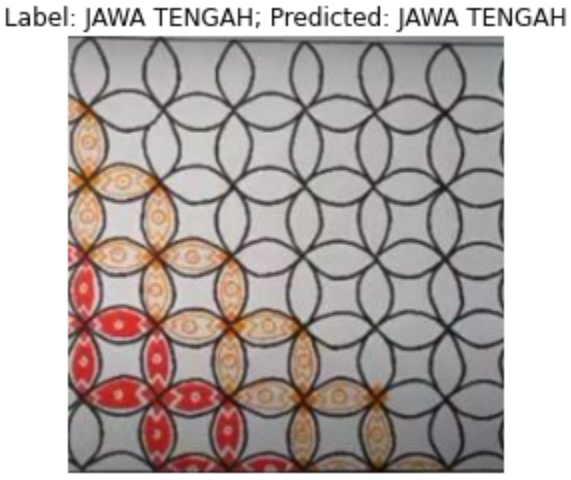
Model A: Head Dense dengan unit 1: Sigmoid	10 epoch Accuracy Train = 0.8864 Accuracy Test = 0.8906
Model B: Head Dense dengan unit 1: Sigmoid	10 epoch Accuracy Train = 0.9398 Accuracy Test = 0.9218

Model C: Head Dense dengan unit 1: Sigmoid	20 epoch Accuracy Train = 0.9187 Accuracy Test = 0.8654

Model Terbaik:

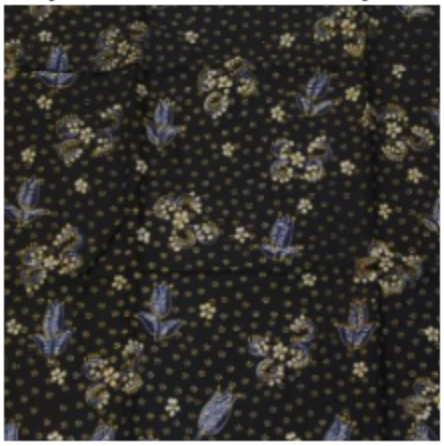


Skenario Terbaik:

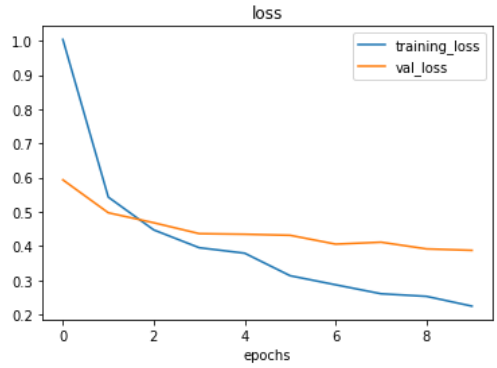
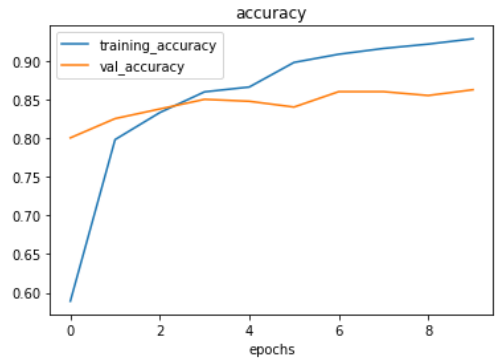


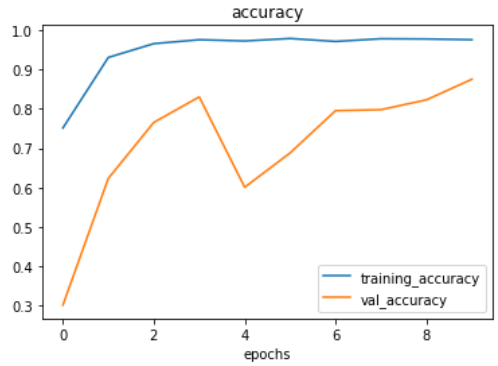
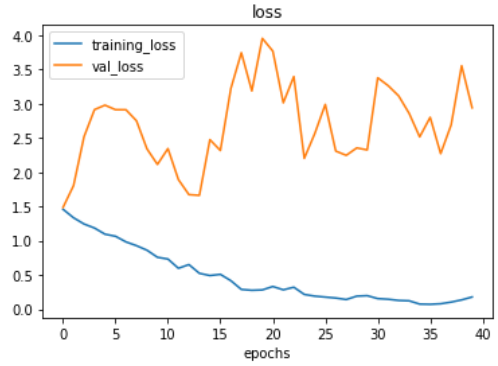
Skenario Terburuk:

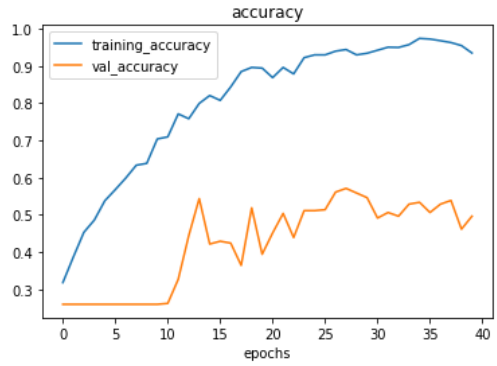
Label: NON JAWA TENGAH; Predicted: JAWA TENGAH



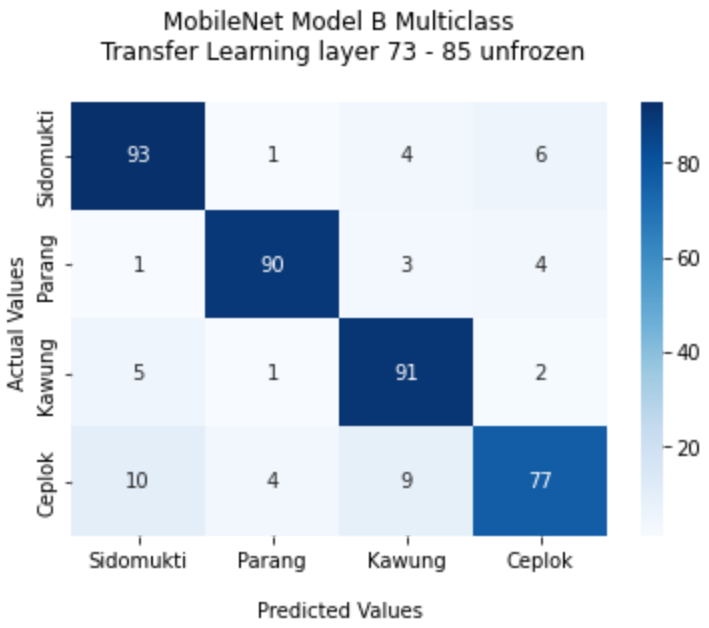
ii. Multi class Classification (4 Class)

Model A: Head Dense dengan unit 4: Softmax	10 epoch Accuracy Train = 0.9387 Accuracy Test = 0.8628  
Model B: Head Dense dengan unit 4: Softmax	10 epoch Accuracy Train = 0.9737 Accuracy Test = 0.8753

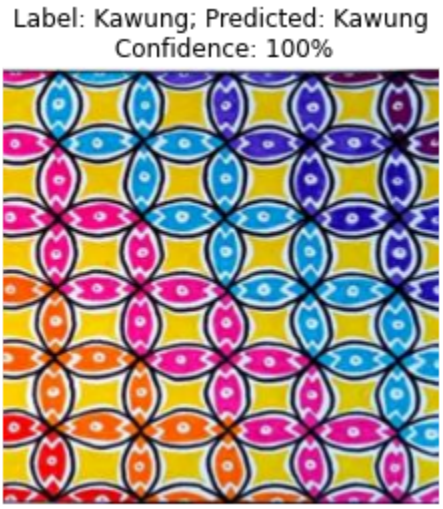
	 
Model C: Head Dense dengan unit 4: Softmax	40 epoch Accuracy Train = 0.9737 Accuracy Test = 0.5710 

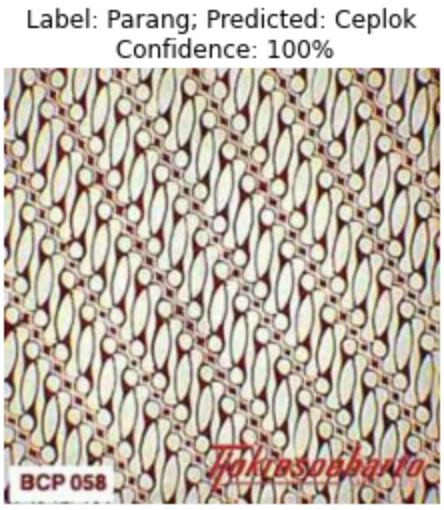
Model Terbaik:



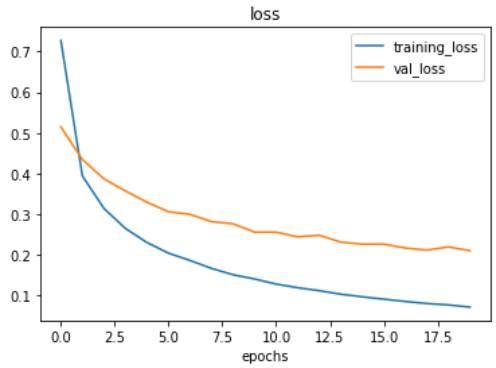
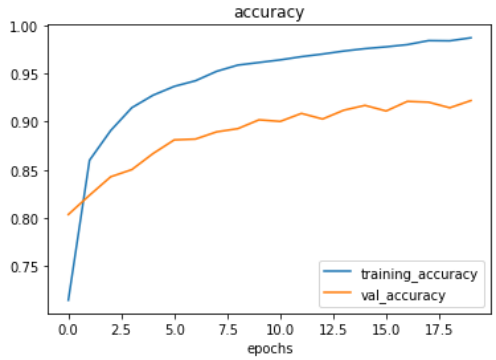
Skenario Terbaik:

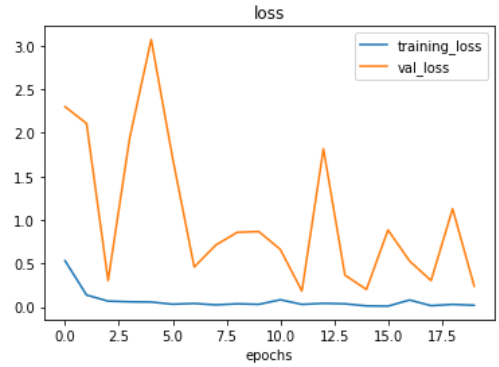
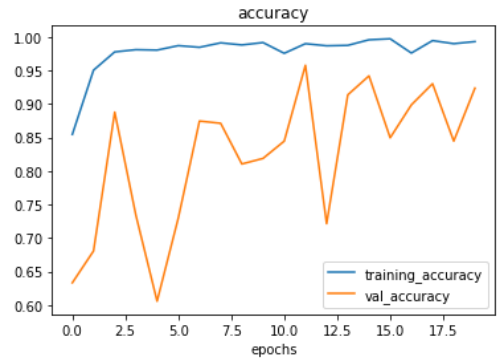
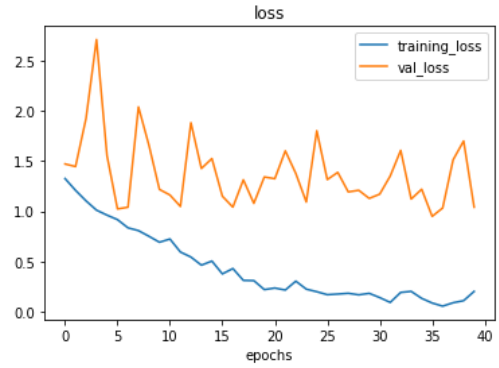


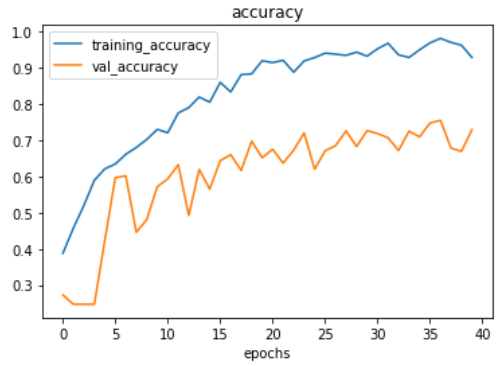
Skenario Terburuk:



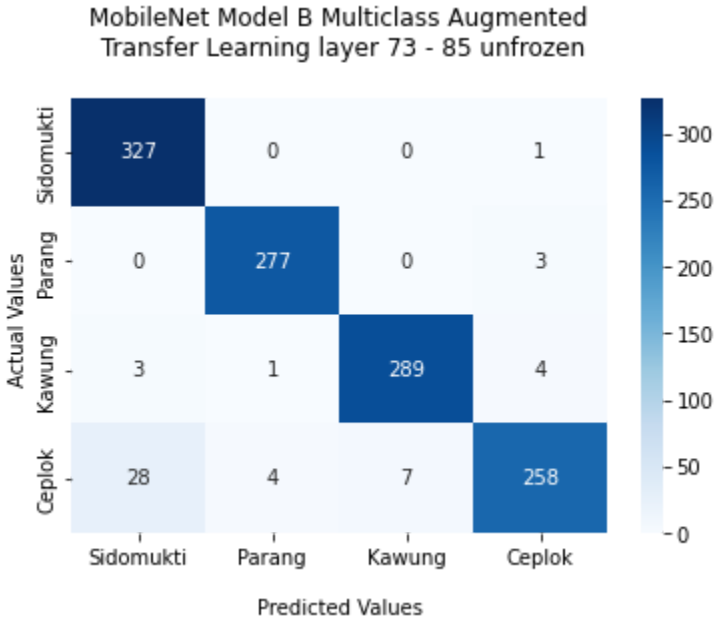
iii. Multi class Classification (4 Class Augmentasi)

Model A: Head Dense dengan unit 4: Softmax	20 epoch Accuracy Train = 0.9908 Accuracy Test = 0.9217  
Model B: Head Dense dengan unit 4: Softmax	20 epoch Accuracy Train = 0.9860

	Accuracy Test = 0.9575  
Model C: Head Dense dengan unit 4: Softmax	40 epoch Accuracy Train = 0.9725 Accuracy Test = 0.7545 

Model Terbaik:



Skenario Terbaik:

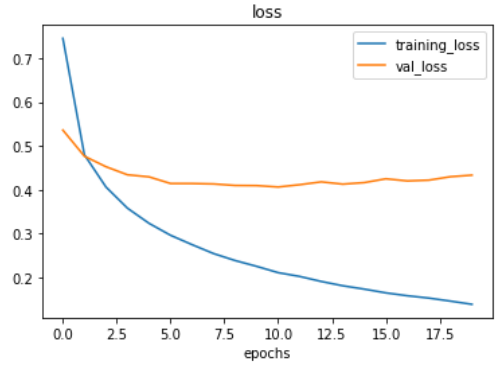
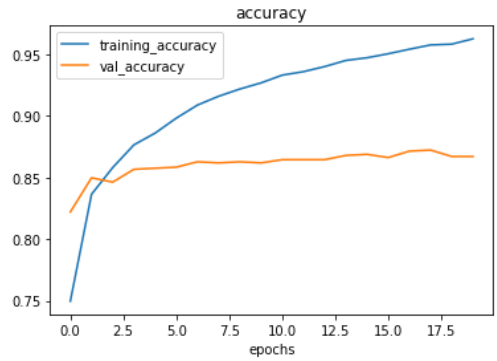
Label: Parang; Predicted: Parang
Confidence: 100%

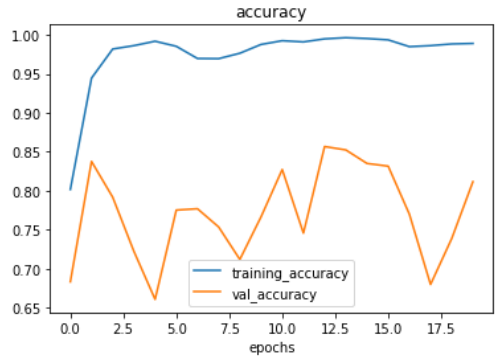
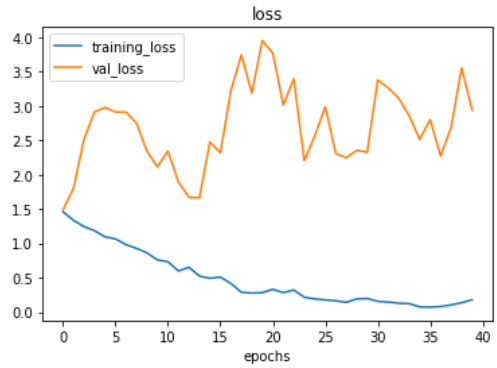
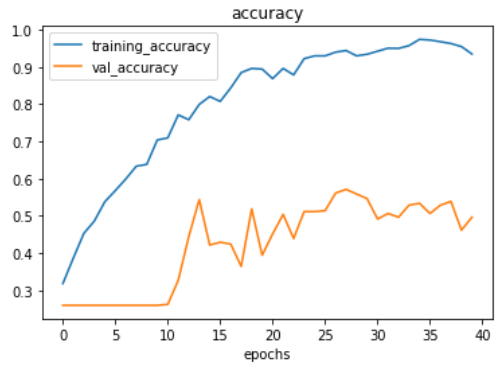
Skenario Terburuk:

Label: Ceplok; Predicted: Sidomukti
Confidence: 77%

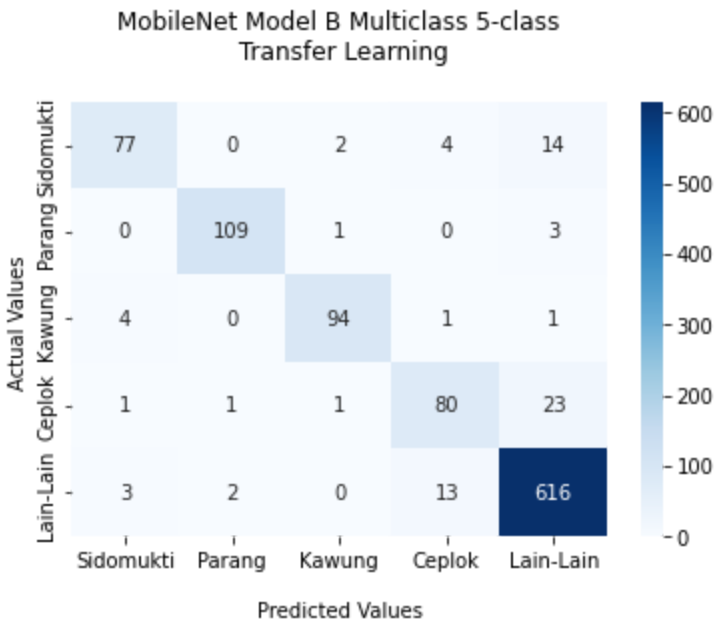
iv. Multi class Classification (5 Class)

Model A: Head Dense dengan unit 5: Softmax	20 epoch Accuracy Train = 0.9648 Accuracy Test = 0.8723

	 
Model B: Head Dense dengan unit 5: Softmax	20 epoch Accuracy Train = 0.9787 Accuracy Test = 0.8567 

	
Model C: Head Dense dengan unit 5: Softmax	40 epoch Accuracy Train = 0.9402 Accuracy Test = 0.7317  

Model Terbaik:

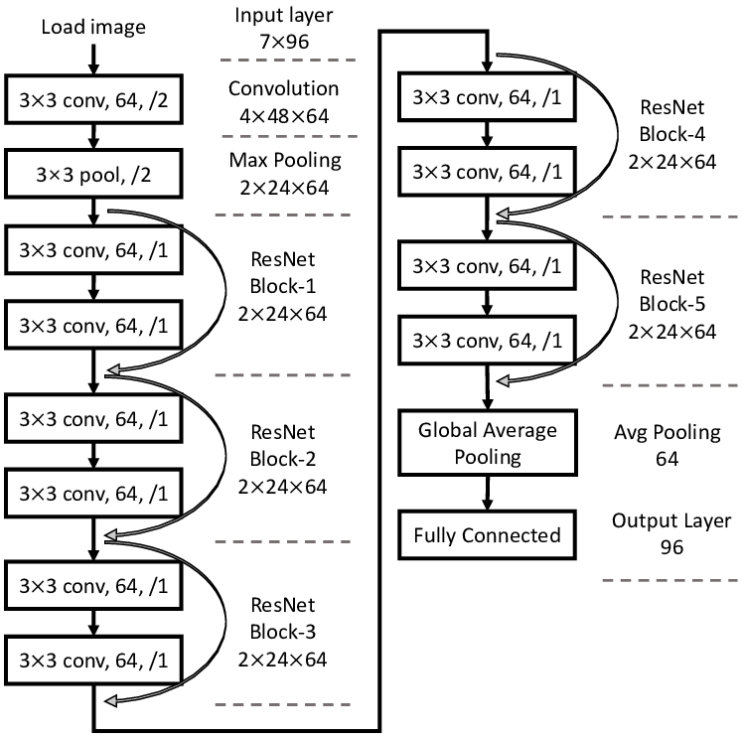


Skenario Terbaik:

Skenario Terburuk:

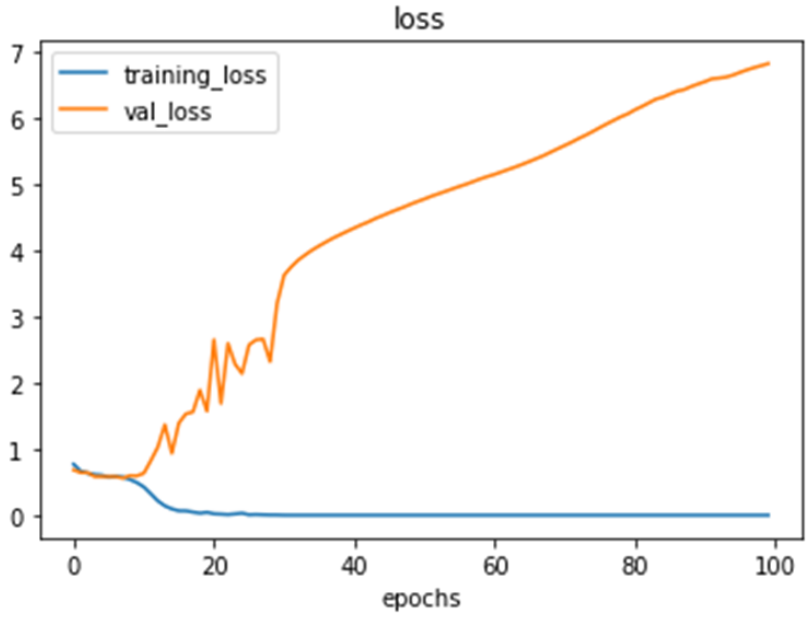
8. Kendala selama Research

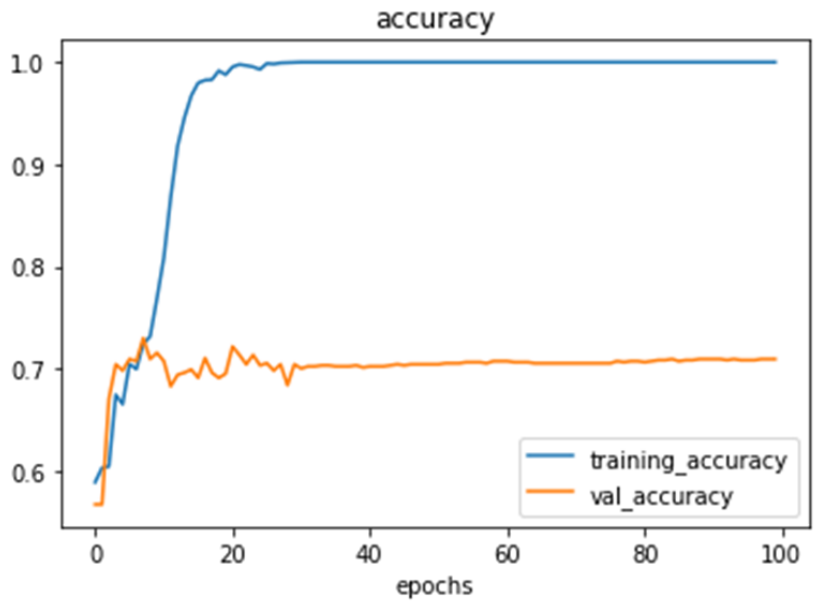
1. Error ResourceExhausted: Out of Memory
2. Kesulitan Training model besar
3. Limit GPU Google Colab
4. History akurasi training hanya tersimpan dalam bentuk plot
5. Kendala VGG-16
    a. Penggunaan layer fully connected memiliki kecenderungan membuat model tidak dapat belajar
6. Kendala Resnet-12
    a. Gagal pada Model Sequential karena model yang salah dan menyebabkan akurasi *stuck*




```
model = Sequential()

model.add(Conv2D(64, kernel_size = (3, 3), activation
= "relu", padding = "Same", input_shape=(250, 250,
3)))

model.add(MaxPooling2D(pool_size = (2, 2),
strides=(2,2)))

# Resnet Block

model.add(Conv2D(64, kernel_size = (3, 3), activation
= "relu", padding = "Same"))

model.add(Conv2D(64, kernel_size = (3, 3), activation
= "relu", padding = "Same"))

# Resnet Block

model.add(Conv2D(64, kernel_size = (3, 3), activation
= "relu", padding = "Same"))

model.add(Conv2D(64, kernel_size = (3, 3), activation
= "relu", padding = "Same"))

# Resnet Block

model.add(Conv2D(64, kernel_size = (3, 3), activation
= "relu", padding = "Same"))
```

```
model.add(Conv2D(64, kernel_size = (3, 3), activation
= "relu", padding = "Same"))

# Resnet Block

model.add(Conv2D(64, kernel_size = (3, 3), activation
= "relu", padding = "Same"))

model.add(Conv2D(64, kernel_size = (3, 3), activation
= "relu", padding = "Same"))

# Resnet Block

model.add(Conv2D(64, kernel_size = (3, 3), activation
= "relu", padding = "Same"))

model.add(Conv2D(64, kernel_size = (3, 3), activation
= "relu", padding = "Same"))

model.add(AveragePooling2D(pool_size = (2, 2), padding
= "Same"))

model.add(Flatten())

model.add(Dense(512, activation = "relu"))

model.add(Dense(2, activation = "sigmoid"))
```





Accuracy: 1.0000, Val Accuracy: 0.7096

7. Kendala MobileNet
    a. Training Model C cenderung memiliki val_accuracy yang berombang ambing dan tidak bisa menyamai Transfer Learning Model A dan Model C.