

BAB VI. HASIL DAN PEMBAHASAN

6.1 Hasil Implementasi Uji Coba

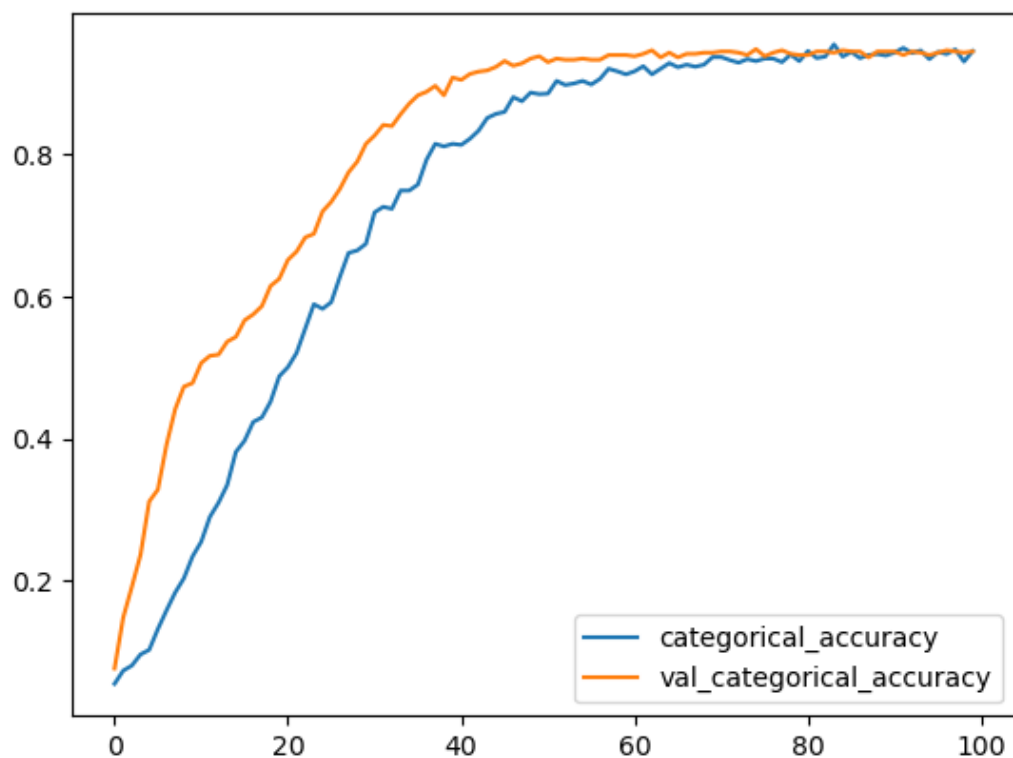
Hasil mplementasi Uji Coba *Image Classification* pada Jenis-Jenis Batik Menggunakan *Convolutional Neural Network* dengan Model *EfficientNet* dengan beberapa skenario yang sudah dirancang untuk menemukan model pelatihan terbaik

6.1.1 Hasil Uji Coba Skenario 1

Pada uji coba skenario 1 ini dilakukan dengan melakukan proses *training* dengan menggunakan *Convolutional Neural Network* tanpa menggunakan *EfficientNet*. Berikut merupakan hasil uji coba skenario 1

6.1.1.1 Grafik *Training Accuracy*

Berdasarkan proses *training* akan mendapatkan hasil berupa grafik *training* sebagai berikut

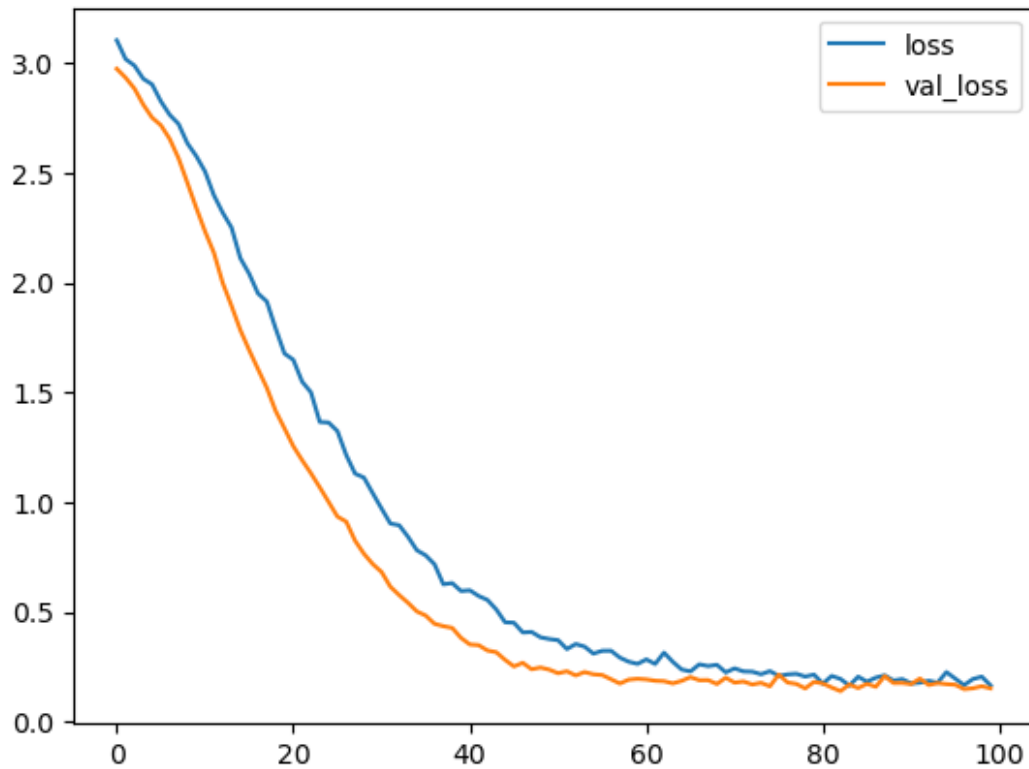


Gambar 6. 1 Hasil Grafik *Training Accuracy* Skenario 1

Berdasarkan grafik proses *training* pada gambar 6.1 menggunakan *Convolutional Neural Network* tanpa menggunakan *EfficientNet* mengalami

kenaikan *accuracy* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di atas 80% yang berakhir pada epoch ke 100.

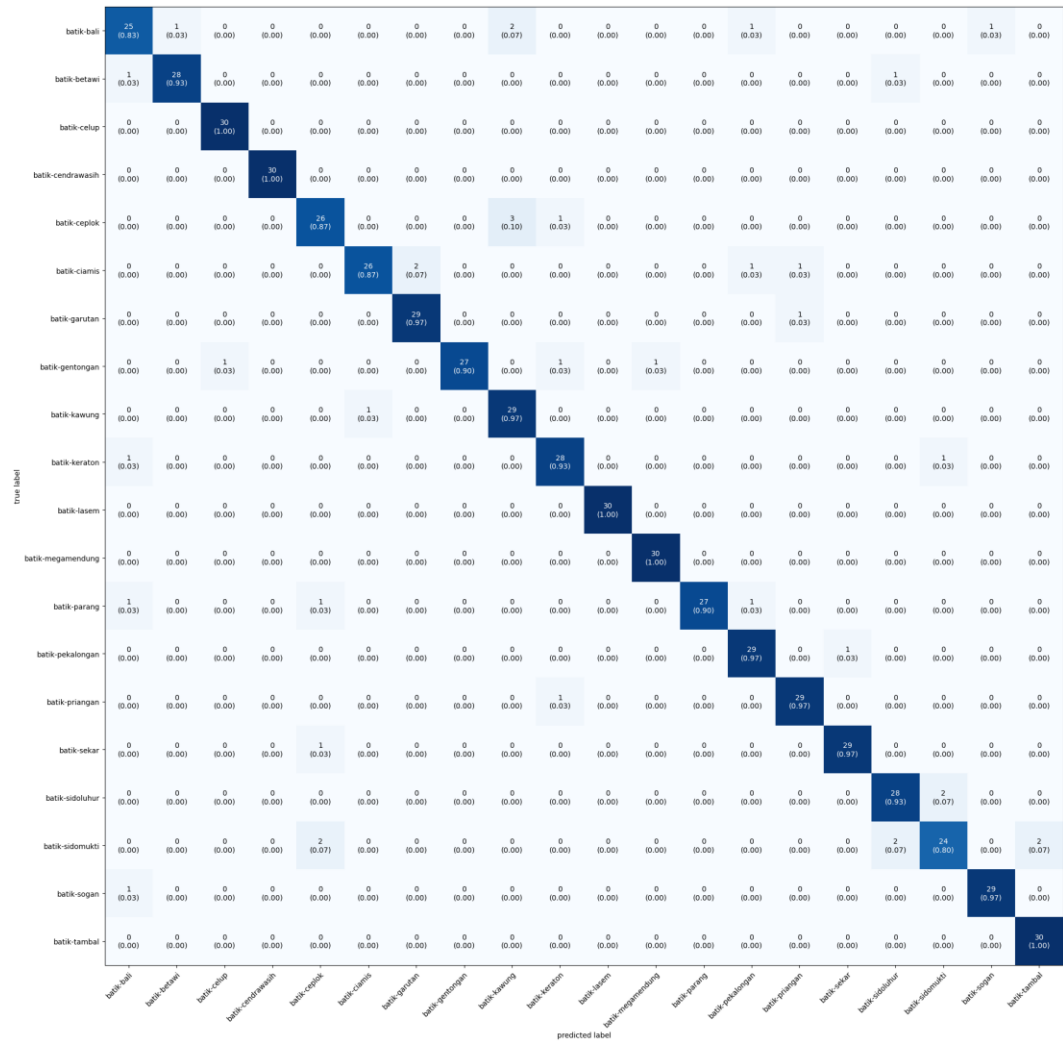
6.1.1.2 Grafik *Training Loss*



Gambar 6. 2 Hasil Grafik *Training Loss* Skenario 1

Berdasarkan grafik proses *training* pada gambar 6.2 menggunakan *Convolutional Neural Network* tanpa menggunakan *EfficientNet* mengalami penurunan *loss* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di angka mendekati 0 yang berakhir pada epoch ke 100.

6.1.1.3 Confusion Matrix



Gambar 6. 3 Confusion Matrix Skenario 1

Berdasarkan *confusion matrix* pada gambar 6.3 dapat dilihat bahwa proses *training* menggunakan *Convolutional Neural Network* tanpa menggunakan *EfficientNet* berhasil mengklasifikasikan dengan benar sebanyak 563 gambar. Berdasarkan hasil tersebut didapat hasil Batik Sidomukti yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Sidomukti hanya mendapatkan 24 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan data latih terhadap hasil pengujian. Pengujian dilakukan dengan membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil *confusion matrix* ini bisa kita klasifikasikan sebagai berikut.

Tabel 6. 1 Tabel Klasifikasi Data Skenario 1

Kelas	Klasifikasi				Jumlah
	TP	FN	FP	TN	
Batik Bali	25	5	4	566	600
Batik Betawi	28	2	1	569	600
Batik Celup	30	0	1	569	600
Batik Cendrawasih	30	0	0	570	600
Batik Ceplok	26	4	4	566	600
Batik Ciamis	26	4	1	569	600
Batik Garutan	29	1	2	568	600
Batik Gentongan	27	3	0	570	600
Batik Kawung	29	1	5	565	600
Batik Keraton	28	2	3	567	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	1	569	600
Batik Parang	27	3	0	570	600
Batik Pekalongan	29	1	3	567	600
Batik Priangan	29	1	2	568	600
Batik Sekar	29	1	1	569	600
Batik Sidoluhur	28	2	3	567	600
Batik Sidomukti	24	6	3	567	600
Batik Sogan	29	1	1	569	600
Batik Tambal	30	0	2	568	600

Berdasarkan data klasifikasi pada Tabel 6.1 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 2 Tabel Hasil Nilai dari Klasifikasi Data Skenario 1

Kelas	Hasil			
	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Batik Bali	0.985	0.862	0.833	0.847
Batik Betawi	0.995	0.965	0.933	0.948
Batik Celup	0.998	0.967	1	0.983
Batik Cendrawasih	1	1	1	1
Batik Ceplok	0.986	0.866	0.866	0.866
Batik Ciamis	0.991	0.962	0.866	0.911
Batik Garutan	0.995	0.935	0.966	0.950
Batik Gentongan	0.995	1	0.9	0.947
Batik Kawung	0.99	0.852	0.966	0.905
Batik Keraton	0.991	0.903	0.933	0.917
Batik Lasem	1	1	1	1
Batik Megamendung	0.998	0.967	1	0.983
Batik Parang	0.995	1	0.9	0.947
Batik Pekalongan	0.993	0.906	0.966	0.935
Batik Priangan	0.995	0.935	0.966	0.950
Batik Sekar	0.996	0.966	0.966	0.966
Batik Sidoluhur	0.991	0.903	0.933	0.917
Batik Sidomukti	0.985	0.888	0.8	0.841
Batik Sogan	0.996	0.966	0.966	0.966
Batik Tambal	0.996	0.937	1	0.967

```

19/19 [=====] - 7s 376ms/step
Prediction Accuracy : 0.9383333333333334
      precision    recall  f1-score   support

     0       0.86      0.83      0.85        30
     1       0.97      0.93      0.95        30
     2       0.97      1.00      0.98        30
     3       1.00      1.00      1.00        30
     4       0.87      0.87      0.87        30
     5       0.96      0.87      0.91        30
     6       0.94      0.97      0.95        30
     7       1.00      0.90      0.95        30
     8       0.85      0.97      0.91        30
     9       0.90      0.93      0.92        30
    10       1.00      1.00      1.00        30
    11       0.97      1.00      0.98        30
    12       1.00      0.90      0.95        30
    13       0.91      0.97      0.94        30
    14       0.94      0.97      0.95        30
    15       0.97      0.97      0.97        30
    16       0.90      0.93      0.92        30
    17       0.89      0.80      0.84        30
    18       0.97      0.97      0.97        30
    19       0.94      1.00      0.97        30

 accuracy                   0.94        600
 macro avg       0.94      0.94      0.94        600
 weighted avg    0.94      0.94      0.94        600

```

Gambar 6. 4 Hasil Accuracy Skenario 1

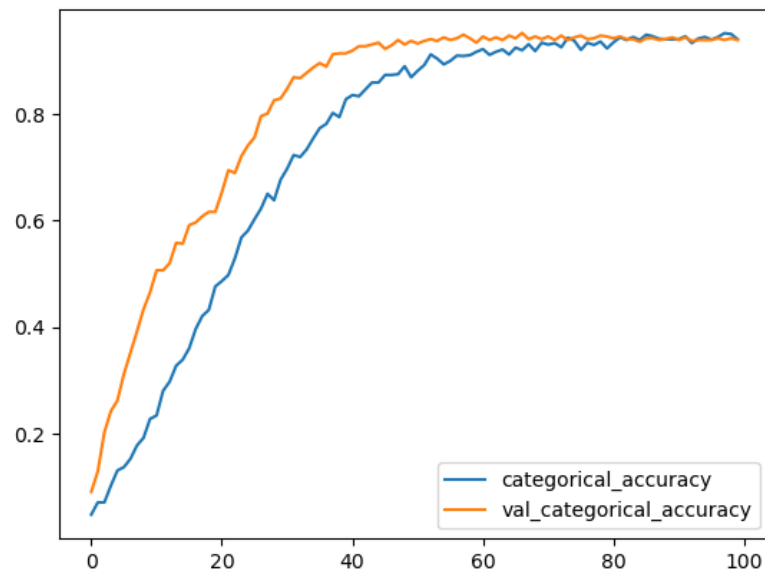
Berdasarkan hasil *accuracy* pada gambar 6.4 dapat dilihat bahwa proses *training* menggunakan *Convolutional Neural Network* tanpa menggunakan *EfficientNet* memperoleh skor sebesar 94% dengan membutuhkan waktu pelatihan selama 2490 detik. Berikut kesimpulan dari hasil pelatihan pada skenario 1

Skenario	Accuracy pada Sistem	Accuracy pada perhitungan manual	Precision	Recall	F1-score
CNN	0.94	0.99	0.94	0.94	0.94

6.1.2 Hasil Uji Coba Skenario 2

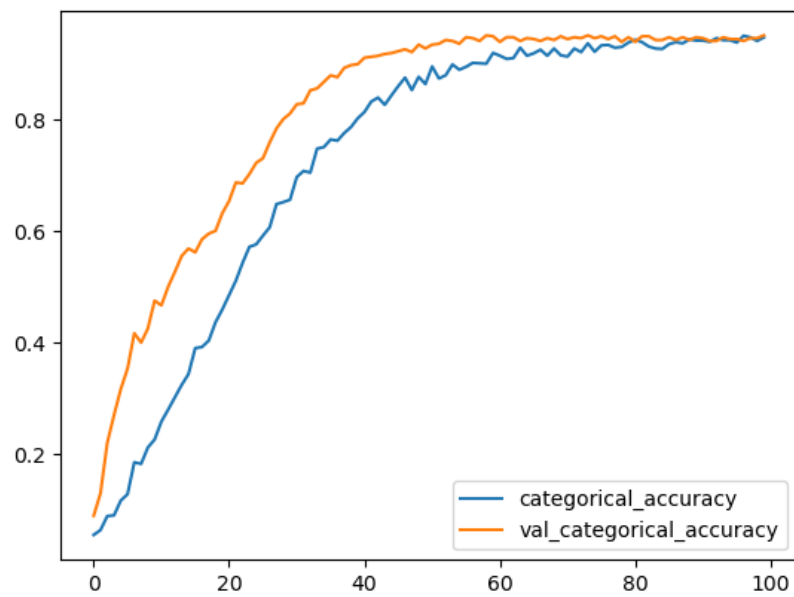
Pada uji coba skenario 2 ini dilakukan dengan membandingkan model dengan menggunakan data *preprocessing* dengan model tanpa *preprocessing*. *Preprocessing* yang digunakan pada uji coba ini yaitu menggunakan filter GaussianNoise dan akan melakukan pelatihan sebanyak 100 *epoch*. Berikut merupakan hasil uji coba skenario 2.

6.1.2.1 Grafik *Training Accuracy*



Gambar 6. 5 Hasil Grafik *Training Accuracy* Tanpa Data *Preprocessing*

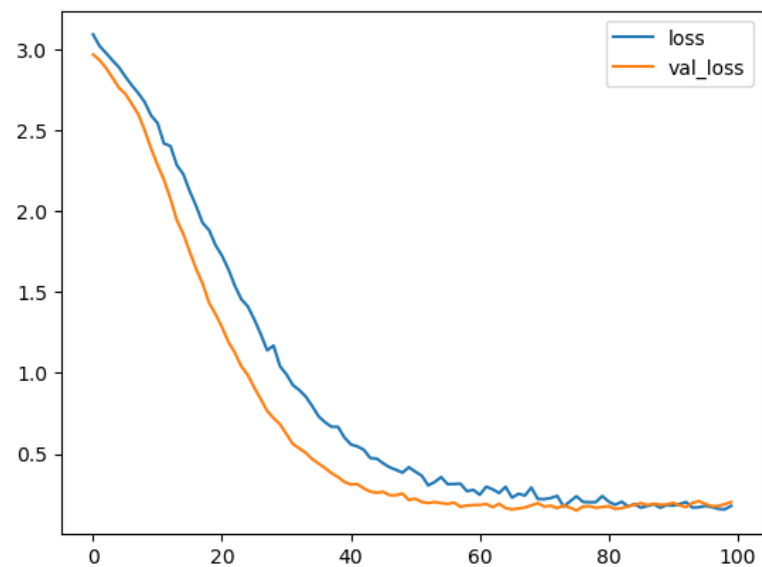
Berdasarkan grafik proses *training* pada gambar 6.5 tanpa data *preprocessing* mengalami kenaikan *accuracy* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di atas 80% yang berakhir pada epoch ke 100.



Gambar 6. 6 Hasil Grafik *Training Accuracy* dengan *Data Preprocessing*

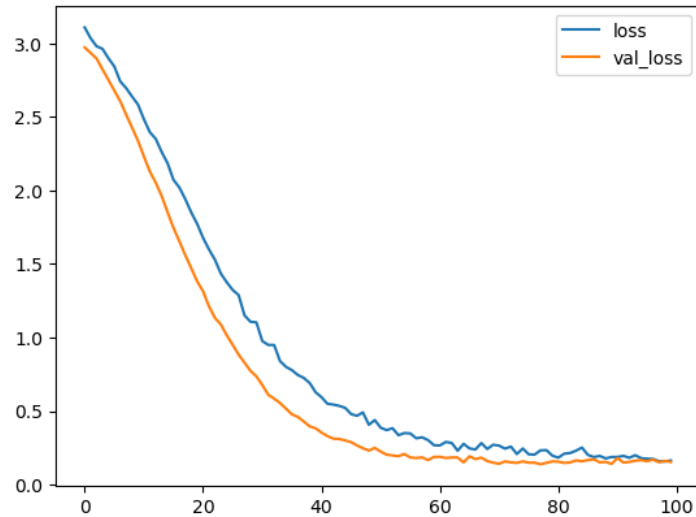
Berdasarkan grafik proses *training* pada gambar 6.6 dengan data *preprocessing* mengalami kenaikan *accuracy* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di atas 80% yang berakhir pada epoch ke 100.

6.1.2.2 Grafik *Training Loss*



Gambar 6. 7 Hasil Grafik *Training Loss* Tanpa *Data Preprocessing*

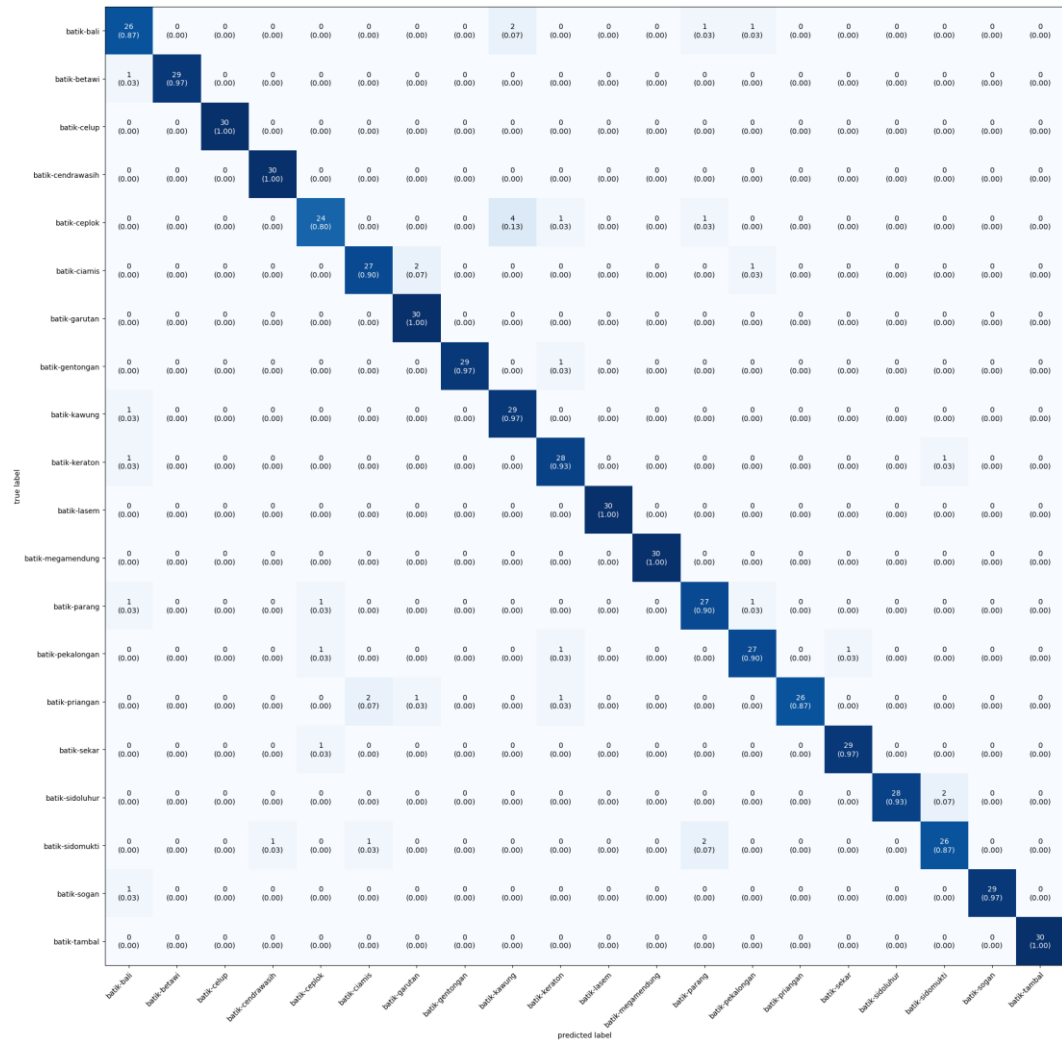
Berdasarkan grafik proses *training* pada gambar 6.7 tanpa data *preprocessing* mengalami penurunan *loss* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di angka mendekati 0 yang berakhir pada epoch ke 100.



Gambar 6. 8 Hasil Grafik *Training Loss* dengan *Data Preprocessing*

Berdasarkan grafik proses *training* pada gambar 6.7 tanpa data *preprocessing* mengalami penurunan *loss* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di angka mendekati 0 yang berakhir pada epoch ke 100.

6.1.2.3 Confusion Matrix



Gambar 6.9 Confusion Matrix Tanpa Data Preprocessing

Berdasarkan *confusion matrix* pada gambar 6.9 dapat dilihat bahwa proses *training* tanpa menggunakan data *preprocessing* berhasil mengklasifikasikan dengan benar sebanyak 564 gambar. Berdasarkan hasil tersebut didapat hasil Batik Bali yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Ceplok hanya mendapatkan 24 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan data latih terhadap hasil pengujian. Pengujian dilakukan dengan membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil *confusion matrix* ini bisa kita klasifikasikan sebagai berikut.

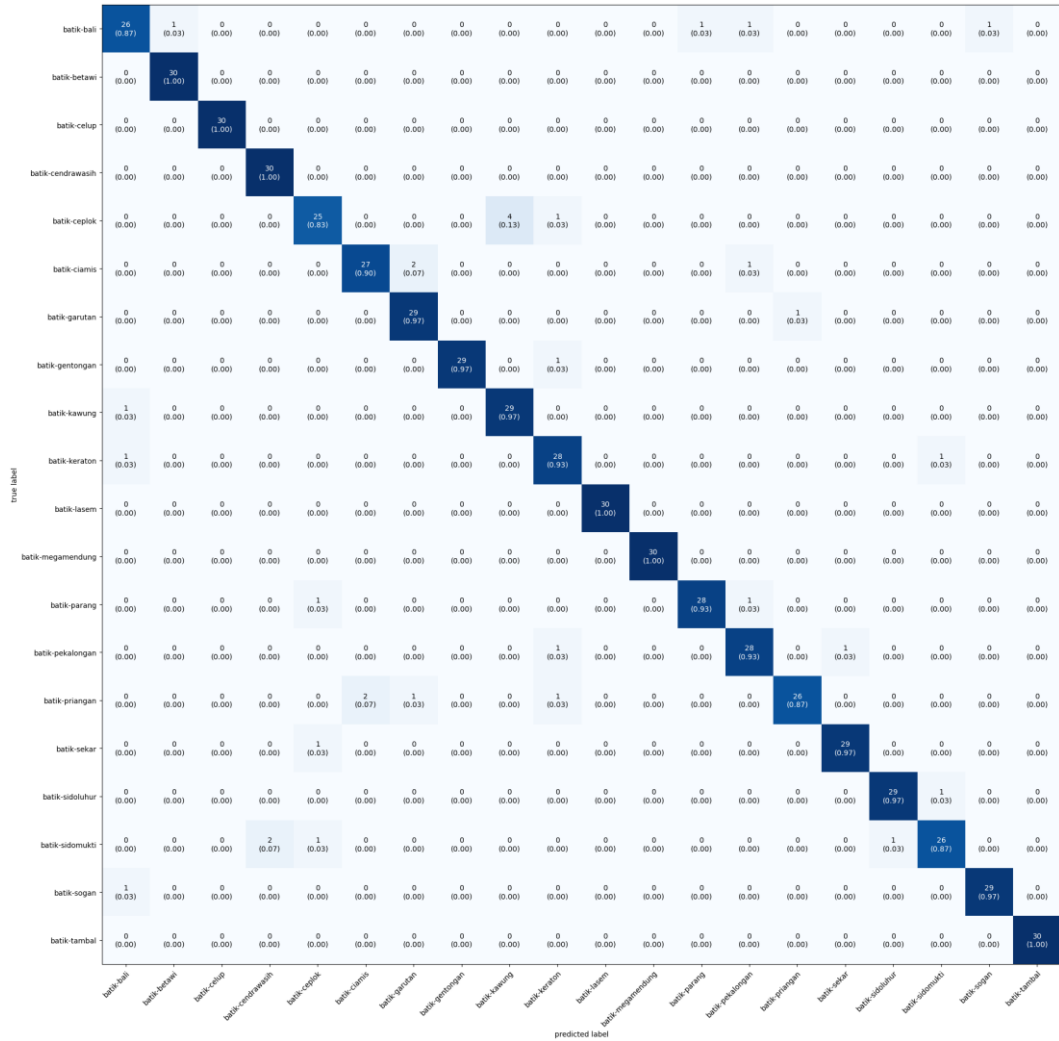
Tabel 6. 3 Tabel Klasifikasi Data Tanpa *Preprocessing*

Kelas	Klasifikasi				Jumlah
	TP	FP	FN	TN	
Batik Bali	26	4	5	565	600
Batik Betawi	29	1	0	570	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	30	0	1	569	600
Batik Ceplok	24	6	3	567	600
Batik Ciamis	27	3	3	567	600
Batik Garutan	30	0	3	567	600
Batik Gentongan	29	1	0	570	600
Batik Kawung	29	1	6	564	600
Batik Keraton	28	2	4	566	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	0	570	600
Batik Parang	27	3	4	566	600
Batik Pekalongan	27	3	3	567	600
Batik Priangan	26	4	0	570	600
Batik Sekar	29	1	1	569	600
Batik Sidoluhur	28	2	0	570	600
Batik Sidomukti	26	4	3	567	600
Batik Sogan	29	1	0	570	600
Batik Tambal	30	0	0	570	600

Berdasarkan data klasifikasi pada Tabel 6.3 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 4 Tabel Hasil Nilai dari Klasifikasi Data Tanpa *Preprocessing*

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.985	0.866667	0.83871	0.852459
Batik Betawi	0.998333	0.966667	1	0.983051
Batik Celup	1	1	1	1
Batik Cendrawasih	0.998333	1	0.967742	0.983607
Batik Ceplok	0.985	0.8	0.888889	0.842105
Batik Ciamis	0.99	0.9	0.9	0.9
Batik Garutan	0.995	1	0.909091	0.952381
Batik Gentongan	0.998333	0.966667	1	0.983051
Batik Kawung	0.988333	0.966667	0.828571	0.892308
Batik Keraton	0.99	0.933333	0.875	0.903226
Batik Lasem	1	1	1	1
Batik Megamendung	1	1	1	1
Batik Parang	0.988333	0.9	0.870968	0.885246
Batik Pekalongan	0.99	0.9	0.9	0.9
Batik Priangan	0.993333	0.866667	1	0.928571
Batik Sekar	0.996667	0.966667	0.966667	0.966667
Batik Sidoluhur	0.996667	0.933333	1	0.965517
Batik Sidomukti	0.988333	0.866667	0.896552	0.881356
Batik Sogan	0.998333	0.966667	1	0.983051
Batik Tambal	1	1	1	1



Gambar 6. 10 Confusion Matrix dengan Data Preprocessing

Berdasarkan *confusion matrix* pada gambar 6.10 dapat dilihat bahwa proses *training* tanpa menggunakan data *preprocessing* berhasil mengklasifikasikan dengan benar sebanyak 568 gambar. Berdasarkan hasil tersebut didapat hasil Batik Ceplok yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Ceplok hanya mendapatkan 24 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan data latih terhadap hasil pengujian. Pengujian dilakukan dengan membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil *confusion matrix* ini bisa kita klasifikasikan sebagai berikut.

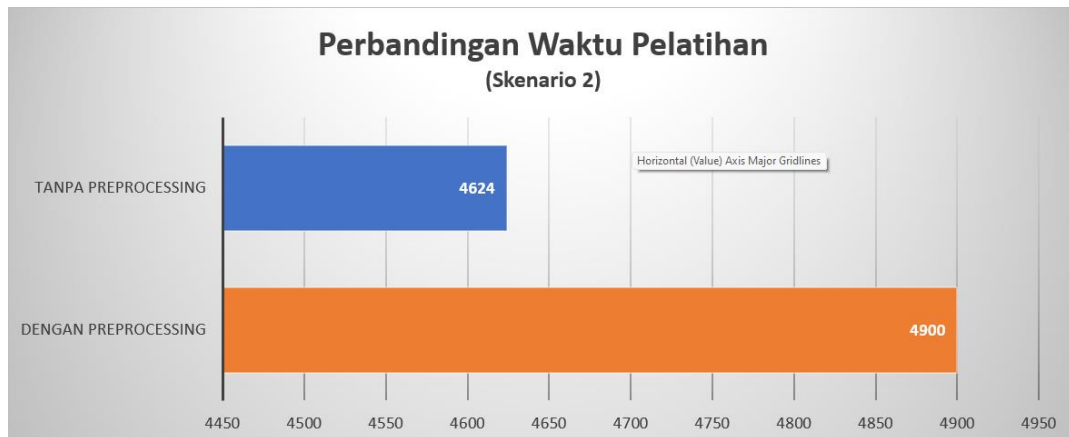
Tabel 6. 5 Tabel Hasil Nilai dari Klasifikasi Data dengan *Preprocessing*

Kelas	Klasifikasi				Jumlah
	TP	FP	FN	TN	
Batik Bali	26	4	3	567	600
Batik Betawi	30	0	1	569	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	30	0	2	568	600
Batik Ceplok	25	5	3	567	600
Batik Ciamis	27	3	2	568	600
Batik Garutan	29	1	3	567	600
Batik Gentongan	29	1	0	570	600
Batik Kawung	29	1	4	566	600
Batik Keraton	28	2	4	566	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	0	570	600
Batik Parang	28	2	1	569	600
Batik Pekalongan	28	2	3	567	600
Batik Priangan	26	4	1	569	600
Batik Sekar	29	1	1	569	600
Batik Sidoluhur	29	1	1	569	600
Batik Sidomukti	26	4	2	568	600
Batik Sogan	29	1	1	569	600
Batik Tambal	30	0	0	570	600

Berdasarkan data klasifikasi pada Tabel 6.5 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 6 Tabel Hasil Nilai dari Klasifikasi Data dengan *Preprocessing*

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.988333	0.866667	0.896552	0.881356
Batik Betawi	0.998333	1	0.967742	0.983607
Batik Celup	1	1	1	1
Batik Cendrawasih	0.996667	1	0.9375	0.967742
Batik Ceplok	0.986667	0.833333	0.892857	0.862069
Batik Ciamis	0.991667	0.9	0.931034	0.915254
Batik Garutan	0.993333	0.966667	0.90625	0.935484
Batik Gentongan	0.998333	0.966667	1	0.983051
Batik Kawung	0.991667	0.966667	0.878788	0.920635
Batik Keraton	0.99	0.933333	0.875	0.903226
Batik Lasem	1	1	1	1
Batik Megamendung	1	1	1	1
Batik Parang	0.995	0.933333	0.965517	0.949153
Batik Pekalongan	0.991667	0.933333	0.903226	0.918033
Batik Priangan	0.991667	0.866667	0.962963	0.912281
Batik Sekar	0.996667	0.966667	0.966667	0.966667
Batik Sidoluhur	0.996667	0.966667	0.966667	0.966667
Batik Sidomukti	0.99	0.866667	0.928571	0.896552
Batik Sogan	0.996667	0.966667	0.966667	0.966667
Batik Tambal	1	1	1	1



Gambar 6. 11 Perbandingan Waktu yang Digunakan untuk Pelatihan pada Skenario 2

Pada gambar 6.11 menunjukkan bahwa pelatihan dengan menggunakan data *preprocessing* mendapatkan waktu yang lebih cepat daripada tidak menggunakan data pelatihan. Hal ini menunjukkan bahwa dengan memberikan data *preprocessing gaussian noise* juga dapat mempercepat mesin untuk melakukan pelatihan.

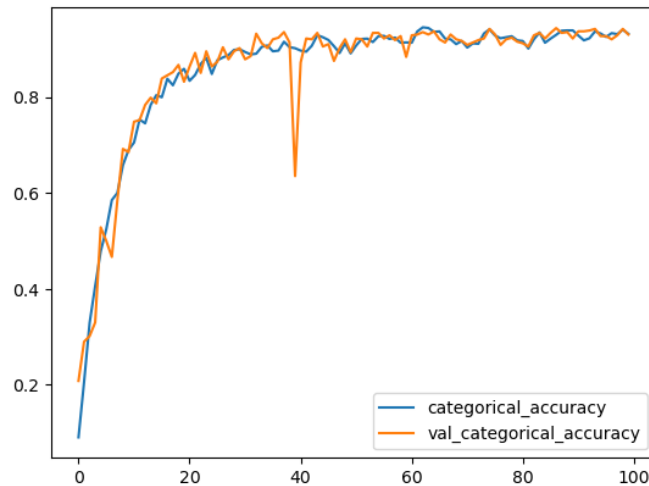
Tabel 6. 7 Tabel *classification metrics* skenario 2

Skenario	<i>Accuracy</i> pada Sistem	<i>Accuracy</i> pada perhitungan manual	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
Tanpa <i>pre-processing</i>	0.94	0.994	0.94	0.94	0.94
Dengan <i>pre-processing</i>	0.946	0.994667	0.95	0.95	0.95

Tabel 6.7 menunjukkan bahwa kedua percobaan memiliki nilai *accuracy*, *precision*, *recall* dan *f1-score* diatas 0.9, dimana ini merupakan hasil pelatihan yang baik. Namun dari kedelapan nilai dari percobaan pelatihan menggunakan data *preprocessing* tetap lebih tinggi dibandingkan dengan percobaan pelatihan tanpa menggunakan data *preprocessing*.

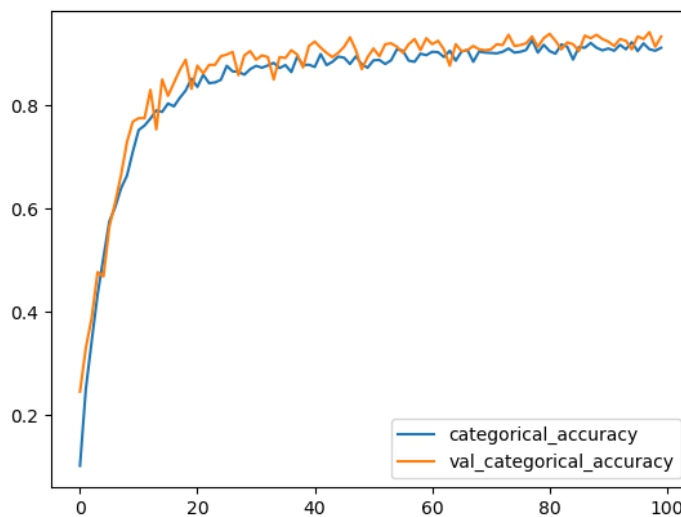
6.1.3 Hasil Uji Coba Skenario 3

Pada uji coba skenario 3 ini dilakukan dengan membandingkan membandingkan model dengan menggunakan *optimizer* Adam dan RMSprop dengan *learning rate* sebesar 0.001 dan melakukan pelatihan sebanyak 100 epoch. Berdasarkan proses *training* aka mendapatkan hasil berupa grafik *training* sebagai berikut



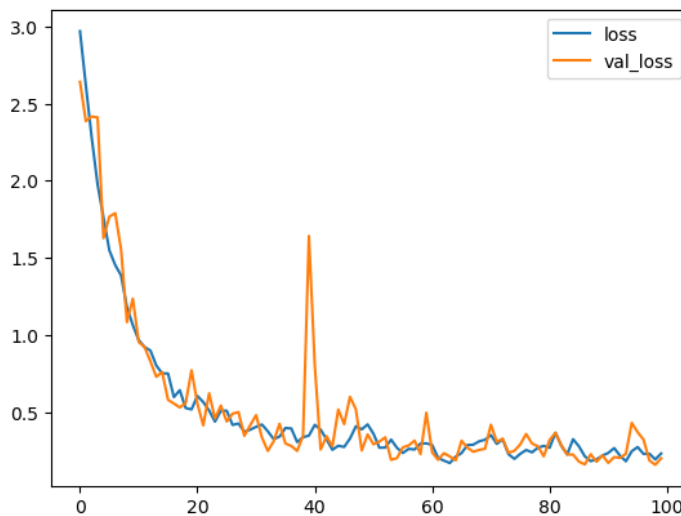
Gambar 6. 12 Hasil Grafik *Training Accuracy* menggunakan *Optimizer* Adam dengan *learning rate* 0.001

Berdasarkan grafik proses *training* pada gambar 6.12 menggunakan *optimizer* Adam dengan *learning rate* sebesar 0.001 mengalami kenaikan *accuracy* yang tidak stabil. Pada data validasi di epoch 39 menuju ke epoch 40 mengalami penurunan dari 0.91 ke 0.63, namun ketika epoch 40 menuju ke epoch 41 mengalami kenaikan dari 0.63 ke 0.87. Begitu juga dengan data *training*, namun tidak terlalu naik turun seperti data validasi.



Gambar 6. 13 Hasil Grafik *Training Accuracy* menggunakan *Optimizer* RMSprop dengan *learning rate* 0.001

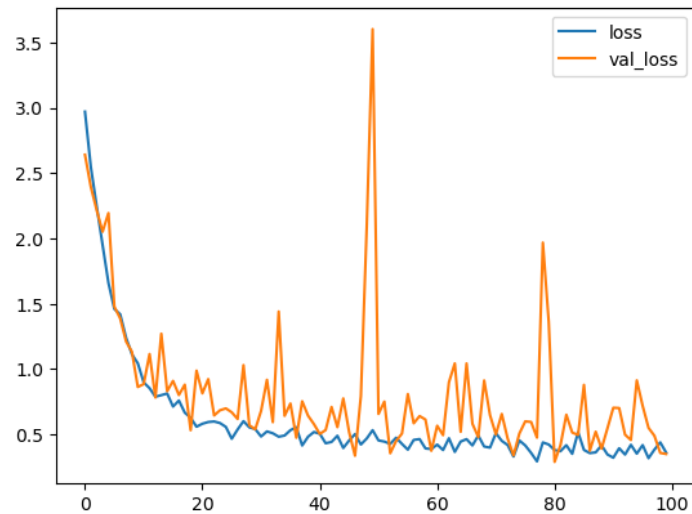
Berdasarkan grafik proses *training* pada gambar 6.13 menggunakan *optimizer* RMSprop dengan *learning rate* sebesar 0.001 juga mengalami kenaikan *accuracy* yang tidak stabil. Begitu juga dengan data *training*, namun tidak terlalu naik turun seperti data validasi.



Gambar 6. 14 Hasil Grafik *Training Loss* menggunakan *Optimizer* Adam dengan *learning rate* 0.001

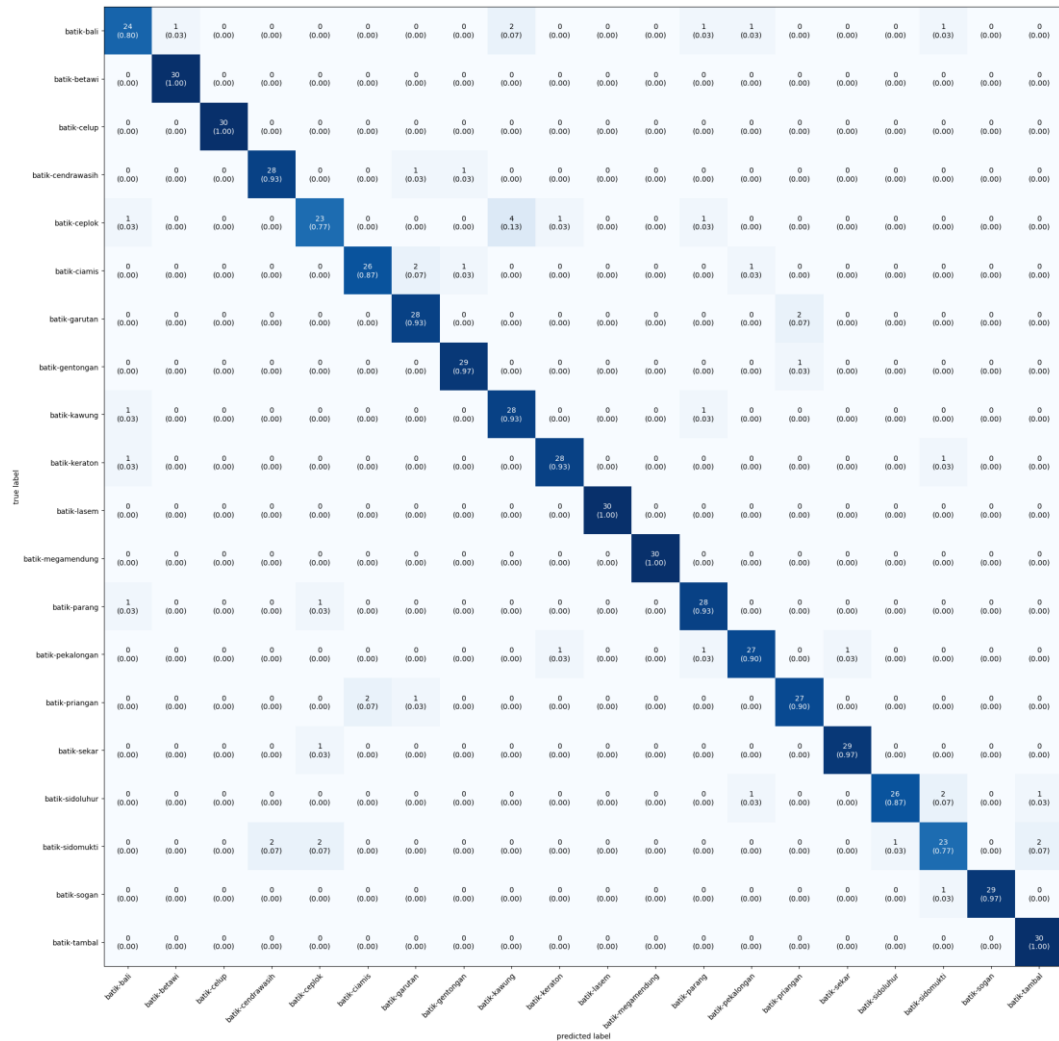
Berdasarkan grafik proses *loss* pada gambar 6.14 menggunakan *optimizer* Adam dengan *learning rate* sebesar 0.001 mengalami penurunan *loss* yang tidak stabil. Pada data validasi di epoch 39 menuju ke epoch 40 mengalami peningkatan dari 0.34 ke 1.64, namun ketika epoch 40 menuju ke epoch 41 mengalami penurunan

dari 1.64 ke 0.80. Hal ini terjadi berulang kali hingga ke epoch 100. Begitu juga dengan data *training*, namun tidak terlalu naik turun seperti data validasi.



Gambar 6. 15 Hasil Grafik *Training Loss* menggunakan *Optimizer RMSprop* dengan *learning rate* 0.001

Berdasarkan grafik proses *loss* pada gambar 6.15 menggunakan *optimizer* RMSprop dengan *learning rate* sebesar 0.001 mengalami penurunan *loss* yang tidak stabil. Pada data validasi di epoch 48 menuju ke epoch 49 mengalami peningkatan dari 0.79 ke 2.09, namun ketika epoch 49 menuju ke epoch 50 mengalami peningkatan kembali dari 2.09 ke 3.60. Dari epoch 50 menuju ke epoch 51 mengalami penurunan dari 3.60 ke 0.65. Hal ini terjadi berulang kali hingga ke epoch 50. Begitu juga dengan data *training*, namun tidak terlalu naik turun seperti data validasi.



Gambar 6. 16 *Confusion Matrix* menggunakan *Optimizer Adam* dengan *learning rate 0.001*

Berdasarkan *confusion matrix* pada gambar 6.16 dapat dilihat bahwa proses *training* menggunakan *Optimizer Adam* dengan *learning rate 0.001* berhasil mengklasifikasikan dengan benar sebanyak 553 gambar. Berdasarkan hasil tersebut didapat hasil Batik Ceplok dan Batik Sidomukti yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Ceplok dan Batik Sidomukti hanya mendapatkan 23 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan data latih terhadap hasil pengujian. Pengujian dilakukan dengan membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil *confusion matrix* ini bisa kita klasifikasikan sebagai berikut.

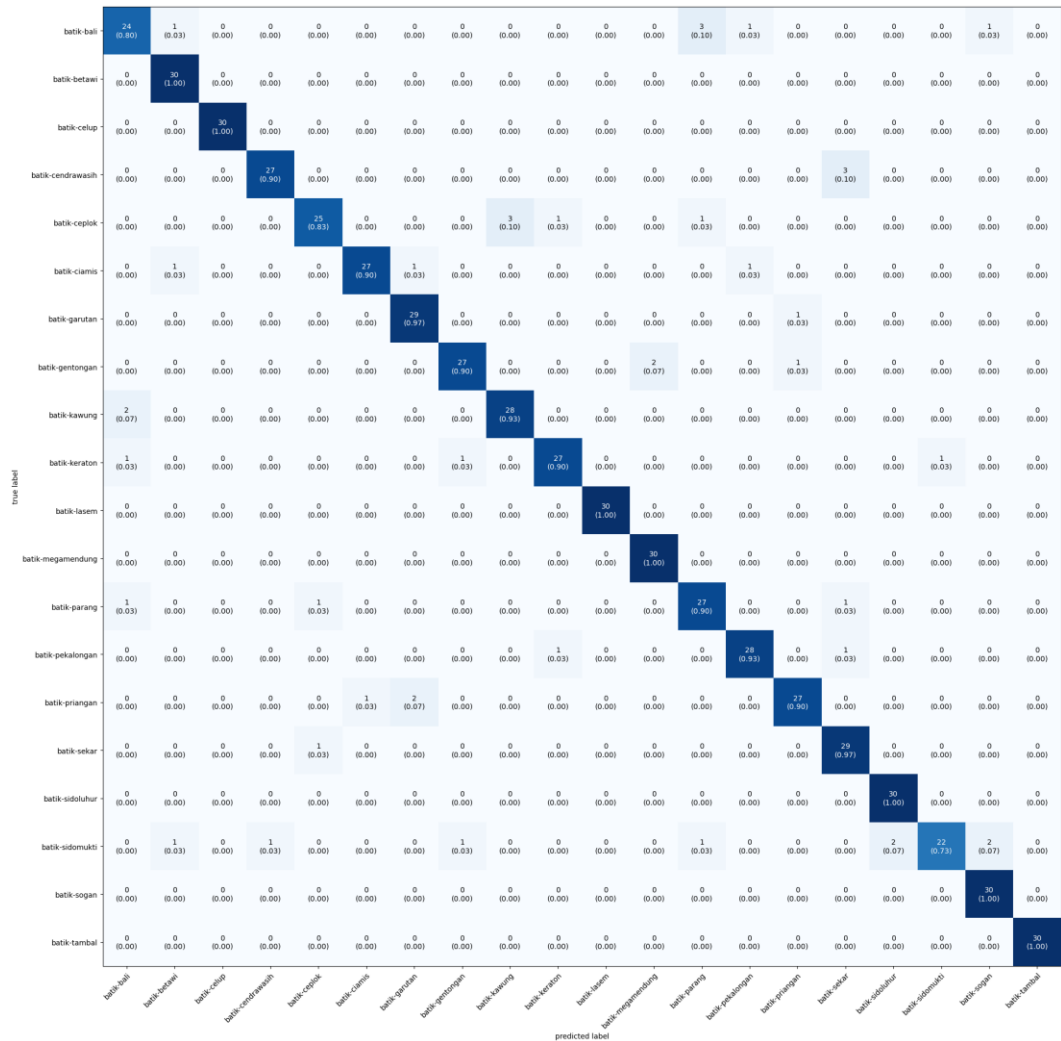
Tabel 6. 8 Tabel Hasil Nilai dari Klasifikasi Data menggunakan *Optimizer Adam* dengan *learning rate* 0.001

Kelas	Klasifikasi				Jumlah
	TP	FN	FP	TN	
Batik Bali	24	6	4	566	600
Batik Betawi	30	0	1	569	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	28	2	2	568	600
Batik Ceplok	23	7	4	566	600
Batik Ciamis	26	4	2	568	600
Batik Garutan	28	2	4	566	600
Batik Gentongan	29	1	2	568	600
Batik Kawung	28	2	6	564	600
Batik Keraton	28	2	2	568	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	0	570	600
Batik Parang	28	2	4	566	600
Batik Pekalongan	27	3	3	567	600
Batik Priangan	27	3	3	567	600
Batik Sekar	29	1	1	569	600
Batik Sidoluhur	26	4	1	569	600
Batik Sidomukti	23	7	5	565	600
Batik Sogan	29	1	0	570	600
Batik Tambal	30	0	3	567	600

Berdasarkan data klasifikasi pada Tabel 6.13 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 9 Tabel Hasil Nilai dari Klasifikasi Data menggunakan *Optimizer Adam* dengan *learning rate* 0.001

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.983333	0.8	0.857143	0.827586
Batik Betawi	0.998333	1	0.967742	0.983607
Batik Celup	1	1	1	1
Batik Cendrawasih	0.993333	0.933333	0.933333	0.933333
Batik Ceplok	0.981667	0.766667	0.851852	0.807018
Batik Ciamis	0.99	0.866667	0.928571	0.896552
Batik Garutan	0.99	0.933333	0.875	0.903226
Batik Gentongan	0.995	0.966667	0.935484	0.95082
Batik Kawung	0.986667	0.933333	0.823529	0.875
Batik Keraton	0.993333	0.933333	0.933333	0.933333
Batik Lasem	1	1	1	1
Batik Megamendung	1	1	1	1
Batik Parang	0.99	0.933333	0.875	0.903226
Batik Pekalongan	0.99	0.9	0.9	0.9
Batik Priangan	0.99	0.9	0.9	0.9
Batik Sekar	0.996667	0.966667	0.966667	0.966667
Batik Sidoluhur	0.991667	0.866667	0.962963	0.912281
Batik Sidomukti	0.98	0.766667	0.821429	0.793103
Batik Sogan	0.998333	0.966667	1	0.983051
Batik Tambal	0.995	1	0.909091	0.952381



Gambar 6. 17 *Confusion Matrix* menggunakan *Optimizer RMSprop* dengan *learning rate* 0.001

Berdasarkan *confusion matrix* pada gambar 6.17 dapat dilihat bahwa proses *training* menggunakan *Optimizer RMSprop* dengan *learning rate* 0.001 berhasil mengklasifikasikan dengan benar sebanyak 557 gambar. Berdasarkan hasil tersebut didapat hasil Batik Sidomukti yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Kedua batik tersebut hanya mendapatkan 22 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan data latih terhadap hasil pengujian. Pengujian dilakukan dengan membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil *confusion matrix* ini bisa kita klasifikasikan sebagai berikut.

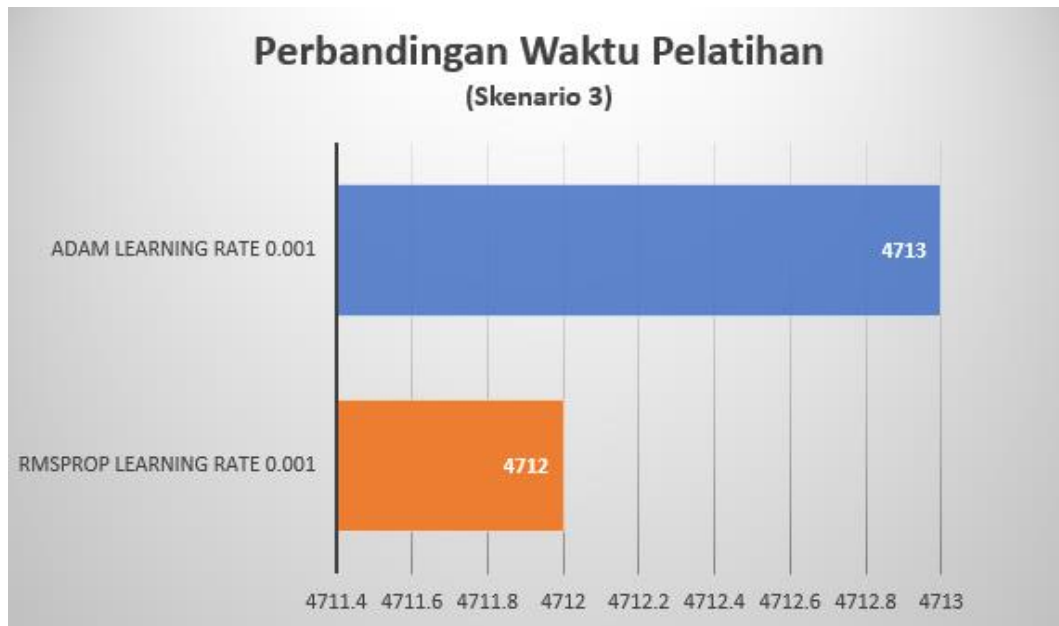
Tabel 6. 10 Tabel Hasil Nilai dari Klasifikasi Data menggunakan *Optimizer* RMSprop dengan *learning rate* 0.001

Kelas	Klasifikasi				Jumlah
	TP	FN	FP	TN	
Batik Bali	24	6	4	566	600
Batik Betawi	30	0	3	567	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	27	3	1	569	600
Batik Ceplok	25	5	2	568	600
Batik Ciamis	27	3	1	569	600
Batik Garutan	29	1	3	567	600
Batik Gentongan	27	3	2	568	600
Batik Kawung	28	2	3	567	600
Batik Keraton	27	3	2	568	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	2	568	600
Batik Parang	27	3	5	565	600
Batik Pekalongan	28	2	2	568	600
Batik Priangan	27	3	2	568	600
Batik Sekar	29	1	5	565	600
Batik Sidoluhur	30	0	2	568	600
Batik Sidomukti	22	8	1	569	600
Batik Sogan	30	0	3	567	600
Batik Tambal	30	0	0	570	600

Berdasarkan data klasifikasi pada Tabel 6.15 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut.

Tabel 6. 11 Tabel Hasil Nilai dari Klasifikasi Data menggunakan *Optimizer* RMSprop dengan *learning rate* 0.001

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.983333	0.8	0.857143	0.827586
Batik Betawi	0.995	1	0.909091	0.952381
Batik Celup	1	1	1	1
Batik Cendrawasih	0.993333	0.9	0.964286	0.931034
Batik Ceplok	0.988333	0.833333	0.925926	0.877193
Batik Ciamis	0.993333	0.9	0.964286	0.931034
Batik Garutan	0.993333	0.966667	0.90625	0.935484
Batik Gentongan	0.991667	0.9	0.931034	0.915254
Batik Kawung	0.991667	0.933333	0.903226	0.918033
Batik Keraton	0.991667	0.9	0.931034	0.915254
Batik Lasem	1	1	1	1
Batik Megamendung	0.996667	1	0.9375	0.967742
Batik Parang	0.986667	0.9	0.84375	0.870968
Batik Pekalongan	0.993333	0.933333	0.933333	0.933333
Batik Priangan	0.991667	0.9	0.931034	0.915254
Batik Sekar	0.99	0.966667	0.852941	0.90625
Batik Sidoluhur	0.996667	1	0.9375	0.967742
Batik Sidomukti	0.985	0.733333	0.956522	0.830189
Batik Sogan	0.995	1	0.909091	0.952381
Batik Tambal	1	1	1	1



Gambar 6. 18 Perbandingan Waktu yang Digunakan untuk Pelatihan pada Skenario 4

Pada gambar 6.18 menunjukkan bahwa pelatihan menggunakan *optimizer* RMSprop dengan *learning rate* 0.001 mendapatkan waktu yang lebih cepat daripada menggunakan *optimizer* Adam dengan *learning rate* 0.001.

Tabel 6. 12 Tabel classification metrics skenario 3

Skenario	<i>Accuracy</i> pada Sistem	<i>Accuracy</i> pada perhitungan manual	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
RMSprop (0.001)	0.9283	0.992833	0.93	0.93	0.93
Adam (0.001)	0.921	0.992167	0.92	0.92	0.92

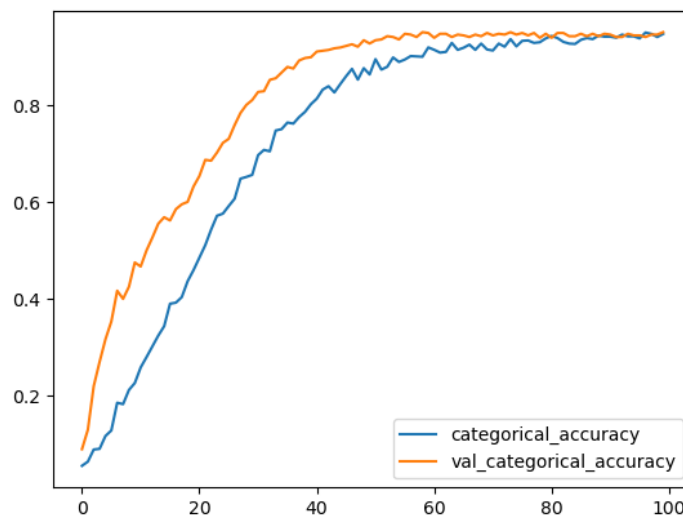
Berdasarkan uji coba membandingkan model antara menggunakan *optimizer* RMSprop dengan *learning rate* 0.001 dan menggunakan *optimizer* Adam dengan *learning rate* 0.001 menunjukkan bahwa kedua percobaan menggunakan *optimizer* RMSprop dengan *learning rate* 0.001 memiliki nilai *accuracy* lebih tinggi daripada menggunakan *optimizer* Adam dengan *learning rate* 0.001. Sedangkan pada nilai *precision* dari kedua uji coba telah menunjukkan angka diatas 0.9, dimana hal ini

menandakan bahwa model pelatihan sudah sangat baik dalam mengenali setiap kelas yang ada pada pelatihan meskipun terdapat perbedaan nilai antara *optimizer* RMSprop dengan *optimizer* Adam. Untuk nilai *recall* dan *f1-score* diatas 0.9, dimana ini merupakan hasil pelatihan yang baik. Namun, berdasarkan grafik *accuracy* dan grafik *loss* bisa disimpulkan bahwa pelatihan menggunakan *optimizer* RMSprop dengan *optimizer* Adam dengan *learning rate* sebesar 0.001 mengalami *overfitting*. Perbandingan ini bisa dilihat pada tabel 6.12

6.1.4 Hasil Uji Coba Skenario 4

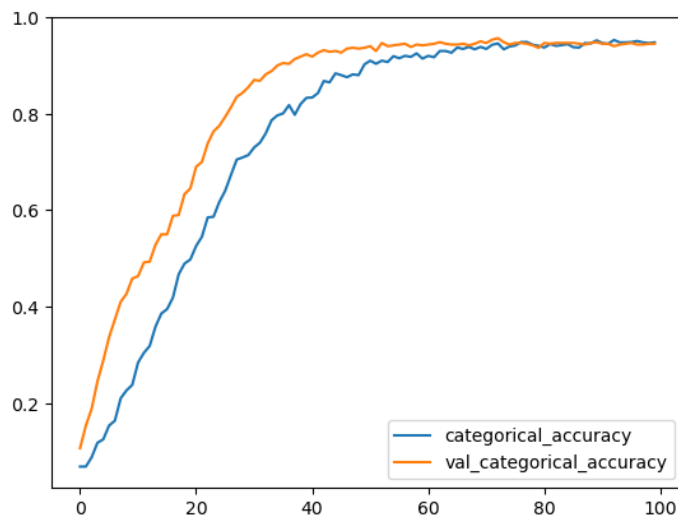
Pada uji coba skenario 4 ini dilakukan dengan membandingkan membandingkan model dengan menggunakan *optimizer* Adam dan RMSprop dengan *learning rate* sebesar 0.0001 dan melakukan pelatihan sebanyak 100 epoch. Berikut merupakan hasil uji coba skenario 4.

6.1.4.1 Grafik *Training Accuracy*



Gambar 6. 19 Hasil Grafik *Training Accuracy* menggunakan *Optimizer* RMSprop dengan *learning rate* 0.0001

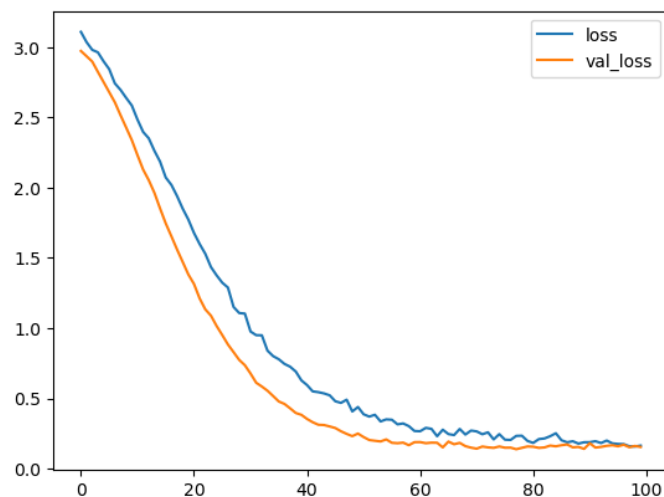
Berdasarkan grafik proses *training* pada gambar 6.19 menggunakan *optimizer* RMSprop dengan *learning rate* sebesar 0.0001 mengalami kenaikan *accuracy* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di atas 80% yang berakhir pada epoch ke 100.



Gambar 6. 20 Hasil Grafik *Training Accuracy* menggunakan *Optimizer Adam* dengan *learning rate* 0.0001

Berdasarkan grafik proses *training* pada gambar 6.20 menggunakan *optimizer Adam* dengan *learning rate* sebesar 0.0001 mengalami kenaikan *accuracy* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di atas 80% yang berakhir pada epoch ke 100.

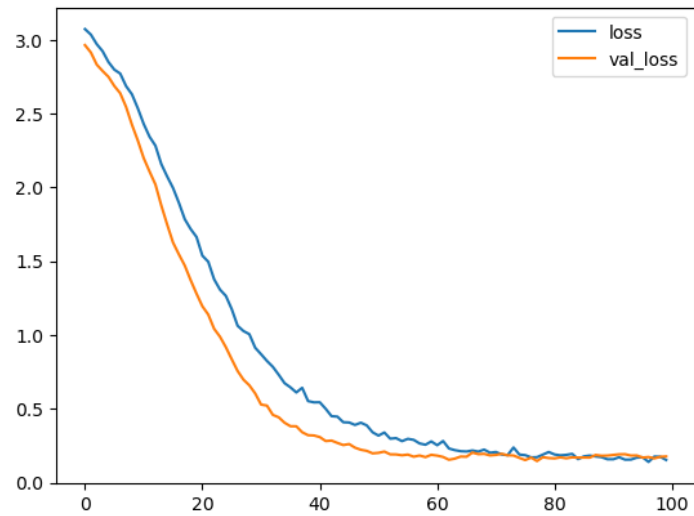
6.1.4.2 Grafik *Training Loss*



Gambar 6. 21 Hasil Grafik *Training Loss* menggunakan *Optimizer RMSprop* dengan *learning rate* 0.0001

Berdasarkan grafik proses *training* pada gambar 6.13 menggunakan *Optimizer RMSprop* dengan *learning rate* 0.0001 mengalami penurunan *loss* pada

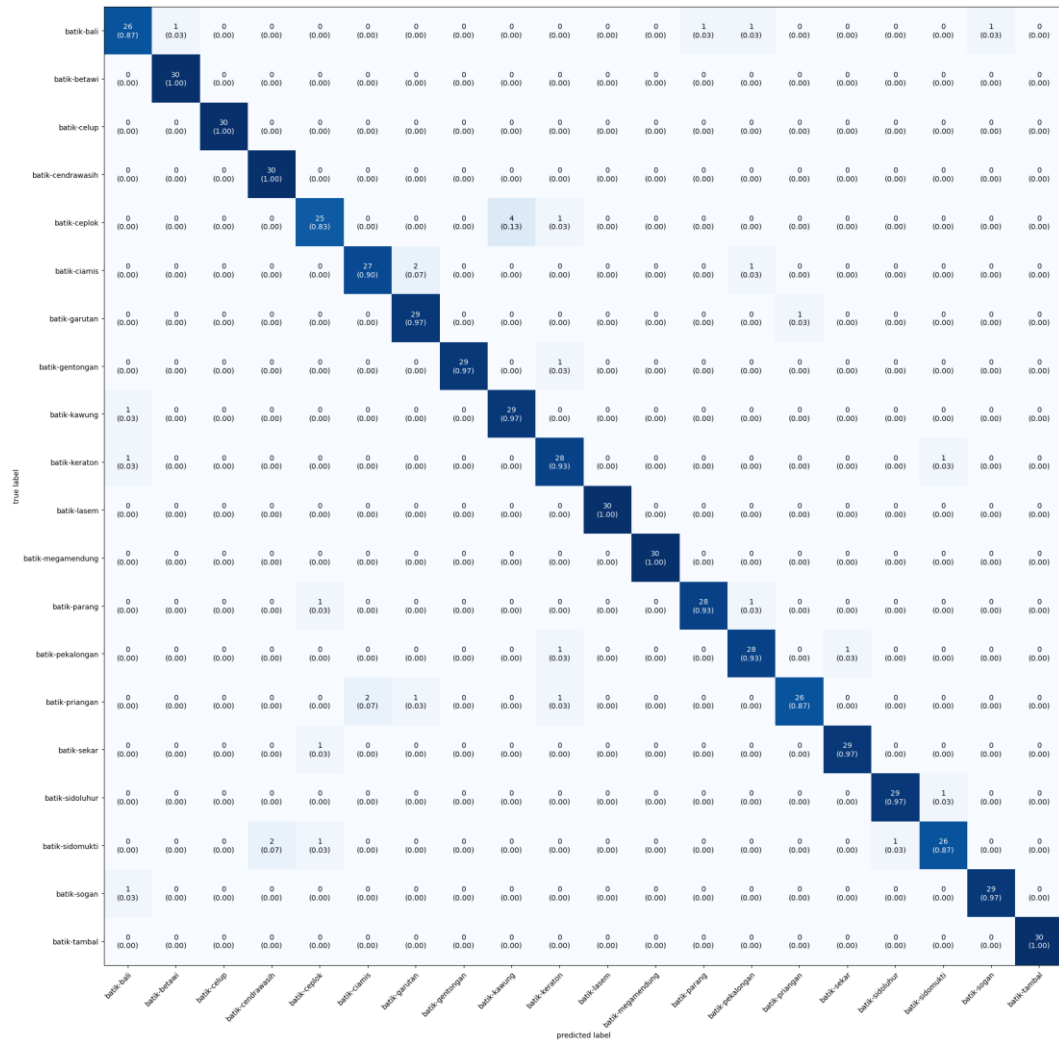
setiap epoch-nya. Hasil akhir pada proses *training* ini berada di angka mendekati 0 yang berakhir pada epoch ke 100.



Gambar 6. 22 Hasil Grafik *Training Loss* menggunakan *Optimizer Adam* dengan learning rate 0.0001

Berdasarkan grafik proses *training* pada gambar 6.22 menggunakan *Optimizer RMSprop* dengan *learning rate* 0.0001 mengalami penurunan *loss* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di angka mendekati 0 yang berakhir pada epoch ke 100.

6.1.4.3 Confusion Matrix



Gambar 6. 23 Confusion Matrix menggunakan Optimizer RMSprop dengan learning rate 0.0001

Berdasarkan confusion matrix pada gambar 6.23 dapat dilihat bahwa proses training menggunakan Optimizer RMSprop dengan learning rate 0.0001 berhasil mengklasifikasikan dengan benar sebanyak 568 gambar. Berdasarkan hasil tersebut didapat hasil Batik Ceplok yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Ceplok hanya mendapatkan 24 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan data latih terhadap hasil pengujian. Pengujian dilakukan dengan

membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil *confusion matrix* ini bisa kita klasifikasikan sebagai berikut.

Tabel 6. 13 Tabel Hasil Nilai dari Klasifikasi Data menggunakan *Optimizer* RMSprop dengan *learning rate* 0.0001

Kelas	Klasifikasi				Jumlah
	TP	FP	FN	TN	
Batik Bali	26	4	3	567	600
Batik Betawi	30	0	1	569	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	30	0	2	568	600
Batik Ceplok	25	5	3	567	600
Batik Ciamis	27	3	2	568	600
Batik Garutan	29	1	3	567	600
Batik Gentongan	29	1	0	570	600
Batik Kawung	29	1	4	566	600
Batik Keraton	28	2	4	566	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	0	570	600
Batik Parang	28	2	1	569	600
Batik Pekalongan	28	2	3	567	600
Batik Priangan	26	4	1	569	600
Batik Sekar	29	1	1	569	600
Batik Sidoluhur	29	1	1	569	600
Batik Sidomukti	26	4	2	568	600

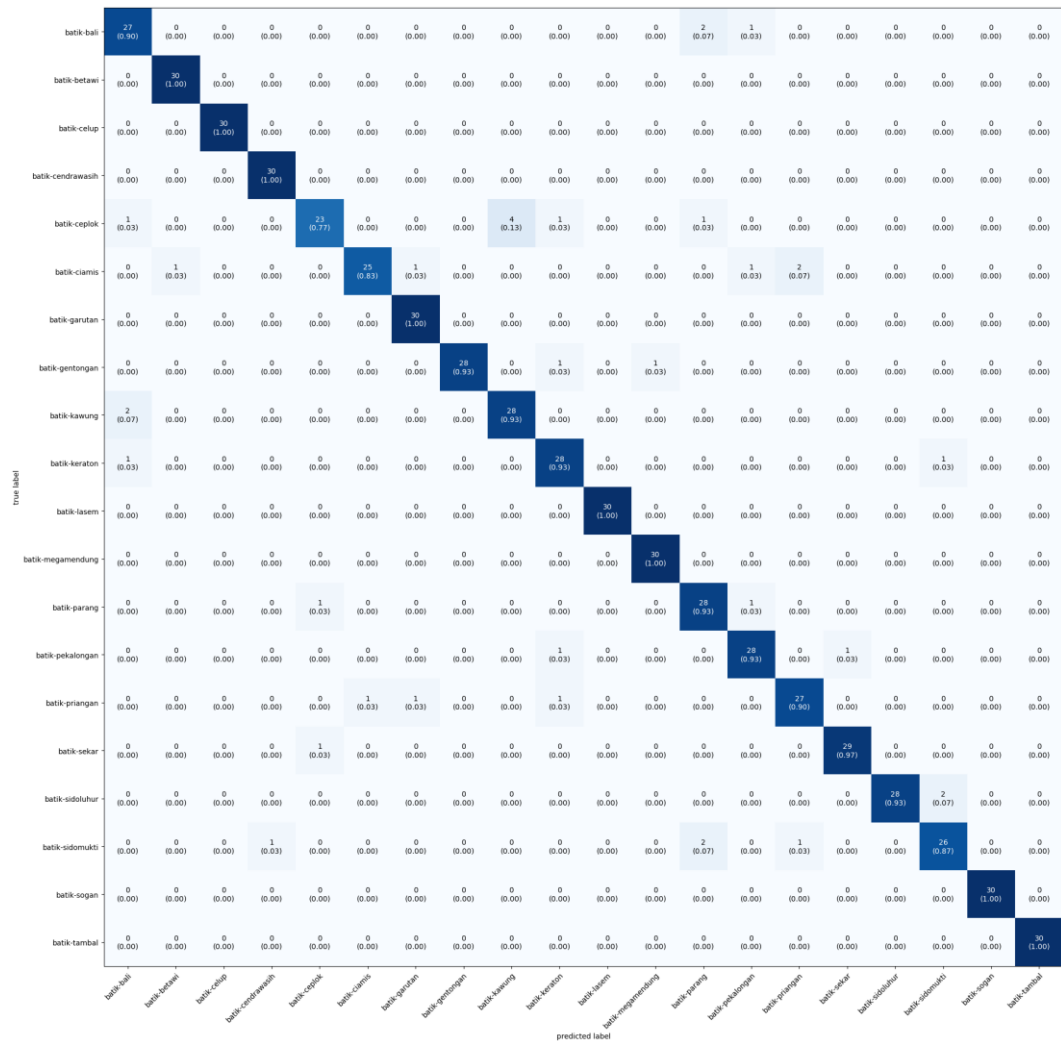
Batik Sogan	29	1	1	569	600
Batik Tambal	30	0	0	570	600

Berdasarkan data klasifikasi pada Tabel 6.8 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 14 Tabel Hasil Nilai dari Klasifikasi Data menggunakan *Optimizer* RMSprop dengan *learning rate* 0.0001

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.988333	0.866667	0.896552	0.881356
Batik Betawi	0.998333	1	0.967742	0.983607
Batik Celup	1	1	1	1
Batik Cendrawasih	0.996667	1	0.9375	0.967742
Batik Ceplok	0.986667	0.833333	0.892857	0.862069
Batik Ciamis	0.991667	0.9	0.931034	0.915254
Batik Garutan	0.993333	0.966667	0.90625	0.935484
Batik Gentongan	0.998333	0.966667	1	0.983051
Batik Kawung	0.991667	0.966667	0.878788	0.920635
Batik Keraton	0.99	0.933333	0.875	0.903226
Batik Lasem	1	1	1	1
Batik Megamendung	1	1	1	1
Batik Parang	0.995	0.933333	0.965517	0.949153
Batik Pekalongan	0.991667	0.933333	0.903226	0.918033
Batik Priangan	0.991667	0.866667	0.962963	0.912281
Batik Sekar	0.996667	0.966667	0.966667	0.966667
Batik Sidoluhur	0.996667	0.966667	0.966667	0.966667

Batik Sidomukti	0.99	0.866667	0.928571	0.896552
Batik Sogan	0.996667	0.966667	0.966667	0.966667
Batik Tambal	1	1	1	1



Gambar 6. 24 Confusion Matrix menggunakan Optimizer Adam dengan learning rate 0.0001

Berdasarkan confusion matrix pada gambar 6.24 dapat dilihat bahwa proses training menggunakan Optimizer Adam dengan learning rate 0.0001 berhasil mengklasifikasikan dengan benar sebanyak 565 gambar. Berdasarkan hasil tersebut didapat hasil Batik Ceplok yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Ceplok hanya mendapatkan 23 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan data latih terhadap hasil pengujian. Pengujian dilakukan dengan membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil *confusion matrix* ini bisa kita klasifikasikan sebagai berikut.

Tabel 6. 15 Tabel Hasil Nilai dari Klasifikasi Data menggunakan *Optimizer Adam* dengan *learning rate* 0.0001

Kelas	Klasifikasi				Jumlah
	TP	FN	FP	TN	
Batik Bali	27	3	4	566	600
Batik Betawi	30	0	1	569	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	30	0	1	569	600
Batik Ceplok	23	7	2	568	600
Batik Ciamis	25	5	1	569	600
Batik Garutan	30	0	2	568	600
Batik Gentongan	28	2	0	570	600
Batik Kawung	28	2	4	566	600
Batik Keraton	28	2	4	566	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	1	569	600
Batik Parang	28	2	5	565	600
Batik Pekalongan	28	2	3	567	600
Batik Priangan	27	3	3	567	600
Batik Sekar	29	1	1	569	600
Batik Sidoluhur	28	2	0	570	600

Batik Sidomukti	26	4	3	567	600
Batik Sogan	30	0	0	570	600
Batik Tambal	30	0	0	570	600

Berdasarkan data klasifikasi pada Tabel 6.10 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 16 Tabel Hasil Nilai dari Klasifikasi Data menggunakan *Optimizer Adam* dengan *learning rate* 0.0001

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.988333	0.9	0.870968	0.885246
Batik Betawi	0.998333	1	0.967742	0.983607
Batik Celup	1	1	1	1
Batik Cendrawasih	0.998333	1	0.967742	0.983607
Batik Ceplok	0.985	0.766667	0.92	0.836364
Batik Ciamis	0.99	0.833333	0.961538	0.892857
Batik Garutan	0.996667	1	0.9375	0.967742
Batik Gentongan	0.996667	0.933333	1	0.965517
Batik Kawung	0.99	0.933333	0.875	0.903226
Batik Keraton	0.99	0.933333	0.875	0.903226
Batik Lasem	1	1	1	1
Batik Megamendung	0.998333	1	0.967742	0.983607
Batik Parang	0.988333	0.933333	0.848485	0.888889
Batik Pekalongan	0.991667	0.933333	0.903226	0.918033
Batik Priangan	0.99	0.9	0.9	0.9
Batik Sekar	0.996667	0.966667	0.966667	0.966667

Batik Sidoluhur	0.996667	0.933333	1	0.965517
Batik Sidomukti	0.988333	0.866667	0.896552	0.881356
Batik Sogan	1	1	1	1
Batik Tambal	1	1	1	1



Gambar 6. 25 Perbandingan Waktu yang Digunakan untuk Pelatihan pada Skenario 4

Pada gambar 6.25 menunjukkan bahwa pelatihan menggunakan *optimizer* RMSprop dengan *learning rate* 0.0001 mendapatkan waktu yang lebih cepat daripada menggunakan *optimizer* Adam dengan *learning rate* 0.001.

Tabel 6. 17 Tabel classification metrics skenario 4

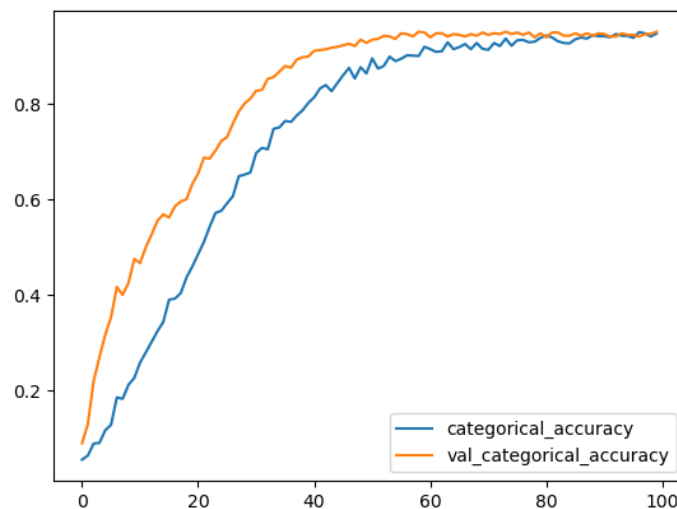
Skenario	<i>Accuracy</i> pada Sistem	<i>Accuracy</i> pada perhitungan manual	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
RMSprop (0.0001)	0.946	0.994667	0.95	0.95	0.95

Adam (0.0001)	0.941	0.994167	0.94	0.94	0.94
------------------	-------	----------	------	------	------

Berdasarkan uji coba membandingkan model antara menggunakan *optimizer* RMSprop dengan *learning rate* 0.0001 dan menggunakan *optimizer* Adam dengan *learning rate* 0.0001 menunjukkan bahwa kedua percobaan menggunakan *optimizer* RMSprop dengan *learning rate* 0.0001 memiliki nilai *accuracy* lebih tinggi daripada menggunakan *optimizer* Adam dengan *learning rate* 0.0001. Sedangkan pada nilai *precision* dari kedua uji coba telah menunjukkan angka diatas 0.9, dimana hal ini menandakan bahwa model pelatihan sudah sangat baik dalam mengenali setiap kelas yang ada pada pelatihan meskipun terdapat perbedaan nilai antara *optimizer* RMSprop dengan *optimizer* Adam. Untuk nilai *recall* dan *f1-score* diatas 0.9, dimana ini merupakan hasil pelatihan yang baik. Perbandingan ini bisa dilihat pada tabel 6.12

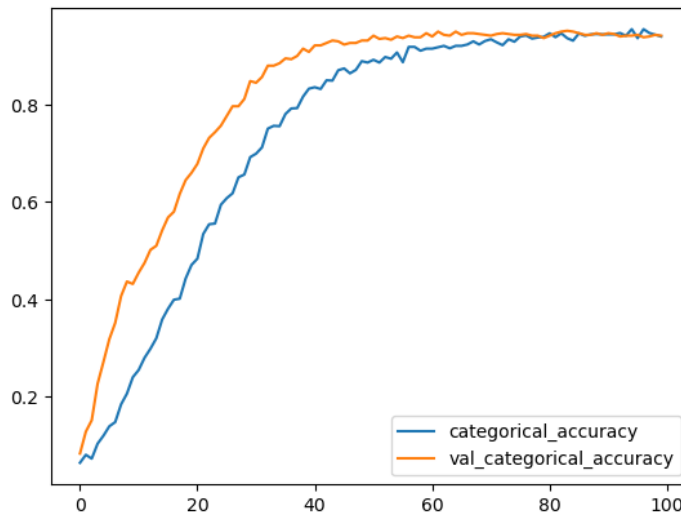
6.1.5 Hasil Uji Coba Skenario 5

Pada uji coba skenario 5 ini dilakukan dengan membandingkan model menggunakan *EfficientNetB0* hingga *EfficientNetB5* dengan *optimizer* RMSprop dengan *learning rate* sebesar 0.0001 dan melakukan pelatihan sebanyak 100 epoch. Berdasarkan proses *training* aka mendapatkan hasil berupa grafik *training* sebagai berikut



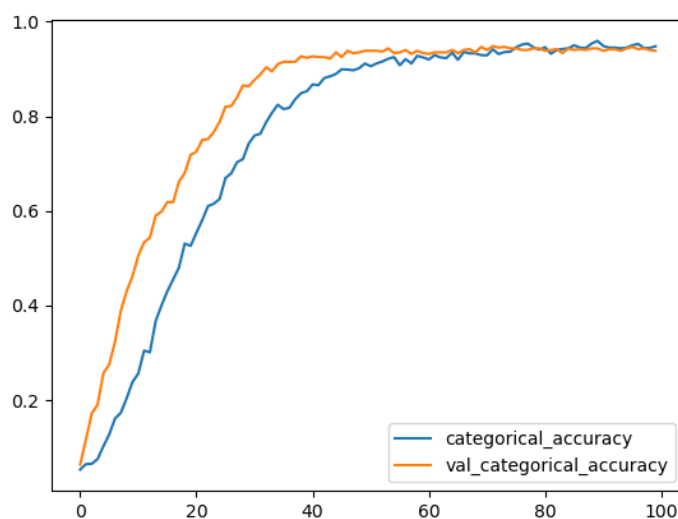
Gambar 6. 26 Hasil Grafik Training Accuracy menggunakan *EfficientNetB0*

Berdasarkan grafik proses *training* pada gambar 6.26 menggunakan *EfficientNetB0* mengalami kenaikan *accuracy* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di atas 80% yang berakhir pada epoch ke 100.



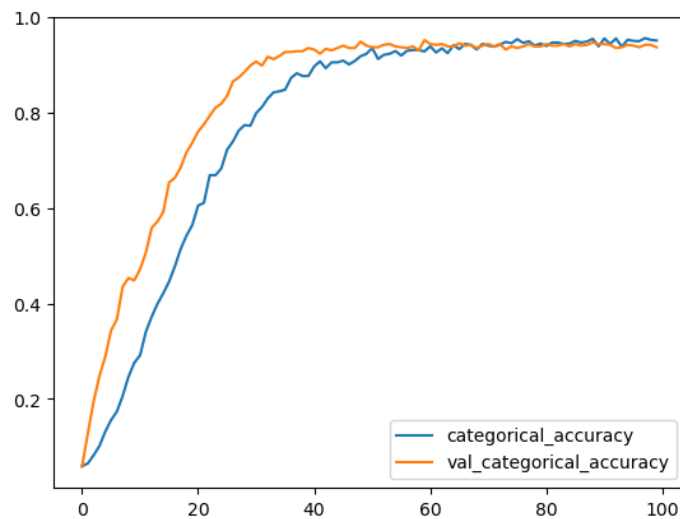
Gambar 6. 27 Hasil Grafik *Training Accuracy* menggunakan *EfficientNetB1*

Berdasarkan grafik proses *training* pada gambar 6.27 menggunakan *EfficientNetB1* mengalami kenaikan *accuracy* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di atas 80% yang berakhir pada epoch ke 100.



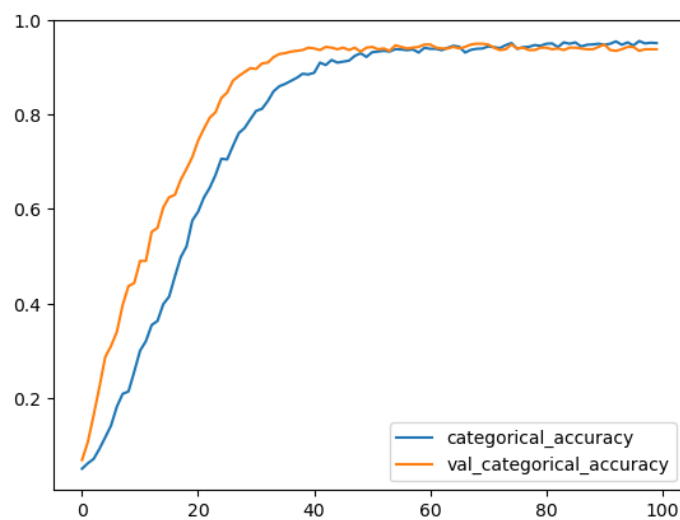
Gambar 6. 28 Hasil Grafik *Training Accuracy* menggunakan *EfficientNetB2*

Berdasarkan grafik proses *training* pada gambar 6.28 menggunakan *EfficientNetB2* mengalami kenaikan *accuracy* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di atas 80% yang berakhir pada epoch ke 100.



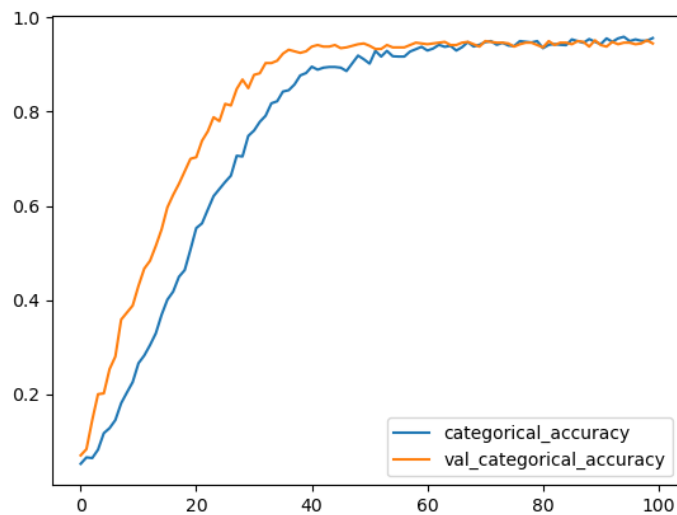
Gambar 6. 29 Hasil Grafik *Training Accuracy* menggunakan *EfficientNetB3*

Berdasarkan grafik proses *training* pada gambar 6.29 menggunakan *EfficientNetB3* mengalami kenaikan *accuracy* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di atas 80% yang berakhir pada epoch ke 100.



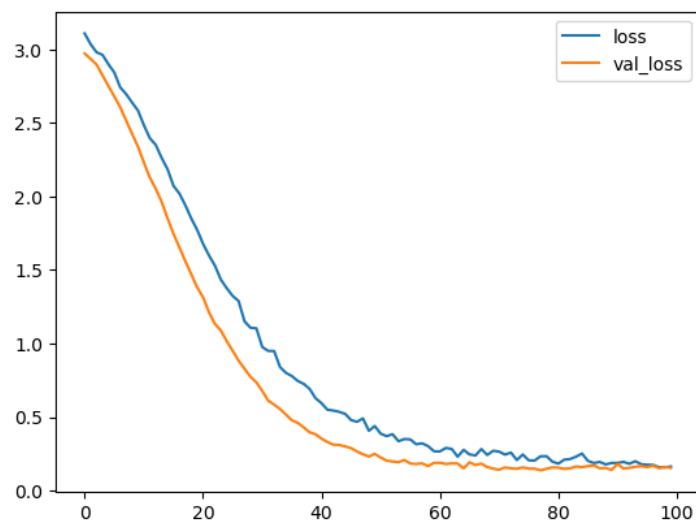
Gambar 6. 30 Hasil Grafik *Training Accuracy* menggunakan *EfficientNetB4*

Berdasarkan grafik proses *training* pada gambar 6.30 menggunakan *EfficientNetB4* mengalami kenaikan *accuracy* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di atas 80% yang berakhir pada epoch ke 100.



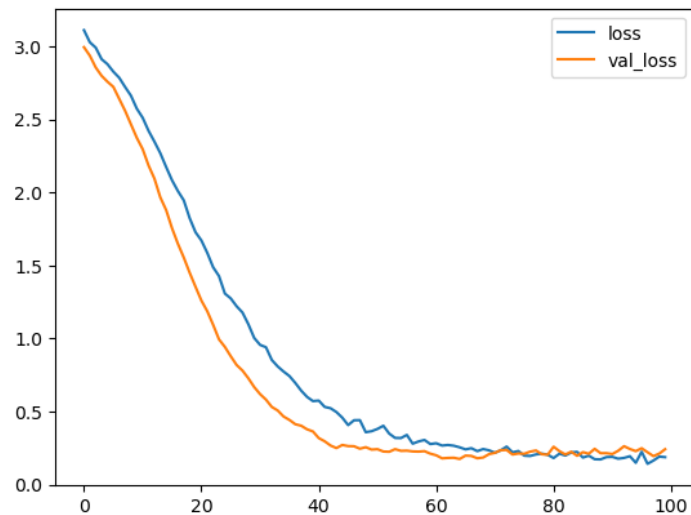
Gambar 6. 31 Hasil Grafik Training Accuracy menggunakan *EfficientNetB5*

Berdasarkan grafik proses *training* pada gambar 6.31 menggunakan *EfficientNetB5* mengalami kenaikan *accuracy* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di atas 80% yang berakhir pada epoch ke 100.



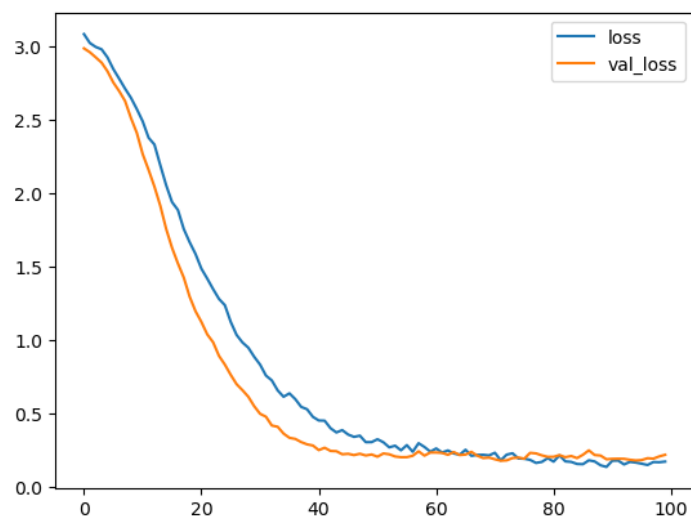
Gambar 6. 32 Hasil Grafik Training Loss menggunakan *EfficientNetB0*

Berdasarkan grafik proses *training* pada gambar 6.32 menggunakan *EfficientNetB0* mengalami penurunan *loss* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di angka mendekati 0 yang berakhir pada epoch ke 100.



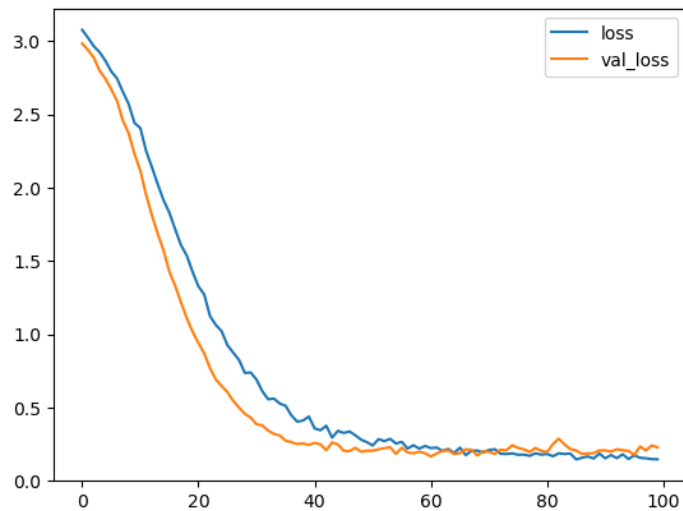
Gambar 6. 33 Hasil Grafik *Training Loss* menggunakan *EfficientNetB1*

Berdasarkan grafik proses *training* pada gambar 6.33 menggunakan *EfficientNetB1* mengalami penurunan *loss* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di angka mendekati 0 yang berakhir pada epoch ke 50



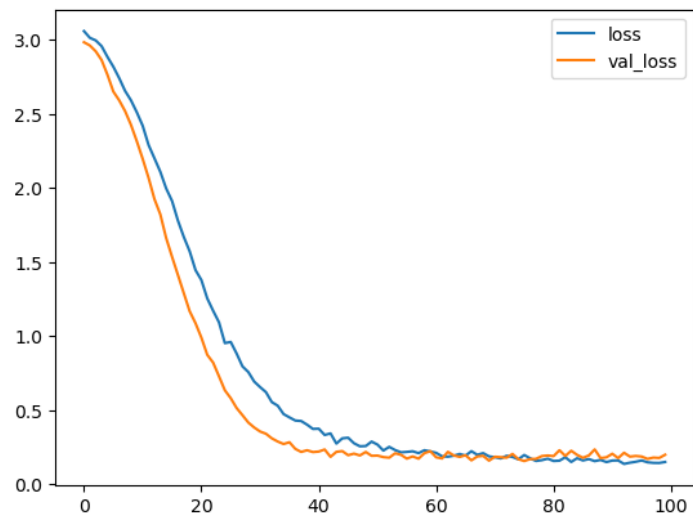
Gambar 6. 34 Hasil Grafik *Training Loss* menggunakan *EfficientNetB2*

Berdasarkan grafik proses *training* pada gambar 6.34 menggunakan *EfficientNetB2* mengalami penurunan *loss* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di angka mendekati 0 yang berakhir pada epoch ke 100.



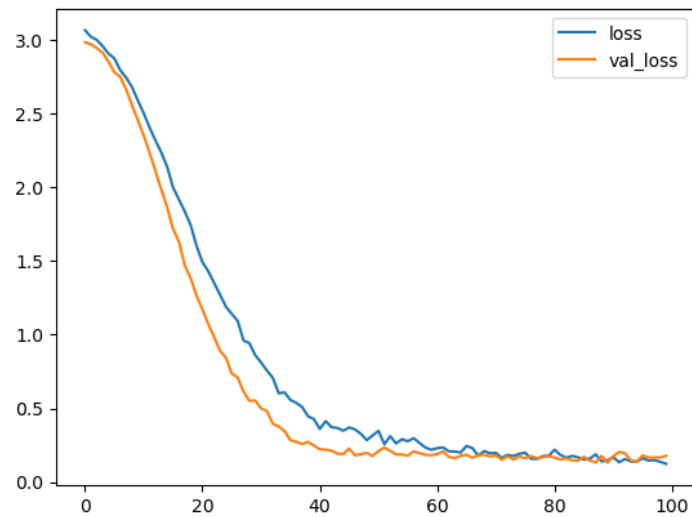
Gambar 6. 35 Hasil Grafik *Training Loss* menggunakan *EfficientNetB3*

Berdasarkan grafik proses *training* pada gambar 6.35 menggunakan *EfficientNetB3* mengalami penurunan *loss* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di angka mendekati 0 yang berakhir pada epoch ke 50



Gambar 6. 36 Hasil Grafik *Training Loss* menggunakan *EfficientNetB4*

Berdasarkan grafik proses *training* pada gambar 6.36 menggunakan *EfficientNetB4* mengalami penurunan *loss* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di angka mendekati 0 yang berakhir pada epoch ke 100.



Gambar 6. 37 Hasil Grafik *Training Loss* menggunakan *EfficientNetB5*

Berdasarkan grafik proses *training* pada gambar 6.37 menggunakan *EfficientNetB5* mengalami penurunan *loss* pada setiap epoch-nya. Hasil akhir pada proses *training* ini berada di angka mendekati 0 yang berakhir pada epoch ke 100.

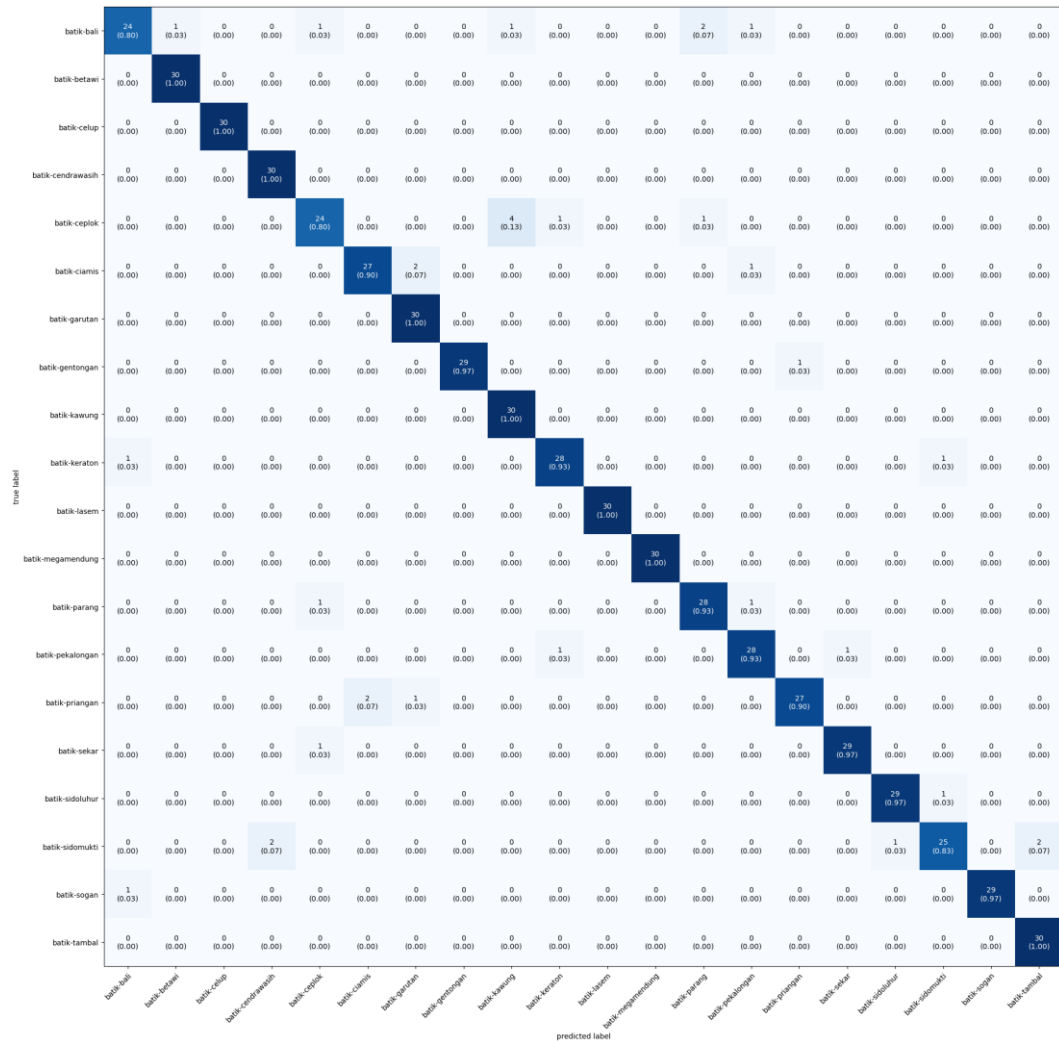
Tabel 6. 18 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB0

Kelas	Klasifikasi				Jumlah
	TP	FP	FN	TN	
Batik Bali	26	4	3	567	600
Batik Betawi	30	0	1	569	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	30	0	2	568	600
Batik Ceplok	25	5	3	567	600
Batik Ciamis	27	3	2	568	600
Batik Garutan	29	1	3	567	600
Batik Gentongan	29	1	0	570	600
Batik Kawung	29	1	4	566	600
Batik Keraton	28	2	4	566	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	0	570	600
Batik Parang	28	2	1	569	600
Batik Pekalongan	28	2	3	567	600
Batik Priangan	26	4	1	569	600
Batik Sekar	29	1	1	569	600
Batik Sidoluhur	29	1	1	569	600
Batik Sidomukti	26	4	2	568	600
Batik Sogan	29	1	1	569	600
Batik Tambal	30	0	0	570	600

Berdasarkan data klasifikasi pada Tabel 6.19 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 19 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB0

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.988333	0.866667	0.896552	0.881356
Batik Betawi	0.998333	1	0.967742	0.983607
Batik Celup	1	1	1	1
Batik Cendrawasih	0.996667	1	0.9375	0.967742
Batik Ceplok	0.986667	0.833333	0.892857	0.862069
Batik Ciamis	0.991667	0.9	0.931034	0.915254
Batik Garutan	0.993333	0.966667	0.90625	0.935484
Batik Gentongan	0.998333	0.966667	1	0.983051
Batik Kawung	0.991667	0.966667	0.878788	0.920635
Batik Keraton	0.99	0.933333	0.875	0.903226
Batik Lasem	1	1	1	1
Batik Megamendung	1	1	1	1
Batik Parang	0.995	0.933333	0.965517	0.949153
Batik Pekalongan	0.991667	0.933333	0.903226	0.918033
Batik Priangan	0.991667	0.866667	0.962963	0.912281
Batik Sekar	0.996667	0.966667	0.966667	0.966667
Batik Sidoluhur	0.996667	0.966667	0.966667	0.966667
Batik Sidomukti	0.99	0.866667	0.928571	0.896552
Batik Sogan	0.996667	0.966667	0.966667	0.966667
Batik Tambal	1	1	1	1



Gambar 6. 39 *Confusion Matrix* menggunakan *EfficientNetB1*

Berdasarkan *confusion matrix* pada gambar 6.36 dapat dilihat bahwa proses *training* menggunakan *EfficientNetB1* berhasil mengklasifikasikan dengan benar sebanyak 567 gambar. Berdasarkan hasil tersebut didapat hasil Batik Bali dan Batik Ceplok yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Bali dan Batik Ceplok hanya mendapatkan 24 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan data latih terhadap hasil pengujian. Pengujian dilakukan dengan membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil *confusion matrix* ini bisa kita klasifikasikan sebagai berikut.

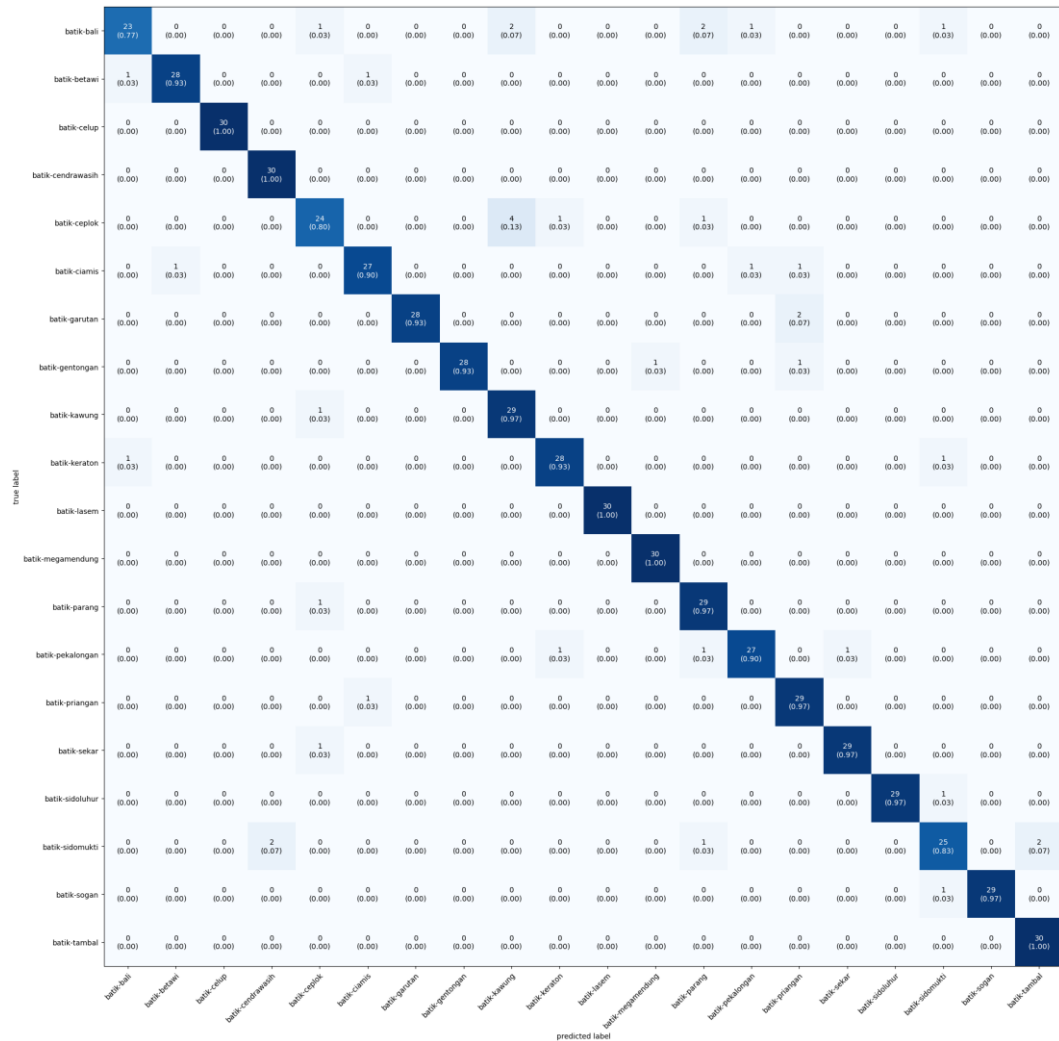
Tabel 6. 20 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB1

Kelas	Klasifikasi				Jumlah
	TP	FN	FP	TN	
Batik Bali	24	6	2	568	600
Batik Betawi	30	0	1	569	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	30	0	2	568	600
Batik Ceplok	24	6	3	567	600
Batik Ciamis	27	3	2	568	600
Batik Garutan	30	0	3	567	600
Batik Gentongan	29	1	0	570	600
Batik Kawung	30	0	5	565	600
Batik Keraton	28	2	2	568	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	0	570	600
Batik Parang	28	2	3	567	600
Batik Pekalongan	28	2	3	567	600
Batik Priangan	27	3	1	569	600
Batik Sekar	29	1	1	569	600
Batik Sidoluhur	29	1	1	569	600
Batik Sidomukti	25	5	2	568	600
Batik Sogan	29	1	0	570	600
Batik Tambal	30	0	2	568	600

Berdasarkan data klasifikasi pada Tabel 6.20 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 21 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB1

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.986667	0.8	0.923077	0.857143
Batik Betawi	0.998333	1	0.967742	0.983607
Batik Celup	1	1	1	1
Batik Cendrawasih	0.996667	1	0.9375	0.967742
Batik Ceplok	0.985	0.8	0.888889	0.842105
Batik Ciamis	0.991667	0.9	0.931034	0.915254
Batik Garutan	0.995	1	0.909091	0.952381
Batik Gentongan	0.998333	0.966667	1	0.983051
Batik Kawung	0.991667	1	0.857143	0.923077
Batik Keraton	0.993333	0.933333	0.933333	0.933333
Batik Lasem	1	1	1	1
Batik Megamendung	1	1	1	1
Batik Parang	0.991667	0.933333	0.903226	0.918033
Batik Pekalongan	0.991667	0.933333	0.903226	0.918033
Batik Priangan	0.993333	0.9	0.964286	0.931034
Batik Sekar	0.996667	0.966667	0.966667	0.966667
Batik Sidoluhur	0.996667	0.966667	0.966667	0.966667
Batik Sidomukti	0.988333	0.833333	0.925926	0.877193
Batik Sogan	0.998333	0.966667	1	0.983051
Batik Tambal	0.996667	1	0.9375	0.967742



Gambar 6. 40 *Confusion Matrix* menggunakan *EfficientNetB2*

Berdasarkan *confusion matrix* pada gambar 6.40 dapat dilihat bahwa proses *training* menggunakan *EfficientNetB2* berhasil mengklasifikasikan dengan benar sebanyak 562 gambar. Berdasarkan hasil tersebut didapat hasil Batik Bali dan Batik Ceplok yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Bali dan Batik Ceplok hanya mendapatkan 24 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan data latih terhadap hasil pengujian. Pengujian dilakukan dengan membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil *confusion matrix* ini bisa kita klasifikasikan sebagai berikut.

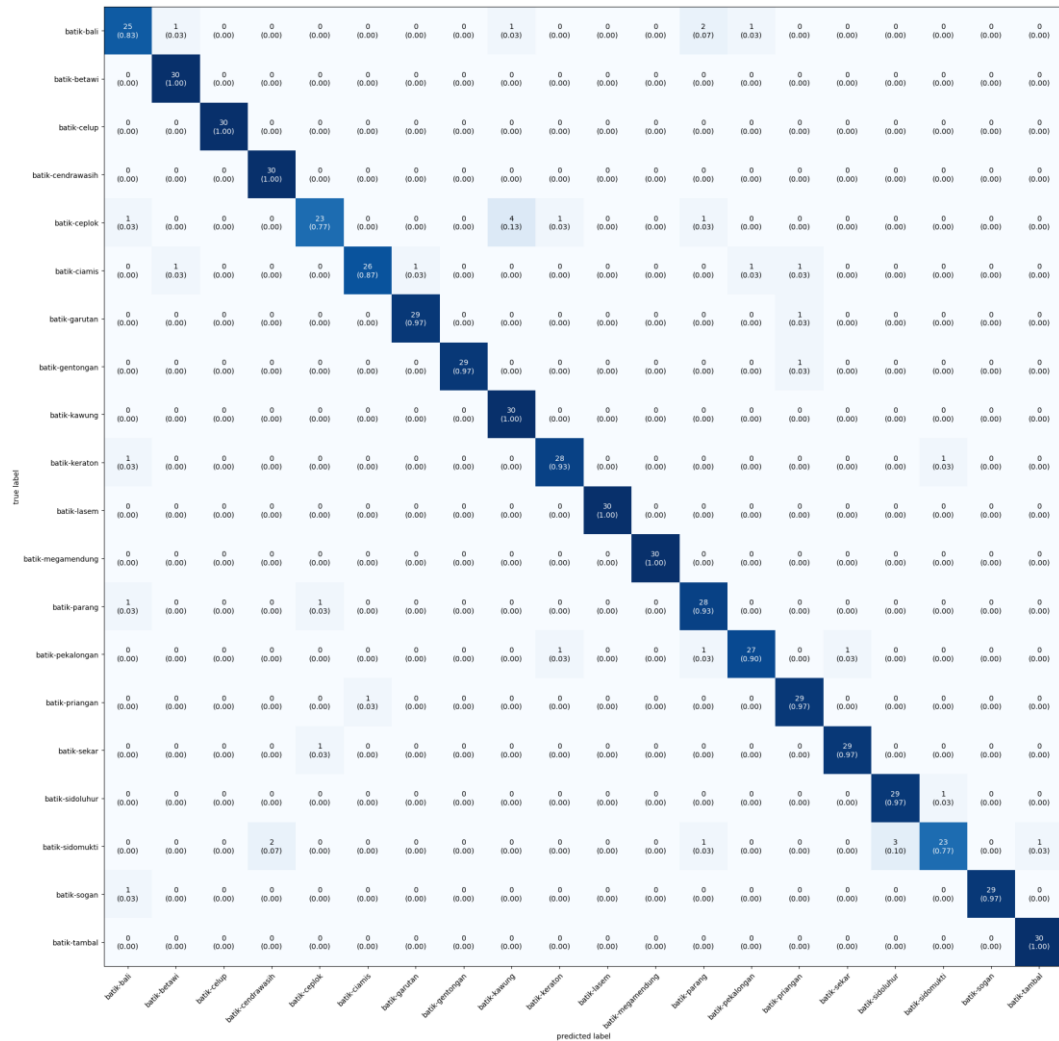
Tabel 6. 22 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB2

Kelas	Klasifikasi				Jumlah
	TP	FP	FN	TN	
Batik Bali	24	6	3	567	600
Batik Betawi	29	1	2	568	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	30	0	2	568	600
Batik Ceplok	24	6	2	568	600
Batik Ciamis	27	3	2	568	600
Batik Garutan	28	2	2	568	600
Batik Gentongan	28	2	0	570	600
Batik Kawung	29	1	5	565	600
Batik Keraton	28	2	3	567	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	1	569	600
Batik Parang	29	1	5	565	600
Batik Pekalongan	27	3	2	568	600
Batik Priangan	26	4	3	567	600
Batik Sekar	29	1	1	569	600
Batik Sidoluhur	29	1	1	569	600
Batik Sidomukti	25	5	2	568	600
Batik Sogan	30	0	1	569	600
Batik Tambal	30	0	1	569	600

Berdasarkan data klasifikasi pada Tabel 6.22 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 23 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB2

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.985	0.8	0.888889	0.842105
Batik Betawi	0.995	0.966667	0.935484	0.95082
Batik Celup	1	1	1	1
Batik Cendrawasih	0.996667	1	0.9375	0.967742
Batik Ceplok	0.986667	0.8	0.923077	0.857143
Batik Ciamis	0.991667	0.9	0.931034	0.915254
Batik Garutan	0.993333	0.933333	0.933333	0.933333
Batik Gentongan	0.996667	0.933333	1	0.965517
Batik Kawung	0.99	0.966667	0.852941	0.90625
Batik Keraton	0.991667	0.933333	0.903226	0.918033
Batik Lasem	1	1	1	1
Batik Megamendung	0.998333	1	0.967742	0.983607
Batik Parang	0.99	0.966667	0.852941	0.90625
Batik Pekalongan	0.991667	0.9	0.931034	0.915254
Batik Priangan	0.988333	0.866667	0.896552	0.881356
Batik Sekar	0.996667	0.966667	0.966667	0.966667
Batik Sidoluhur	0.996667	0.966667	0.966667	0.966667
Batik Sidomukti	0.988333	0.833333	0.925926	0.877193
Batik Sogan	0.998333	1	0.967742	0.983607
Batik Tambal	0.998333	1	0.967742	0.983607



Gambar 6. 41 *Confusion Matrix* menggunakan *EfficientNetB3*

Berdasarkan *confusion matrix* pada gambar 6.36 dapat dilihat bahwa proses *training* menggunakan *EfficientNetB3* berhasil mengklasifikasikan dengan benar sebanyak 564 gambar. Berdasarkan hasil tersebut didapat hasil Batik Ceplok dan Batik Sidomukti yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Ceplok dan Batik Sidomukti hanya mendapatkan 23 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan data latih terhadap hasil pengujian. Pengujian dilakukan dengan membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil *confusion matrix* ini bisa kita klasifikasikan sebagai berikut.

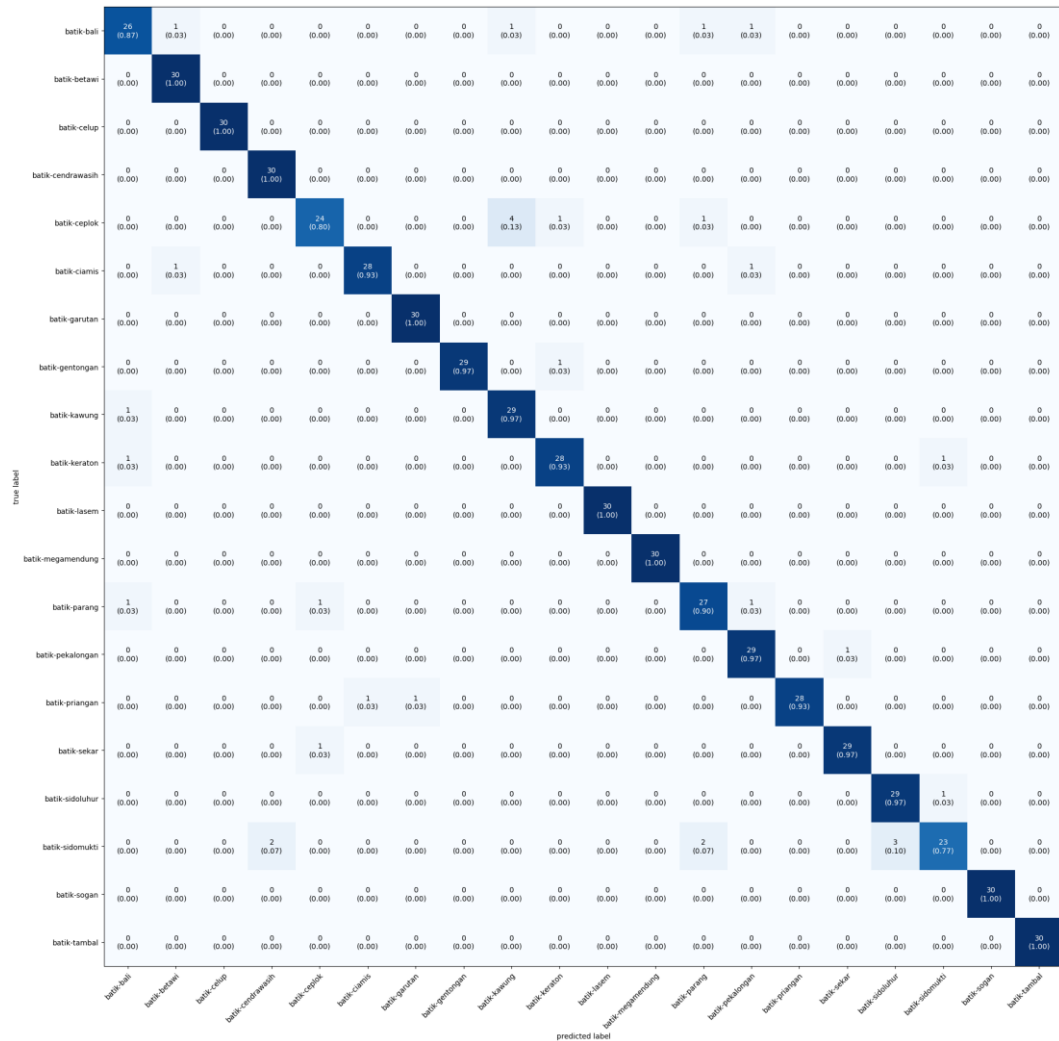
Tabel 6. 24 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB3

Kelas	Klasifikasi				Jumlah
	TP	FP	FN	TN	
Batik Bali	25	5	4	566	600
Batik Betawi	30	0	2	568	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	30	0	2	568	600
Batik Ceplok	23	7	2	568	600
Batik Ciamis	26	4	1	569	600
Batik Garutan	29	1	1	569	600
Batik Gentongan	29	1	0	570	600
Batik Kawung	30	0	5	565	600
Batik Keraton	28	2	2	568	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	0	570	600
Batik Parang	28	2	5	565	600
Batik Pekalongan	27	3	2	568	600
Batik Priangan	29	1	3	567	600
Batik Sekar	29	1	1	569	600
Batik Sidoluhur	29	1	3	567	600
Batik Sidomukti	23	7	2	568	600
Batik Sogan	29	1	0	570	600
Batik Tambal	30	0	1	569	600

Berdasarkan data klasifikasi pada Tabel 6.24 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 25 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB3

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.985	0.833333	0.862069	0.847458
Batik Betawi	0.996667	1	0.9375	0.967742
Batik Celup	1	1	1	1
Batik Cendrawasih	0.996667	1	0.9375	0.967742
Batik Ceplok	0.985	0.766667	0.92	0.836364
Batik Ciamis	0.991667	0.866667	0.962963	0.912281
Batik Garutan	0.996667	0.966667	0.966667	0.966667
Batik Gentongan	0.998333	0.966667	1	0.983051
Batik Kawung	0.991667	1	0.857143	0.923077
Batik Keraton	0.993333	0.933333	0.933333	0.933333
Batik Lasem	1	1	1	1
Batik Megamendung	1	1	1	1
Batik Parang	0.988333	0.933333	0.848485	0.888889
Batik Pekalongan	0.991667	0.9	0.931034	0.915254
Batik Priangan	0.993333	0.966667	0.90625	0.935484
Batik Sekar	0.996667	0.966667	0.966667	0.966667
Batik Sidoluhur	0.993333	0.966667	0.90625	0.935484
Batik Sidomukti	0.985	0.766667	0.92	0.836364
Batik Sogan	0.998333	0.966667	1	0.983051
Batik Tambal	0.998333	1	0.967742	0.983607



Gambar 6. 42 *Confusion Matrix* menggunakan *EfficientNetB4*

Berdasarkan *confusion matrix* pada gambar 6.42 dapat dilihat bahwa proses *training* menggunakan *EfficientNetB4* berhasil mengklasifikasikan dengan benar sebanyak 569 gambar. Berdasarkan hasil tersebut didapat hasil Batik Sidomukti yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Sidomukti hanya mendapatkan 23 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan data latih terhadap hasil pengujian. Pengujian dilakukan dengan membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil *confusion matrix* ini bisa kita klasifikasikan sebagai berikut.

Tabel 6. 26 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB4

Kelas	Klasifikasi				Jumlah
	TP	FP	FN	TN	
Batik Bali	26	4	3	567	600
Batik Betawi	30	0	2	568	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	30	0	2	568	600
Batik Ceplok	24	6	2	568	600
Batik Ciamis	28	2	1	569	600
Batik Garutan	30	0	1	569	600
Batik Gentongan	29	1	0	570	600
Batik Kawung	29	1	5	565	600
Batik Keraton	28	2	2	568	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	0	570	600
Batik Parang	27	3	4	566	600
Batik Pekalongan	29	1	3	567	600
Batik Priangan	28	2	0	570	600
Batik Sekar	29	1	1	569	600
Batik Sidoluhur	29	1	3	567	600
Batik Sidomukti	23	7	2	568	600
Batik Sogan	30	0	0	570	600
Batik Tambal	30	0	0	570	600

Berdasarkan data klasifikasi pada Tabel 6.26 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 27 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB4

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.988333	0.866667	0.896552	0.881356
Batik Betawi	0.996667	1	0.9375	0.967742
Batik Celup	1	1	1	1
Batik Cendrawasih	0.996667	1	0.9375	0.967742
Batik Ceplok	0.986667	0.8	0.923077	0.857143
Batik Ciamis	0.995	0.933333	0.965517	0.949153
Batik Garutan	0.998333	1	0.967742	0.983607
Batik Gentongan	0.998333	0.966667	1	0.983051
Batik Kawung	0.99	0.966667	0.852941	0.90625
Batik Keraton	0.993333	0.933333	0.933333	0.933333
Batik Lasem	1	1	1	1
Batik Megamendung	1	1	1	1
Batik Parang	0.988333	0.9	0.870968	0.885246
Batik Pekalongan	0.993333	0.966667	0.90625	0.935484
Batik Priangan	0.996667	0.933333	1	0.965517
Batik Sekar	0.996667	0.966667	0.966667	0.966667
Batik Sidoluhur	0.993333	0.966667	0.90625	0.935484
Batik Sidomukti	0.985	0.766667	0.92	0.836364
Batik Sogan	1	1	1	1
Batik Tambal	1	1	1	1

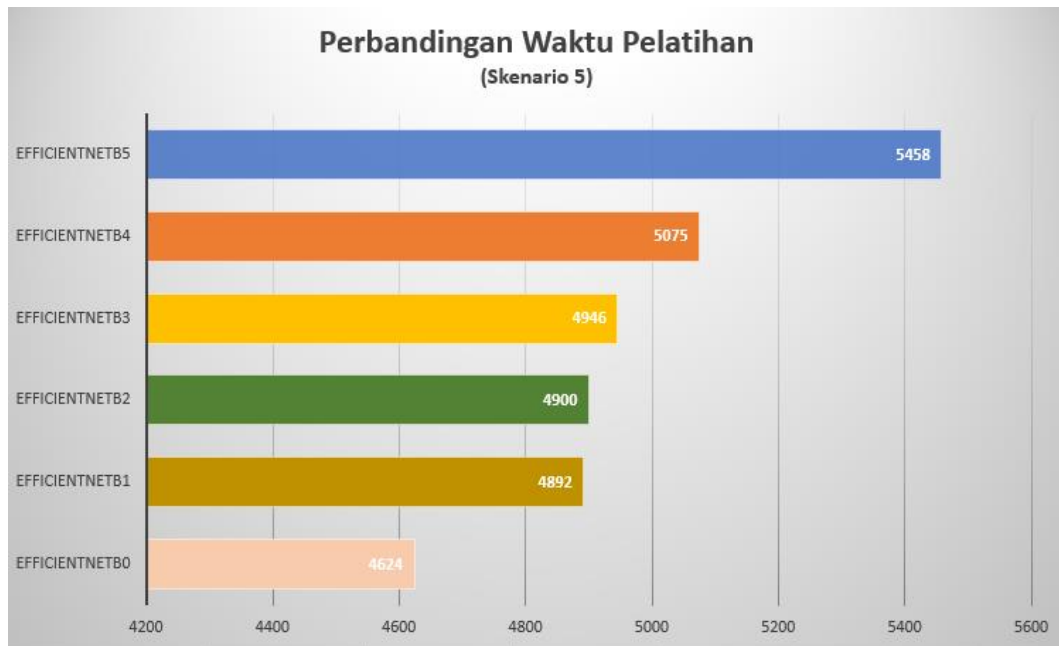
Tabel 6. 28 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB5

Kelas	Klasifikasi				Jumlah
	TP	FP	FN	TN	
Batik Bali	25	5	1	569	600
Batik Betawi	30	0	1	569	600
Batik Celup	30	0	0	570	600
Batik Cendrawasih	30	0	2	568	600
Batik Ceplok	24	6	3	567	600
Batik Ciamis	27	3	2	568	600
Batik Garutan	30	0	2	568	600
Batik Gentongan	29	1	0	570	600
Batik Kawung	30	0	5	565	600
Batik Keraton	28	2	2	568	600
Batik Lasem	30	0	0	570	600
Batik Megamendung	30	0	0	570	600
Batik Parang	28	2	5	565	600
Batik Pekalongan	29	1	3	567	600
Batik Priangan	27	3	1	569	600
Batik Sekar	29	1	0	570	600
Batik Sidoluhur	29	1	1	569	600
Batik Sidomukti	25	5	2	568	600
Batik Sogan	30	0	0	570	600
Batik Tambal	30	0	0	570	600

Berdasarkan data klasifikasi pada Tabel 6.28 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

Tabel 6. 29 Tabel Hasil Nilai dari Klasifikasi Data menggunakan Model
EfficientNetB5

Kelas	Hasil			
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Batik Bali	0.99	0.833333	0.961538	0.892857
Batik Betawi	0.998333	1	0.967742	0.983607
Batik Celup	1	1	1	1
Batik Cendrawasih	0.996667	1	0.9375	0.967742
Batik Ceplok	0.985	0.8	0.888889	0.842105
Batik Ciamis	0.991667	0.9	0.931034	0.915254
Batik Garutan	0.996667	1	0.9375	0.967742
Batik Gentongan	0.998333	0.966667	1	0.983051
Batik Kawung	0.991667	1	0.857143	0.923077
Batik Keraton	0.993333	0.933333	0.933333	0.933333
Batik Lasem	1	1	1	1
Batik Megamendung	1	1	1	1
Batik Parang	0.988333	0.933333	0.848485	0.888889
Batik Pekalongan	0.993333	0.966667	0.90625	0.935484
Batik Priangan	0.993333	0.9	0.964286	0.931034
Batik Sekar	0.998333	0.966667	1	0.983051
Batik Sidoluhur	0.996667	0.966667	0.966667	0.966667
Batik Sidomukti	0.988333	0.833333	0.925926	0.877193
Batik Sogan	1	1	1	1
Batik Tambal	1	1	1	1



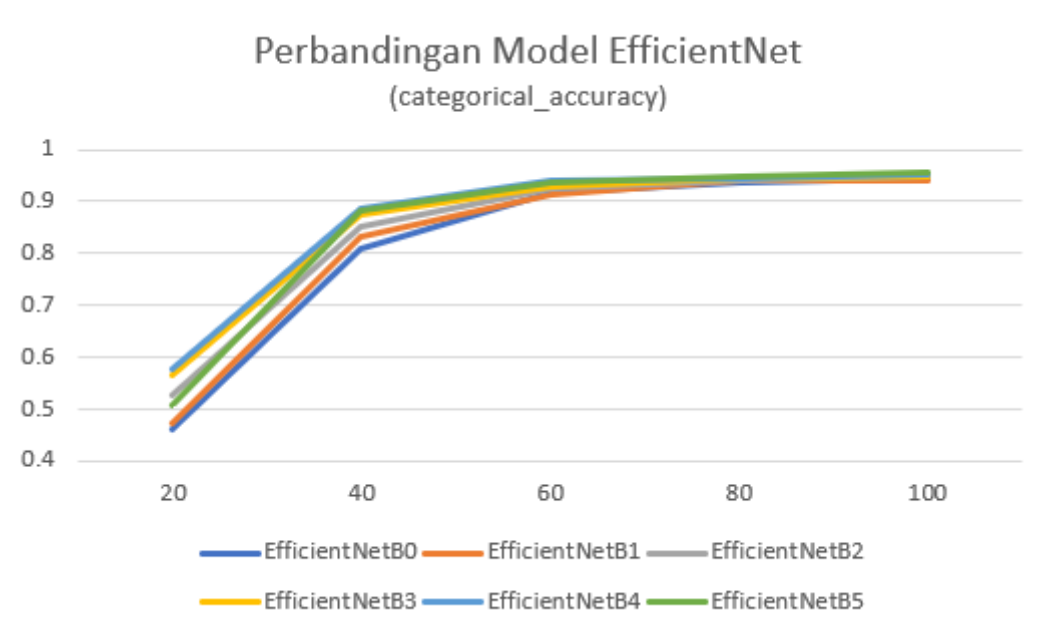
Gambar 6. 44 Perbandingan Waktu yang Digunakan untuk Pelatihan pada Skenario 5

Pada gambar 6.44 menunjukkan bahwa pelatihan menggunakan model *EfficientNetB0* mendapatkan waktu yang lebih cepat daripada menggunakan model yang lainnya. Hal ini dikarenakan perbedaan pada jumlah layer di setiap modelnya.

Berdasarkan uji coba membandingkan model dengan menggunakan *EfficientNetB0* hingga *EfficientNetB5* menunjukkan bahwa percobaan menggunakan model *EfficientNetB5* memiliki nilai yang lebih tinggi daripada model yang lainnya. Perbandingan ini bisa dilihat pada tabel 6.30

Tabel 6. 30 Tabel classification metrics skenario 5

Model	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
<i>EfficientNetB0</i>	0.946	0.95	0.95	0.95
<i>EfficientNetB1</i>	0.945	0.95	0.94	0.94
<i>EfficientNetB2</i>	0.936	0.94	0.94	0.94
<i>EfficientNetB3</i>	0.94	0.94	0.94	0.94
<i>EfficientNetB4</i>	0.948	0.95	0.95	0.95
<i>EfficientNetB5</i>	0.95	0.95	0.95	0.95



Gambar 6. 45 Perbandingan Hasil categorical_accuracy pada *EfficientNet*