

# IMPLEMENTASI DAN KLASIFIKASI JENIS-JENIS BATIK MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN MODEL ARSITEKTUR RESNET

Rizal Whisnu Wiratama<sup>1</sup>

<sup>1</sup>Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang

<sup>1</sup>rizalwhisnu00@gmail.com

---

## Abstrak

Penelitian ini mengeksplorasi implementasi metode *deep learning* menggunakan *Convolutional Neural Network (CNN)* dengan model *ResNet* untuk mengenali jenis-jenis batik. Batik, sebagai warisan budaya Indonesia, seringkali sulit diidentifikasi motifnya, terutama oleh masyarakat awam. Kecerdasan buatan, khususnya dalam bentuk AI, dapat mempermudah proses pengenalan pola batik melalui pengolahan citra digital. *Deep learning*, sebagai bagian dari AI, memungkinkan komputer untuk belajar mengklasifikasikan objek langsung dari gambar dengan memanfaatkan CPU, RAM, dan GPU secara efisien. *Convolutional Neural Network (CNN)*, terutama dengan arsitektur *ResNet*, terbukti efektif dalam pengenalan citra digital dengan tingkat akurasi tinggi dan efisiensi. Model *ResNet*, dengan ukuran yang relatif lebih kecil dan waktu inferensi lebih cepat, dapat dengan mudah disesuaikan untuk berbagai tugas pembelajaran transfer. Melalui implementasi ini, penelitian ini bertujuan meningkatkan pemahaman dan apresiasi terhadap kebudayaan Indonesia melalui bantuan pengenalan jenis-jenis batik.

**Kata kunci :** Pengolahan Citra, Batik, Convolutional Neural Network, ResNet

---

## 1. Pendahuluan

Indonesia, dengan populasi lebih dari 255 juta jiwa pada tahun 2015, adalah negara dengan jumlah penduduk terbesar kelima di dunia dan juga negara kepulauan terbesar dengan 13.466 pulau (Statistik Indonesia, 2002). Kekayaan budaya dan keanekaragaman suku, ras, agama, dan bahasa daerahnya menjadikan Indonesia unik dan beragam. Salah satu aspek penting dari kekayaan budaya ini adalah Batik, sebuah hasil kebudayaan yang memiliki motif khas di setiap daerah (Arsa & Susila, 2019).

Namun, banyaknya motif batik di Indonesia membuat identifikasi jenisnya menjadi tantangan, terutama bagi masyarakat awam. Oleh karena itu, penelitian ini berfokus pada penggunaan teknologi, khususnya *Convolutional Neural Network (CNN)* dan *Residual Network (ResNet)*, untuk membantu dalam proses identifikasi dan klasifikasi motif batik. Penelitian sebelumnya oleh Tumewu et al., (2020) telah menunjukkan bahwa penggunaan CNN dan teknik augmentasi data dapat meningkatkan akurasi identifikasi motif batik. Namun, penelitian ini mencoba untuk lebih jauh lagi

dengan menerapkan augmentasi data yang lebih beragam dan menggunakan arsitektur CNN yang lebih canggih, yaitu *ResNet*.

*ResNet*, yang diusulkan oleh He, dkk. pada tahun 2015, adalah arsitektur CNN yang dirancang untuk mengatasi permasalahan dalam pelatihan *Deep Learning (DL)*, seperti waktu pelatihan yang lama dan keterbatasan jumlah lapisan. Dalam penelitian ini, *ResNet* digunakan untuk mengklasifikasikan citra batik dengan target hasil yang maksimal. Beberapa jenis arsitektur *ResNet* yang digunakan antara lain *ResNet50*, *ResNet101*, dan *ResNet152*. Selain itu, teknik augmentasi data seperti *Scale*, *Random Erase*, *Rotation*, dan *Flip* diterapkan pada *dataset* untuk memperbesar jumlah dan meningkatkan variasi data.

Tujuan dari penelitian ini adalah untuk menerapkan implementasi dari metode *deep learning* menggunakan *Convolutional Neural Network (CNN)* dengan arsitektur *ResNet* untuk membantu mengenali jenis-jenis batik. Diharapkan, penelitian ini dapat memberikan kontribusi positif dalam pelestarian dan pengenalan batik, serta membantu dalam

pengembangan teknologi pengenalan pola dan citra.

## 2. Landasan Teori

### 2.1 Machine Learning

*Machine Learning (ML)* adalah salah satu aplikasi dari *Artificial Intelligent (AI)* yang fokus kepada pengembangan sebuah sistem yang mampu belajar sendiri tanpa harus diprogram berulang kali. ML membutuhkan sebuah data (*data training*) sebagai proses learning sebelum menghasilkan sebuah hasil. Jadi, secara sederhana dapat dijelaskan bahwa *Machine Learning* adalah pemrograman komputer untuk mencapai kriteria/performa tertentu dengan menggunakan sekumpulan data training atau pengalaman di masa lalu (*past experience*) (Chazar, 2020).

Dalam pembelajaran *machine learning*, terdapat beberapa skenario-skenario, seperti:

#### 1. Supervised Learning

Penggunaan skenario *supervised learning*, pembelajaran menggunakan masukan data pembelajaran yang telah diberi label. Setelah itu, membuat prediksi dari data yang telah diberi label.

#### 2. Unsupervised Learning

Penggunaan skenario *unsupervised learning*, pembelajaran menggunakan masukan data pembelajaran yang tidak diberi label. Setelah itu, mencoba untuk mengelompokkan data berdasarkan karakteristik-karakteristik yang ditemui.

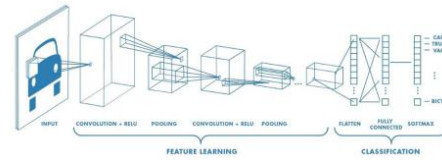
#### 3. Reinforcement Learning

Pada skenarion *reinforcement learning* fase pembelajaran dan tes saling dicampur. Untuk mengumpulkan informasi pembelajar secara aktif dengan berinteraksi ke lingkungan sehingga untuk mendapatkan balasan untuk setiap aksi dari pembelajar

### 2.2 Convolutional Neural Network (CNN)

*Convolutional Neural Network (CNN)* merupakan pengembangan dari *multilayer perceptron (MLP)* yang didesain untuk mengolah data dua dimensi dalam bentuk citra. *Convolutional Neural Network (CNN)* ini termasuk kedalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada dasarnya klasifikasi citra dapat digunakan dengan *multilayer perceptron (MLP)*, akan tetapi dengan metode *multilayer perceptron (MLP)* kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data cita dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik. Penelitian awal yang mendasari penemuan *Convolutional Neural Network (CNN)* ini pertama kali dilakukan oleh Hubel dan Wiesel (Hubel & Wiesel, T, 1968) mengenai *virtual cortex* pada indera penglihatan kucing. Secara teknis, *Convolutional Neural Network (CNN)* adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (*input*) dan

keluaran (*output*) dari setiap tahap adalah terdiri dari beberapa *array* yang biasa disebut *feature map*. Setiap tahap terdiri dari tiga layer yaitu konvolusi, fungsi aktivasi layer dan pooling layer. Berikut adalah jaringan arsitektur *Convolutional Neural Network*:



Gambar 2. 1 *Convolutional Neural Network (CNN)*

Berdasarkan gambar diatas, Tahap pertama pada arsitektur *Convolutional Neural Network (CNN)* adalah tahap konvolusi. Tahap ini dilakukan dengan menggunakan sebuah kernel dengan ukuran tertentu. Perhitungan jumlah kernel yang dipakai tergantung dari jumlah fitur yang dihasilkan. Kemudian dilanjutkan menuju fungsi aktivasi, biasanya menggunakan fungsi aktivasi ReLU (*Rectifier Linear Unit*), Selanjutnya setelah keluar dari proses fungsi aktivasi kemudian melalui proses *pooling*. Proses ini diulang beberapa kali sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully connected neural network*, dan dari *fully connected network* adalah *output class*.

### 2.3 Residual Network

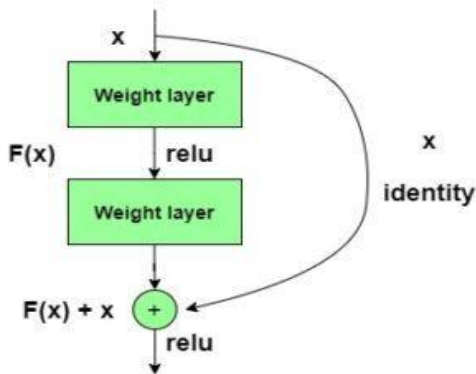
*Residual Network (ResNet)* dan variasinya (*ResNet-18, ResNet-50, ResNet-101, ResNet-152, dan ResNext*) adalah arsitektur *deep neural network (DNN)* yang sukses. *ResNet* dirancang untuk mengatasi masalah gradien yang menghilang dalam pelatihan *deep learning*, yang terjadi ketika gradien menjadi semakin kecil dengan bertambahnya jumlah lapisan (Wu et al., 2019). *ResNet* mencapai ini dengan mengusulkan koneksi pintasan identitas yang melewati beberapa lapisan dan menggunakan fungsi aktivasi lapisan sebelumnya. Ini memungkinkan desain DNN buatan dengan banyak lapisan tanpa khawatir tentang masalah gradien yang hilang. Pada blok residual, perhitungan Persamaan dilakukan:

$$x_{l+1} = x_l + F(x_l \cdot W_l)$$

$x_l$  dan  $x_{l+1}$  adalah input dan output dari blok residual ke- $l$ .  $W_l$  adalah himpunan bobot, dan  $F(x_l \cdot W_l)$  adalah fungsi residual. Persamaan 5 dapat diwujudkan dengan '*shortcut connection*' dengan melakukan pemetaan identitas dan melewati satu atau lebih lapisan. Hal ini dilakukan berulang-ulang untuk setiap blok sisa  $L$  sehingga diperoleh:

$$x_L = x_1 + \sum_{i=1}^{L-1} F(x_i \cdot W_i)$$

Dari *gradien* fungsi *loss*, ketika bobot sangat kecil, gradien lapisan tidak hilang. Namun demikian, masih belum ada pengetahuan tentang posisi bagian objek (Falahkhi et al., n.d.).



ResNet memiliki berbagai jenis arsitektur, mulai dari 18 hingga 152 layer (Hasan Mahmud & al Faraby, 2019). ResNet adalah model yang telah dilatih sebelumnya, sehingga tidak memerlukan konfigurasi khusus untuk mengatur layer di dalamnya. ResNet bekerja dengan membangun jaringan yang lebih dalam dibandingkan dengan jaringan biasa lainnya dan secara bersamaan menemukan jumlah lapisan yang dioptimalkan untuk meniadakan masalah gradien yang hilang.

Pada ResNet-50 layer, untuk meningkatkan dimensi menggunakan projection shortcut pada pintasan identitas dan mengubah 2 layer block pada ResNet 34 layer dengan 3 layer bottleneck block. Sedangkan pada ResNet 101 dan 152 layer menggunakan lebih dari 3 layer block. Meskipun ResNet 152 layer kedalamannya meningkat secara signifikan, namun masih mempunyai kompleksitas yang lebih rendah dari ResNet50.

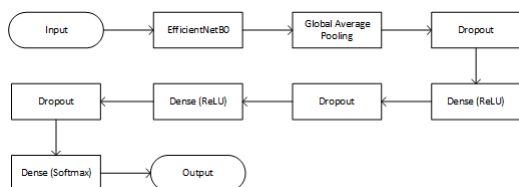
Penelitian sebelumnya telah menunjukkan bahwa ResNet dapat digunakan untuk mendeteksi penyakit Covid-19 dengan akurasi tinggi dan evaluasi matriks yang baik, meskipun data yang digunakan sedikit. Dengan akurasi yang tinggi dan hasil evaluasi matriks yang baik, model tersebut dapat dipertimbangkan untuk diverifikasi sehingga bisa digunakan untuk mendeteksi penyakit Covid-19.

Penelitian ini berfokus pada penggunaan ResNet untuk klasifikasi jenis-jenis batik berdasarkan motifnya. Tujuan penelitian ini adalah untuk mengetahui seberapa akurat metode ResNet dalam melakukan klasifikasi jenis-jenis batik berdasarkan motifnya dan menjadi referensi bagi pembaca untuk melakukan penelitian yang berkaitan dengan klasifikasi gambar menggunakan metode ResNet.

3. Metodologi Penelitian

3.1 Perancangan Arsitektur

Pada penelitian ini, peneliti akan menggunakan model Convolutional Neural Network (CNN) dengan menggunakan arsitektur ResNet untuk memperoleh akurasi dalam penentuan jenis-jenis batik.



Data yang masukannya berukuran 224x224x3, dimana angka 3 merupakan channel RGB (karena data masukannya adalah gambar berwarna). Selanjutnya digunakan arsitektur ResNet-50 sebagai base model dari layer Convolutional Neural Network (CNN) yang akan digunakan. Selanjutnya ditambahkan layer Global Average Pooling untuk mengurangi jumlah parameter sehingga proses komputasi dapat berjalan lebih cepat. Lalu ditambahkan dengan dense dan dropout, gunanya yaitu untuk mengurangi kemungkinan overfitting. Overfitting terjadi saat model terlalu menggeneralisasi data sehingga tidak dapat mengenali data yang baru dengan baik. Hal ini ditandai dengan nilai akurasi yang jauh lebih tinggi dibandingkan dengan nilai validasi. Layer dense yang terakhir menggunakan fungsi aktivasi softmax karena hasilnya akan berupa kategorik. Kemudian output didapatkan berupa hasil klasifikasi 20 jenis batik.

4. Analisis dan Perancangan Sistem

4.1 Kebutuhan Proses

Setelah didapatkan dataset, kemudian dataset tersebut akan diproses supaya mendapatkan informasi yang dibutuhkan. Ada beberapa proses yang harus dilakukan, diantaranya:

4.1.1 Proses Augmentasi

Augmentasi data adalah strategi yang memungkinkan praktisi untuk secara signifikan meningkatkan keragaman data yang tersedia untuk model pelatihan, tanpa benar-benar mengumpulkan data baru (Sanjaya & Ayub, 2020). Penggunaan proses augmentasi diharapkan dapat menaikkan performa model karena mesin akan berhasil mengenali lebih banyak objek dari bentuk serta pola yang beragam jenisnya. Berikut merupakan hasil augmentasi dataset batik untuk pelatihan ini.



Gambar 4. 1 Hasil Proses Augmentasi

4.1.2 Proses Preprocessing

Proses preprocessing digunakan untuk meningkatkan kualitas gambar agar lebih baik daripada sebelumnya, dengan tujuan mempermudah pengenalan ciri-ciri pada gambar tersebut. Pada pelatihan kali ini citra akan melalui proses preprocessing dengan menggunakan filter Gaussian Noise.

4.1.3 Proses Penambahan Model Pelatihan

Pada proses ini, pelatihan akan menambahkan model pelatihan yang berfokus pada ResNet-50. Pada proses ini pelatihan akan mengaktifkan model

*ResNet-50*. Model akan menerima citra dengan ukuran 224 x 224 piksel dan 3 saluran warna (RGB).

**4.1.4 Proses Penambahan Layer**

Pada proses ini, pelatihan akan menambahkan beberapa layer guna meningkatkan hasil pelatihan dan mendapatkan hasil yang terbaik.

**4.1.5 Proses Penambahan Optimizer**

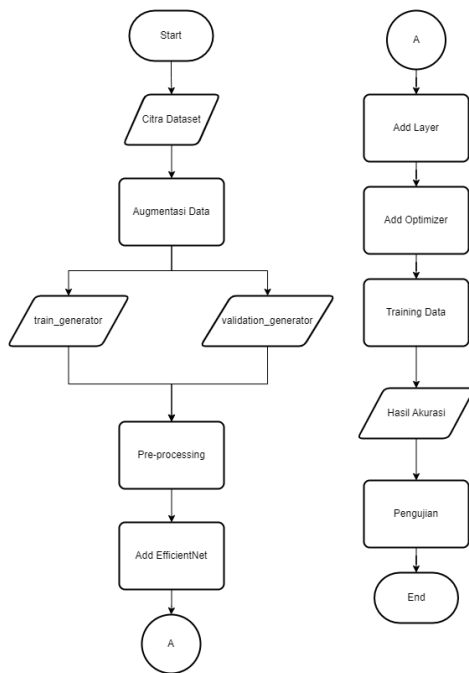
Pada proses ini, pelatihan akan menambahkan optimizer. Optimizer ini digunakan mengoptimalkan parameter-parameter model neural network yang akan digunakan selama pelatihan. Pada proses ini pelatihan akan membuat objek model.compile untuk menambahkan optimizer RMSprop yang akan digunakan selama proses pelatihan dengan menggunakan learning rate (tingkat pembelajaran) sebesar 0.0001

**4.1.6 Proses Pelatihan**

Pada proses ini mesin akan melakukan pelatihan dengan proses-proses yang telah dirancang

**4.2 Perancangan Sistem**

Pada tahap ini merupakan proses perancangan sistem untuk menghasilkan akurasi yang terbaik dari pelatihan yang telah dirancang



Pada pelatihan Implementasi *Image Classification* pada Jenis-Jenis Batik Menggunakan Algoritma *Convolutional Neural Network* dengan Model Arsitektur *ResNet-50*, alur sistem akan dijelaskan pada berikut ini:

1. Mesin akan melakukan input data berupa dataset citra batik yang berisikan 20 class dengan jumlah citra sebanyak 2000 gambar.
2. Mesin akan melakukan proses augmentasi dataset yang berisikan citra batik.
3. Mesin akan membagi data citra menjadi 2, yaitu data train yang diargumenkan dengan

train\_genetator dan data validasi yang diargumenkan dengan validation\_generator.

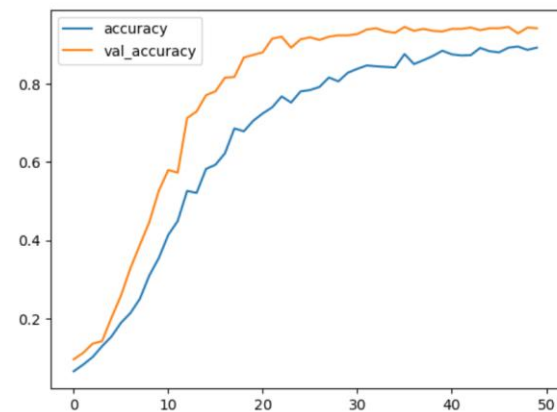
4. Mesin akan melakukan proses pre-processing. Pada tahap proses pre-processing kali ini, mesin akan menambahkan filter GaussianNoise senilai 0,2 yang akan digunakan untuk menghasilkan noise.
5. Mesin akan menambahkan model *ResNet-50* didalam objek base\_model.
6. Mesin akan memperkenalkan lapisan tambahan yang bermanfaat untuk meningkatkan hasil pelatihan dan mencapai hasil yang optimal.
7. Menambahkan *optimizer* yang digunakan mengoptimalkan parameter-parameter model neural network yang akan digunakan selama pelatihan.
8. Mesin akan melakukan proses training data. Proses training data ini akan melibatkan seluruh proses yang telah dirancang

**5. Implementasi**

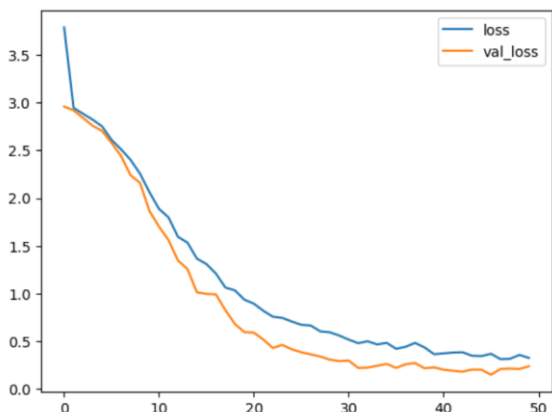
Hasil mplementasi Uji Coba Image Classification pada Jenis-Jenis Batik Menggunakan Algoritma Convolutional Neural Network dengan Model Arsitektur ResNet dengan beberapa skenario yang sudah dirancang untuk menemukan model pelatihan terbaik.

**5.1. Skenario 1**

Pada uji coba skenario 1 ini dilakukan dengan melakukan proses training dengan menggunakan algoritma Convolutional Neural Network tanpa menggunakan ResNet. Berikut merupakan hasil uji coba skenario 1. Berdasarkan proses *training* akan mendapatkan hasil berupa grafik *training* sebagai berikut.



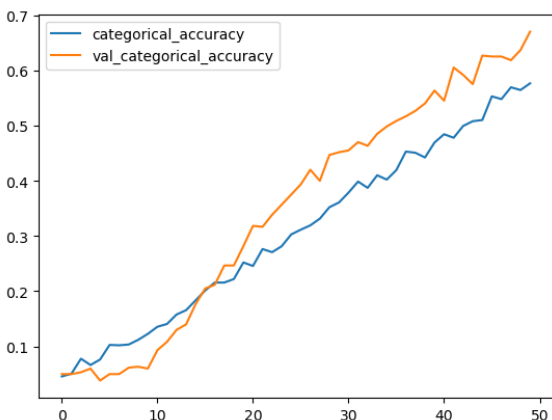
Berdasarkan grafik proses training pada gambar 6.1 menggunakan algoritma Convolutional Neural Network tanpa menggunakan ResNet mengalami kenaikan accuracy pada setiap epoch-nya. Hasil akhir pada proses training ini berada di atas 80% yang berakhir pada epoch ke 50.



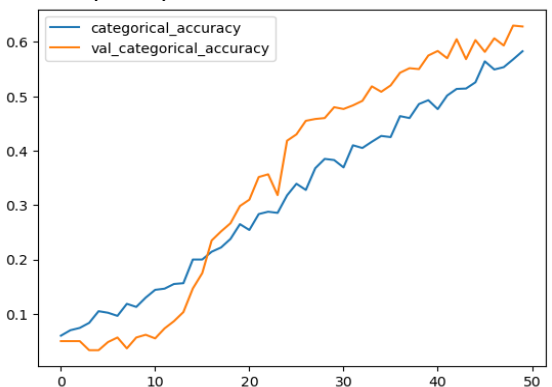
Berdasarkan grafik proses training pada gambar diatas menggunakan algoritma Convolutional Neural Network tanpa menggunakan ResNet mengalami penurunan loss pada setiap epoch-nya. Hasil akhir pada proses training ini berada di angka mendekati 0 yang berakhir pada epoch ke 50.

**5.2. Skenario 2**

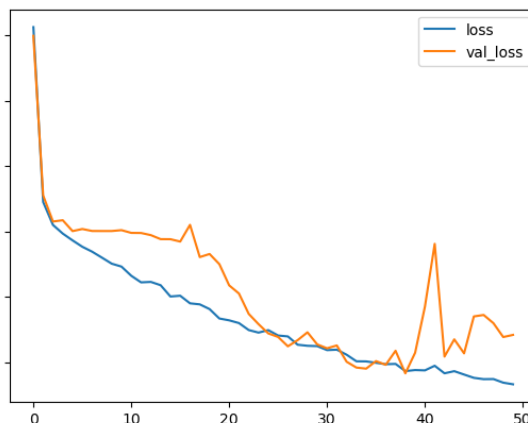
Pada uji coba skenario 2 ini dilakukan dengan membandingkan model dengan menggunakan data preprocessing dan dengan model tanpa preprocessing. Preprocessing yang digunakan pada uji coba ini yaitu menggunakan filter GaussianNoise dan akan melakukan pelatihan sebanyak 50 epoch. Berikut merupakan hasil uji coba skenario 2.



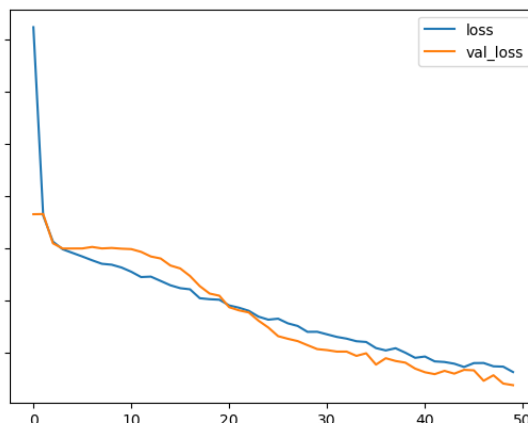
Berdasarkan grafik proses training pada gambar.. dengan data preprocessing mengalami kenaikan accuracy pada setiap epoch-nya. Hasil akhir pada proses training ini hampir berada di 70% yang berakhir pada epoch ke 50.



Berdasarkan grafik proses training pada gambar .. tanpa data preprocessing mengalami kenaikan accuracy pada setiap epoch-nya. Hasil akhir pada proses training ini berada di atas 60% yang berakhir pada epoch ke 50.



Berdasarkan grafik proses training pada gambar 6.13 menggunakan Optimizer RMSprop dengan learning rate 0.0001 mengalami penurunan loss yang tidak stabil setiap epoch-nya. Pada data validasi di epoch 41 menuju epoch 42 mengalami kenaikan yang cukup drastis dari 1.8 ke 2.8, namun langsung mengalami penurunan kembali pada epoch 42 menuju epoch 43 dari 2.8 ke 1.1. Hal ini terjadi berulang kali hingga ke epoch 50. Dan hasil akhir pada proses training ini berada di sekitar angka 1.3 yang berakhir pada epoch ke 50.

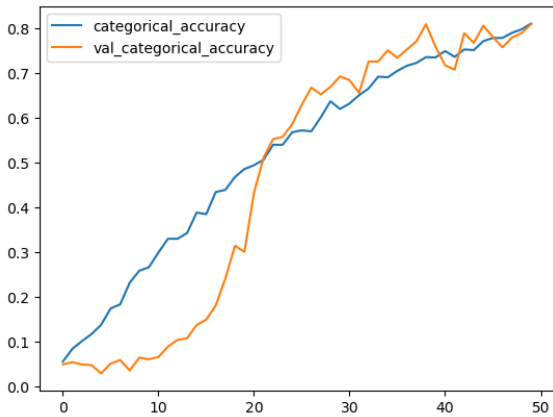


Berdasarkan grafik proses training pada gambar 6.14 menggunakan Optimizer Adam dengan learning rate 0.0001 mengalami penurunan loss pada setiap epoch-nya. Hasil akhir pada proses training ini berada di angka mendekati 0 yang berakhir pada epoch ke 50.

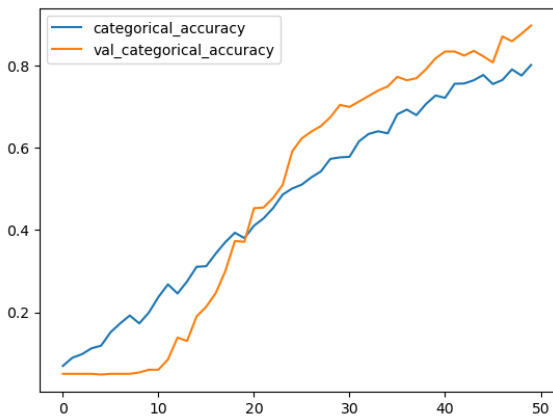
**5.3. Skenario 3**

Pada uji coba skenario 3 ini dilakukan dengan membandingkan membandingkan model dengan menggunakan optimizer Adam dan RMSprop dengan learning rate sebesar 0.0001 dan melakukan pelatihan sebanyak 50 epoch. Berikut merupakan hasil uji coba skenario 3.

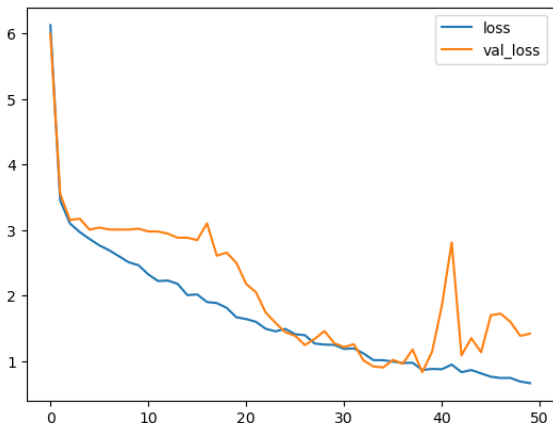




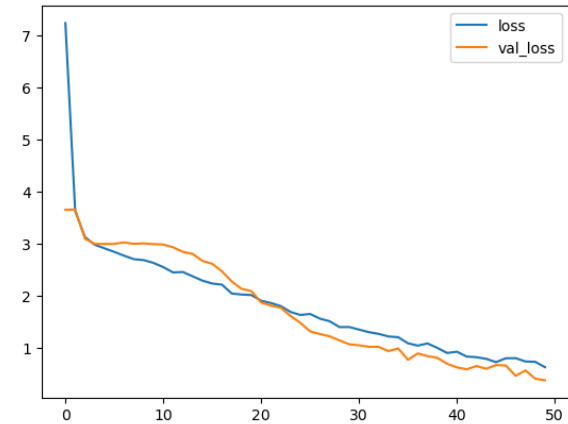
Berdasarkan grafik proses training pada gambar 6.11 menggunakan optimizer RMSprop dengan learning rate sebesar 0.0001 mengalami kenaikan accuracy pada setiap epoch-nya. Pada data validasi pada epoch 39 menuju epoch 40 sempat mengalami penurunan yang cukup lumayan dari 0.81 ke 0.76. Kemudian mengalami kenaikan lagi di epoch 42 menuju ke epoch 43 dari 0.70 ke 0.79. Hal ini terjadi berulang kali hingga ke epoch 50. Hasil akhir pada proses training ini berada di atas 80% yang berakhir pada epoch ke 50.



Berdasarkan grafik proses training pada gambar 6.12 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 50.



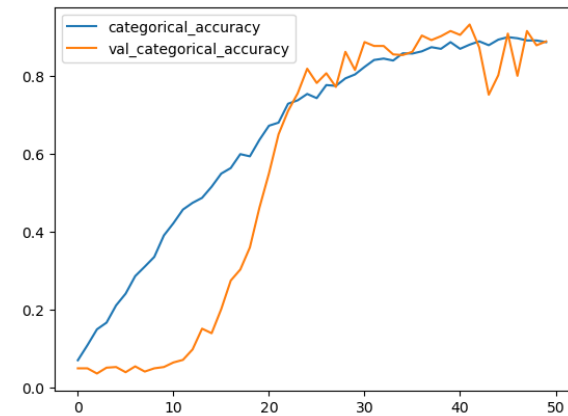
Berdasarkan grafik proses training pada gambar 6.13 menggunakan Optimizer RMSprop dengan learning rate 0.0001 mengalami penurunan loss yang tidak stabil setiap epoch-nya. Pada data validasi di epoch 41 menuju epoch 42 mengalami kenaikan yang cukup drastis dari 1.8 ke 2.8, namun langsung mengalami penurunan kembali pada epoch 42 menuju epoch 43 dari 2.8 ke 1.1. Hal ini terjadi berulang kali hingga ke epoch 50. Dan hasil akhir pada proses training ini berada di sekitar angka 1.3 yang berakhir pada epoch ke 50.



Berdasarkan grafik proses training pada gambar 6.14 menggunakan Optimizer Adam 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 50.

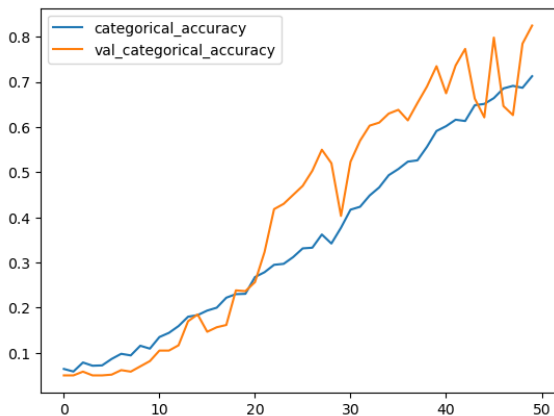
#### 5.4. Skenario 4

Pada uji coba skenario 4 ini dilakukan dengan membandingkan model menggunakan ResNet50, ResNet 101, dan ResNet152 dengan optimizer RMSprop dengan learning rate sebesar 0.0001 dan melakukan pelatihan sebanyak 50 epoch. Berdasarkan proses training akan mendapatkan hasil berupa grafik training sebagai berikut :

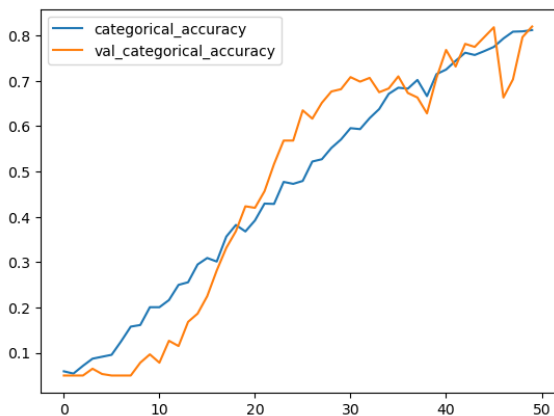


Berdasarkan grafik proses training pada gambar 6.17 menggunakan ResNet50 mengalami kenaikan accuracy yang tidak stabil pada setiap epoch-nya. Pada data validasi di epoch 43 menuju epoch 44 juga sempat mengalami penurunan terbesar dari 0.90 ke 0.73. Kemudian di epoch 47 menuju 48 mengalami kenaikan dari 0.80 ke 0.91. Begitu juga dengan data

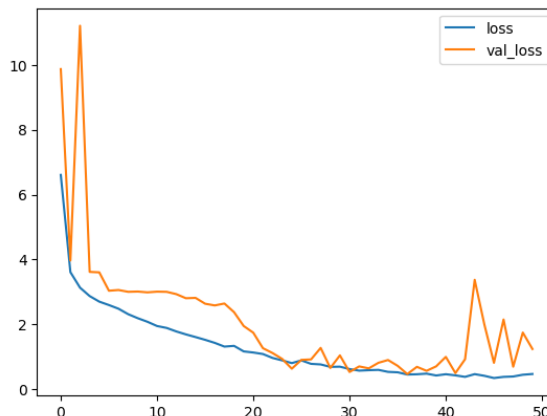
training, namun tidak terlalu naik turun seperti data validasi. Dan hasil akhir pada proses training ini berada di atas 80% yang berakhir pada epoch ke 50.



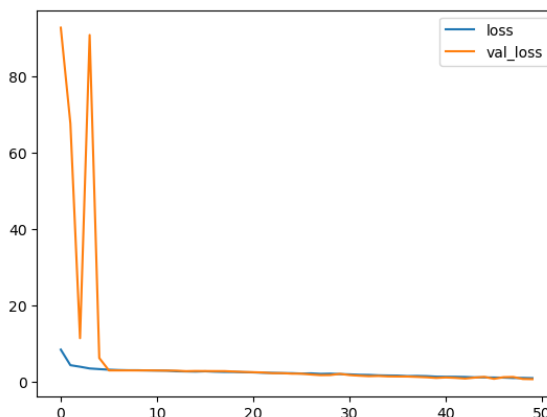
Berdasarkan grafik proses training pada gambar 6.18 menggunakan ResNet101 mengalami kenaikan accuracy yang tidak stabil pada setiap epoch-nya. Pada data validasi di epoch 29 menuju epoch 30 juga sempat mengalami penurunan drastis dari 0.52 ke 0.40. Namun di epoch 30 menuju epoch 31 langsung mengalami kenaikan dari 0.40 ke 0.53. Kemudian setelah mulai stabil mengalami penurunan lagi pada epoch 43 menuju epoch 44. Dan mengalami kenaikan lagi pada epoch 45 menuju epoch 46 dari 0.62 ke 0.78. Kemudian hal itu terjadi berulang kali, begitu juga dengan data training, namun tidak terlalu naik turun seperti data validasi. Dan hasil akhir pada proses training ini berada di atas 80% yang berakhir pada epoch ke 50.



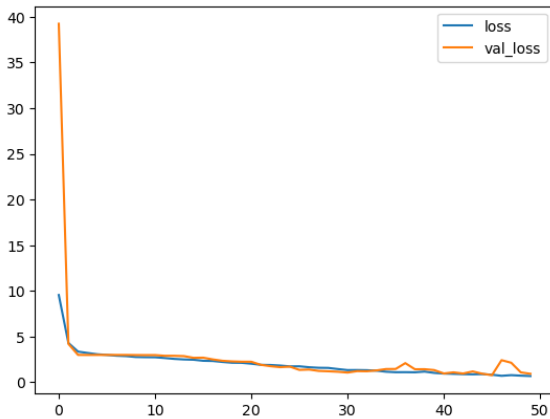
Berdasarkan grafik proses training pada gambar 6.19 menggunakan ResNet152 mengalami kenaikan accuracy yang tidak stabil pada setiap epoch-nya. Pada data validasi di epoch 36 menuju epoch 37 juga sempat mengalami penurunan dari 0.71 ke 0.67. Namun pada epoch 39 menuju epoch 40 langsung mengalami kenaikan dari 0.62 ke 0.70. Kemudian sempat mengalami penurunan lagi pada epoch 46 menuju epoch 47 dari 0.82 ke 0.66. Hal itu terjadi berulang kali, begitu juga dengan data training, namun tidak terlalu naik turun seperti data validasi. Dan hasil akhir pada proses training ini berada di atas 80% yang berakhir pada epoch ke 50.



Berdasarkan grafik proses loss pada gambar 6.20 menggunakan ResNet50 mengalami penurunan loss yang tidak stabil. Pada data validasi di epoch 1 menuju epoch 2 mengalami penurunan drastis dari 9.8 ke 3.9, namun langsung mengalami peningkatan drastis pada epoch 2 menuju epoch 3 dari 3.9 ke 11.2. Setelah turun terus menerus hingga mengalami peningkatan lagi pada epoch 43 menuju epoch 44 dari 0.8 ke 3.7. Hal ini terjadi berulang kali hingga ke epoch 50. Begitu juga dengan data training, namun tidak terlalu naik turun seperti data validasi. Hasil akhir pada proses training ini berada di sekitar angka 1 dan akan mendekati 0 yang berakhir pada epoch ke 50.



Berdasarkan grafik proses loss pada gambar 6.21 menggunakan ResNet101 mengalami penurunan loss yang tidak stabil. Pada data validasi di epoch 1 menuju epoch 2 mengalami penurunan drastis dari 92.8 ke 67.7, namun langsung mengalami peningkatan drastis juga pada epoch 3 menuju epoch 4 dari 11.4 ke 90.8. Kemudian langsung mengalami penurunan drastis lagi dari epoch 4 menuju epoch 5 dari 90.8 ke 6.2. Dan setelah turun terus menerus hingga epoch 50. Hasil akhir pada proses training ini berada di angka mendekati 0 yang berakhir pada epoch ke 50.



Berdasarkan grafik proses loss pada gambar 6.22 menggunakan ResNet152 mengalami penurunan loss yang tidak stabil. Pada data validasi di epoch 1 menuju epoch 2 mengalami penurunan drastis dari 39.2 ke 4.2. Setelah mengalami penurunan yang terus menerus, pada epoch 46 menuju epoch 47 sedikit mengalami kenaikan dari 0.7 ke 2.4. Dan setelah itu turun hingga epoch 50. Begitu juga dengan data training, namun tidak terlalu naik turun seperti data validasi. Hasil akhir pada proses training ini berada di angka mendekati 0 yang berakhir pada epoch ke 50.

**6. Hasil dan Evaluasi**

**6.1 Skenario 1**

No	Kelas	Klasifikasi				Jumlah
		TP	FN	FP	TN	
0	Batik Bali	25	5	2	568	600
1	Batik Betawi	29	1	1	569	600
2	Batik Celup	30	0	0	570	600
3	Batik Cendrawasih	30	0	0	570	600
4	Batik Ceplok	26	4	2	568	600
5	Batik Ciamis	28	2	3	567	600
6	Batik Garutan	30	0	5	565	600
7	Batik Gentongan	29	1	0	570	600
8	Batik Kawung	28	2	4	566	600
9	Batik Keraton	29	1	4	566	600
10	Batik Lasem	30	0	0	570	600
11	Batik Megamendung	30	0	0	570	600

12	Batik Parang	30	0	2	568	600
13	Batik Pekalongan	29	1	3	567	600
14	Batik Priangan	25	5	0	570	600
15	Batik Sekar	30	0	0	570	600
16	Batik Sidoluhur	28	2	2	568	600
17	Batik Sidomukti	22	8	3	567	600
18	Batik Sogan	30	0	1	569	600
19	Batik Tambal	30	0	0	570	600

Berdasarkan confusion matrix pada tabel.. dapat dilihat bahwa proses training menggunakan algoritma Convolutional Neural Network tanpa menggunakan ResNet berhasil mengklasifikasikan dengan benar sebanyak 568 gambar. Berdasarkan hasil tersebut didapat hasil Batik Sidomukti yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Sidomukti hanya mendapatkan 22 gambar dari 30 gambar.

```

19/19 [=====] - 8s 387ms/step
Prediction Accuracy : 0.9466666666666667
precision  recall  f1-score  support
0          0.93    0.83    0.88    30
1          0.97    0.97    0.97    30
2          1.00    1.00    1.00    30
3          1.00    1.00    1.00    30
4          0.93    0.87    0.90    30
5          0.90    0.93    0.92    30
6          0.86    1.00    0.92    30
7          1.00    0.97    0.98    30
8          0.88    0.93    0.90    30
9          0.88    0.97    0.92    30
10         1.00    1.00    1.00    30
11         1.00    1.00    1.00    30
12         0.94    1.00    0.97    30
13         0.91    0.97    0.94    30
14         1.00    0.83    0.91    30
15         1.00    1.00    1.00    30
16         0.93    0.93    0.93    30
17         0.88    0.73    0.80    30
18         0.97    1.00    0.98    30
19         1.00    1.00    1.00    30

accuracy              0.95    600
macro avg             0.95    0.95    0.95    600
weighted avg          0.95    0.95    0.95    600
    
```

Berdasarkan hasil accuracy pada gambar... dapat dilihat bahwa proses training menggunakan algoritma Convolutional Neural Network tanpa menggunakan ResNet memperoleh skor sebesar 95%.

**6.2 Skenario 2**



No	Kelas	Klasifikasi				Jumlah
		TP	FP	FN	TN	
0	Batik Bali	22	8	8	562	600
1	Batik Betawi	29	1	22	548	600
2	Batik Celup	21	3	9	567	600
3	Batik Cendrawasih	18	12	13	557	600
4	Batik Ceplok	17	13	4	566	600
5	Batik Ciamis	7	23	10	560	600
6	Batik Garutan	24	6	13	557	600
7	Batik Gentongan	16	14	10	560	600
8	Batik Kawung	29	1	6	564	600
9	Batik Keraton	17	13	7	563	600
10	Batik Lasem	21	9	2	568	600
11	Batik Megamendung	30	0	4	566	600
12	Batik Parang	28	2	6	564	600
13	Batik Pekalongan	15	15	5	565	600
14	Batik Priangan	10	20	9	561	600
15	Batik Sekar	23	7	33	537	600
16	Batik Sidoluhur	21	9	5	565	600
17	Batik Sidomukti	14	16	4	566	600
18	Batik Sogan	20	10	26	544	600
19	Batik Tambal	20	10	8	562	600

Berdasarkan confusion matrix pada tabel... dapat dilihat bahwa proses training tanpa menggunakan data preprocessing berhasil mengklasifikasikan dengan benar hanya sebanyak 384 gambar. Berdasarkan hasil tersebut didapat hasil Batik Sekar yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Sekar hanya mendapatkan 11 gambar dari 30 gambar.

Pengujian ini dilakukan untuk mengukur presisi, sensitifitas, dan keakuratan dari data uji dan

data latihan terhadap hasil pengujian. Pengujian dilakukan dengan membandingkan antara hasil prediksi dengan hasil klasifikasi. Dari hasil confusion matrix ini bisa kita klasifikasikan sebagai berikut :

No	Kelas	Klasifikasi				Jumlah
		TP	FP	FN	TN	
0	Batik Bali	17	13	2	568	600
1	Batik Betawi	24	6	20	550	600
2	Batik Celup	21	1	9	569	600
3	Batik Cendrawasih	16	14	15	555	600
4	Batik Ceplok	12	18	5	565	600
5	Batik Ciamis	7	23	2	568	600
6	Batik Garutan	19	11	3	567	600
7	Batik Gentongan	21	9	15	555	600
8	Batik Kawung	30	0	9	561	600
9	Batik Keraton	13	17	6	564	600
10	Batik Lasem	27	3	4	566	600
11	Batik Megamendung	30	0	1	569	600
12	Batik Parang	29	1	6	564	600
13	Batik Pekalongan	23	7	12	558	600
14	Batik Priangan	11	19	7	563	600
15	Batik Sekar	6	24	5	565	600
16	Batik Sidoluhur	23	7	10	560	600
17	Batik Sidomukti	20	10	11	559	600
18	Batik Sogan	15	15	78	492	600
19	Batik Tambal	20	10	4	566	600

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

No	Kelas	Hasil			
		Accuracy	Recall	Precision	F1-Score
0	Batik Bali	0,97	0,73	0,73	0,73
1	Batik Betawi	0,96	0,97	0,57	0,72

2	Batik Celup	0,98	0,88	0,70	0,78
3	Batik Cendrawasih	0,96	0,60	0,58	0,59
4	Batik Ceplok	0,97	0,57	0,81	0,67
5	Batik Ciamis	0,95	0,23	0,41	0,30
6	Batik Garutan	0,97	0,80	0,65	0,72
7	Batik Gentongan	0,96	0,53	0,62	0,57
8	Batik Kawung	0,99	0,97	0,83	0,89
9	Batik Keraton	0,97	0,57	0,71	0,63
10	Batik Lasem	0,98	0,70	0,91	0,79
11	Batik Megamendung	0,99	1,00	0,88	0,94
12	Batik Parang	0,99	0,93	0,82	0,88
13	Batik Pekalongan	0,97	0,50	0,75	0,60
14	Batik Priangan	0,95	0,33	0,53	0,41
15	Batik Sekar	0,93	0,77	0,41	0,53
16	Batik Sidoluhur	0,98	0,70	0,81	0,75
17	Batik Sidomukti	0,97	0,47	0,78	0,58
18	Batik Sogan	0,94	0,67	0,43	0,53
19	Batik Tambal	0,97	0,67	0,71	0,69

Berdasarkan confusion matrix pada tabel dibawah dapat dilihat bahwa proses training tanpa menggunakan data preprocessing berhasil mengklasifikasikan dengan benar hanya sebanyak 384 gambar. Berdasarkan hasil tersebut didapat hasil Batik Sekar yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Sekar hanya mendapatkan 11 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 :

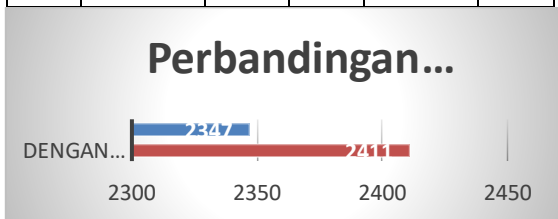
No	Kelas	Klasifikasi			
		TP	FP	FN	TN
0	Batik Bali	17	13	2	568
1	Batik Betawi	24	6	20	550
2	Batik Celup	21	1	9	569

3	Batik Cendrawasih	16	14	15	555
4	Batik Ceplok	12	18	5	565
5	Batik Ciamis	7	23	2	568
6	Batik Garutan	19	11	3	567
7	Batik Gentongan	21	9	15	555
8	Batik Kawung	30	0	9	561
9	Batik Keraton	13	17	6	564
10	Batik Lasem	27	3	4	566
11	Batik Megamendung	30	0	1	569
12	Batik Parang	29	1	6	564
13	Batik Pekalongan	23	7	12	558
14	Batik Priangan	11	19	7	563
15	Batik Sekar	6	24	5	565
16	Batik Sidoluhur	23	7	10	560
17	Batik Sidomukti	20	10	11	559
18	Batik Sogan	15	15	78	492
19	Batik Tambal	20	10	4	566

Berdasarkan data klasifikasi pada Tabel 6.4 Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

No	Kelas	Hasil			
		Accuracy	Recall	Precision	F1-Score
0	Batik Bali	0,98	0,57	0,89	0,69
1	Batik Betawi	0,96	0,80	0,55	0,65
2	Batik Celup	0,98	0,95	0,70	0,81
3	Batik Cendrawasih	0,95	0,53	0,52	0,52
4	Batik Ceplok	0,96	0,40	0,71	0,51
5	Batik Ciamis	0,96	0,23	0,78	0,36
6	Batik Garutan	0,98	0,63	0,86	0,73
7	Batik Gentongan	0,96	0,70	0,58	0,64
8	Batik Kawung	0,99	1,00	0,77	0,87
9	Batik Keraton	0,96	0,43	0,68	0,53
10	Batik Lasem	0,99	0,90	0,87	0,89
11	Batik Megamendung	1,00	1,00	0,97	0,98
12	Batik Parang	0,99	0,97	0,83	0,89

13	Batik Pekalongan	0,97	0,77	0,66	0,71
14	Batik Priangan	0,96	0,37	0,61	0,46
15	Batik Sekar	0,95	0,20	0,55	0,29
16	Batik Sidoluhur	0,97	0,77	0,70	0,73
17	Batik Sidomukti	0,97	0,67	0,65	0,66
18	Batik Sogan	0,85	0,50	0,16	0,24
19	Batik Tambal	0,98	0,67	0,83	0,74



Pada gambar 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.

Skenario	Accuracy pada Sistem	Accuracy pada perhitungan manual	Precision	Recall	F1-score
Tanpa preprocessing	0.64	0.967	0.71	0.64	0.65
Dengan preprocessing	0.67	0.964	0.69	0.67	0.66

Tabel menunjukkan bahwa kedua percobaan memiliki nilai accuracy yang sudah baik tapi untuk precision, recall dan f1-score masih mendapatkan hasil yang kurang baik dibawah 0.9. Namun dari kedelapan nilai dari percobaan pelatihan menggunakan data preprocessing tetap lebih tinggi dibandingkan dengan percobaan pelatihan tanpa menggunakan data preprocessing.

### 6.3 Skenario 3

Berdasarkan confusion matrix pada tabel dibawah dapat dilihat bahwa proses training menggunakan Optimizer RMSprop dengan learning rate 0.0001 berhasil mengklasifikasikan dengan benar sebanyak 503 gambar. Berdasarkan hasil tersebut

didapat hasil Batik Sogan yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Sogan hanya mendapatkan 18 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:

No	Kelas	Klasifikasi			
		TP	FP	FN	TN
0	Batik Bali	26	4	14	556
1	Batik Betawi	30	0	12	558
2	Batik Celup	30	3	0	567
3	Batik Cendrawasih	27	3	3	567
4	Batik Ceplok	23	7	2	568
5	Batik Ciamis	22	8	8	562
6	Batik Garutan	25	5	6	564
7	Batik Gentongan	22	8	2	568
8	Batik Kawung	29	1	5	565
9	Batik Keraton	24	6	6	564
10	Batik Lasem	28	2	4	566
11	Batik Megamendung	30	0	2	568
12	Batik Parang	29	1	8	562
13	Batik Pekalongan	20	10	1	569
14	Batik Priangan	20	10	2	568
15	Batik Sekar	29	1	2	568
16	Batik Sidoluhur	25	5	2	568
17	Batik Sidomukti	22	8	10	560
18	Batik Sogan	18	12	1	569
19	Batik Tambal	24	6	4	566

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

No	Kelas	Hasil			
		Accuracy	Recall	Precision	F1-Score
0	Batik Bali	0,97	0,87	0,65	0,74
1	Batik Betawi	0,98	1,00	0,71	0,83
2	Batik Celup	1,00	0,91	1,00	0,95
3	Batik Cendrawasih	0,99	0,90	0,90	0,90
4	Batik Ceplok	0,99	0,77	0,92	0,84
5	Batik Ciamis	0,97	0,73	0,73	0,73
6	Batik Garutan	0,98	0,83	0,81	0,82
7	Batik Gentongan	0,98	0,73	0,92	0,81
8	Batik Kawung	0,99	0,97	0,85	0,91
9	Batik Keraton	0,98	0,80	0,80	0,80
10	Batik Lasem	0,99	0,93	0,88	0,90
11	Batik Megamendung	1,00	1,00	0,94	0,97
12	Batik Parang	0,99	0,97	0,78	0,87
13	Batik Pekalongan	0,98	0,67	0,95	0,78
14	Batik Priangan	0,98	0,67	0,91	0,77
15	Batik Sekar	1,00	0,97	0,94	0,95
16	Batik Sidoluhur	0,99	0,83	0,93	0,88
17	Batik Sidomukti	0,97	0,73	0,69	0,71
18	Batik Sogan	0,98	0,60	0,95	0,73
19	Batik Tambal	0,98	0,80	0,86	0,83

Berdasarkan confusion matrix pada tabel dibawah dapat dilihat bahwa proses training menggunakan Optimizer Adam dengan learning rate 0.0001 berhasil mengklasifikasikan dengan benar sebanyak 531 gambar. Berdasarkan hasil tersebut didapat hasil Batik Priangan yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Priangan hanya mendapatkan 20 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:

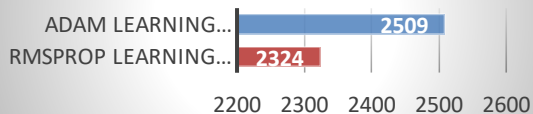
No	Kelas	Klasifikasi			
		TP	FN	FP	TN
0	Batik Bali	24	6	4	566
1	Batik Betawi	30	0	3	567
2	Batik Celup	30	1	0	569
3	Batik Cendrawasih	29	1	0	570
4	Batik Ceplok	22	8	1	569
5	Batik Ciamis	27	3	9	561
6	Batik Garutan	27	3	2	568
7	Batik Gentongan	26	4	4	566
8	Batik Kawung	29	1	5	565
9	Batik Keraton	25	5	3	567
10	Batik Lasem	29	1	2	568
11	Batik Megamendung	30	0	1	569
12	Batik Parang	26	4	3	567
13	Batik Pekalongan	26	4	4	566
14	Batik Priangan	20	10	3	567
15	Batik Sekar	30	0	5	565
16	Batik Sidoluhur	26	4	4	566
17	Batik Sidomukti	23	7	6	564
18	Batik Sogan	25	5	4	566
19	Batik Tambal	27	3	5	565

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

No	Kelas	Hasil			
		Accuracy	Recall	Precision	F1-Score
0	Batik Bali	0,98	0,80	0,86	0,83
1	Batik Betawi	1,00	1,00	0,91	0,95
2	Batik Celup	1,00	0,97	1,00	0,98

3	Batik Cendrawasih	1,00	0,97	1,00	0,98
4	Batik Ceplok	0,99	0,73	0,96	0,83
5	Batik Ciamis	0,98	0,90	0,75	0,82
6	Batik Garutan	0,99	0,90	0,93	0,92
7	Batik Gentongan	0,99	0,87	0,87	0,87
8	Batik Kawung	0,99	0,97	0,85	0,91
9	Batik Keraton	0,99	0,83	0,89	0,86
10	Batik Lasem	1,00	0,97	0,94	0,95
11	Batik Megamendung	1,00	1,00	0,97	0,98
12	Batik Parang	0,99	0,87	0,90	0,88
13	Batik Pekalongan	0,99	0,87	0,87	0,87
14	Batik Priangan	0,98	0,67	0,87	0,75
15	Batik Sekar	0,99	1,00	0,86	0,92
16	Batik Sidaluhur	0,99	0,87	0,87	0,87
17	Batik Sidomukti	0,98	0,77	0,79	0,78
18	Batik Sogan	0,99	0,83	0,86	0,85
19	Batik Tambal	0,99	0,90	0,84	0,87

**Perbandingan...**



Pada gambar diatas menunjukkan bahwa pelatihan menggunakan optimizer RMSprop dengan learning rate 0.0001 mendapatkan waktu yang lebih

cepat daripada menggunakan optimizer Adam dengan learning rate 0.0001.

Skenario	Accuracy pada Sistem	Accuracy pada perhitungan manual	Precision	Recall	F1-score
RMSprop (0.0001)	0.84	0,983833	0.85	0.84	0.84
Adam (0.0001)	0.89	0,9885	0.89	0.89	0.88

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 yang hampir di angka 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 0.8, dimana ini merupakan hasil pelatihan yang baik. Perbandingan ini bisa dilihat pada tabel diatas.

**6.4 Skenario 4**

Berdasarkan confusion matrix pada tabel dibawah dapat dilihat bahwa proses training menggunakan Resnet50 berhasil mengklasifikasikan dengan benar sebanyak 531 gambar. Berdasarkan hasil tersebut didapat hasil Batik Sidomukti yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Sidomukti hanya mendapatkan 18 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:

No	Kelas	Klasifikasi			
		TP	FP	FN	TN
0	Batik Bali	22	8	3	567
1	Batik Betawi	29	1	6	564
2	Batik Celup	30	2	0	568



3	Batik Cendrawasih	30	0	11	559
4	Batik Ceplok	25	5	1	569
5	Batik Ciamis	26	4	1	569
6	Batik Garutan	29	1	3	567
7	Batik Gentongan	21	9	0	570
8	Batik Kawung	29	1	6	564
9	Batik Keraton	25	5	6	564
10	Batik Lasem	29	1	0	570
11	Batik Megamendung	30	0	5	565
12	Batik Parang	26	4	3	567
13	Batik Pekalongan	25	5	1	569
14	Batik Priangan	26	4	1	569
15	Batik Sekar	30	0	4	566
16	Batik Sidoluhur	27	3	2	568
17	Batik Sidomukti	18	12	2	568
18	Batik Sogan	24	6	3	567
19	Batik Tambal	30	0	9	561

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

No	Kelas	Hasil			
		Accuracy	Recall	Precision	F1-Score
0	Batik Bali	0,98	0,73	0,88	0,80
1	Batik Betawi	0,99	0,97	0,83	0,89
2	Batik Celup	1,00	0,94	1,00	0,97
3	Batik Cendrawasih	0,98	1,00	0,73	0,85
4	Batik Ceplok	0,99	0,83	0,96	0,89
5	Batik Ciamis	0,99	0,87	0,96	0,91
6	Batik Garutan	0,99	0,97	0,91	0,94
7	Batik Gentongan	0,99	0,70	1,00	0,82

8	Batik Kawung	0,99	0,97	0,83	0,89
9	Batik Keraton	0,98	0,83	0,81	0,82
10	Batik Lasem	1,00	0,97	1,00	0,98
11	Batik Megamendung	0,99	1,00	0,86	0,92
12	Batik Parang	0,99	0,87	0,90	0,88
13	Batik Pekalongan	0,99	0,83	0,96	0,89
14	Batik Priangan	0,99	0,87	0,96	0,91
15	Batik Sekar	0,99	1,00	0,88	0,94
16	Batik Sidoluhur	0,99	0,90	0,93	0,92
17	Batik Sidomukti	0,98	0,60	0,90	0,72
18	Batik Sogan	0,99	0,80	0,89	0,84
19	Batik Tambal	0,99	1,00	0,77	0,87

Berdasarkan confusion matrix pada tabel dibawah dapat dilihat bahwa proses training menggunakan ResNet101 berhasil mengklasifikasikan dengan benar sebanyak 494 gambar. Berdasarkan hasil tersebut didapat hasil Batik Ciamis yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Ciamis hanya mendapatkan 18 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 :

No	Kelas	Klasifikasi			
		TP	FN	FP	TN
0	Batik Bali	26	5	25	544
1	Batik Betawi	26	4	17	553
2	Batik Celup	20	4	10	566
3	Batik Cendrawasih	22	8	2	568
4	Batik Ceplok	23	7	6	564
5	Batik Ciamis	24	6	23	547
6	Batik Garutan	19	11	6	564

7	Batik Gentongan	18	11	4	567
8	Batik Kawung	26	4	6	564
9	Batik Keraton	19	11	0	570
10	Batik Lasem	10	20	0	570
11	Batik Megamendung	16	14	1	569
12	Batik Parang	27	3	29	541
13	Batik Pekalongan	20	10	3	567
14	Batik Priangan	16	14	2	568
15	Batik Sekar	19	11	0	570
16	Batik Sidaluhur	20	10	9	561
17	Batik Sidomukti	22	8	2	568
18	Batik Sogan	24	6	4	566
19	Batik Tambal	28	2	32	538

Berdasarkan data klasifikasi pada tabel di atas Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

No	Kelas	Hasil			
		Accuracy	Recall	Precision	F1-Score
0	Batik Bali	0,95	0,84	0,51	0,63
1	Batik Betawi	0,97	0,87	0,60	0,71
2	Batik Celup	0,98	0,83	0,67	0,74
3	Batik Cendrawasih	0,98	0,73	0,92	0,81
4	Batik Ceplok	0,98	0,77	0,79	0,78
5	Batik Ciamis	0,95	0,80	0,51	0,62
6	Batik Garutan	0,97	0,63	0,76	0,69
7	Batik Gentongan	0,98	0,62	0,82	0,71
8	Batik Kawung	0,98	0,87	0,81	0,84
9	Batik Keraton	0,98	0,63	1,00	0,78
10	Batik Lasem	0,97	0,33	1,00	0,50
11	Batik Megamendung	0,98	0,53	0,94	0,68

12	Batik Parang	0,95	0,90	0,48	0,63
13	Batik Pekalongan	0,98	0,67	0,87	0,75
14	Batik Priangan	0,97	0,53	0,89	0,67
15	Batik Sekar	0,98	0,63	1,00	0,78
16	Batik Sidaluhur	0,97	0,67	0,69	0,68
17	Batik Sidomukti	0,98	0,73	0,92	0,81
18	Batik Sogan	0,98	0,80	0,86	0,83
19	Batik Tambal	0,94	0,93	0,47	0,62

Berdasarkan confusion matrix pada tabel dibawah dapat dilihat bahwa proses training menggunakan ResNet152 berhasil mengklasifikasikan dengan benar sebanyak 490 gambar. Berdasarkan hasil tersebut didapat hasil Batik Priangan yang mendapatkan jumlah terendah dalam pengklasifikasian menggunakan model ini. Batik Ciamis hanya mendapatkan 13 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 :

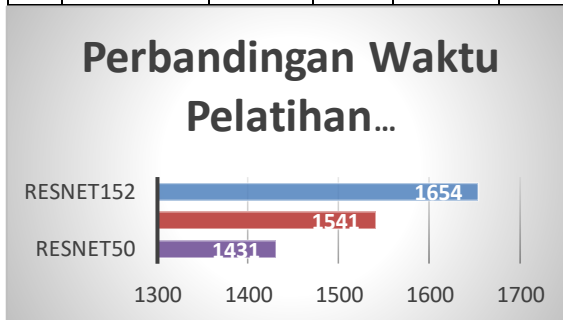
No	Kelas	Klasifikasi			
		TP	FN	FP	TN
0	Batik Bali	21	9	2	568
1	Batik Betawi	26	4	1	569
2	Batik Celup	24	0	6	570
3	Batik Cendrawasih	28	2	3	567
4	Batik Ceplok	23	7	4	566
5	Batik Ciamis	22	8	1	569
6	Batik Garutan	28	2	14	556
7	Batik Gentongan	21	9	7	563
8	Batik Kawung	24	6	2	568
9	Batik Keraton	27	3	16	554
10	Batik Lasem	29	1	6	564

11	Batik Megamendung	30	0	2	568
12	Batik Parang	28	2	4	566
13	Batik Pekalongan	28	2	9	561
14	Batik Priangan	13	17	3	567
15	Batik Sekar	22	8	15	555
16	Batik Sidoluhur	27	3	9	561
17	Batik Sidomukti	19	11	1	569
18	Batik Sogan	26	4	10	560
19	Batik Tambal	24	6	1	569

Berdasarkan data klasifikasi pada tabel di atas Klasifikasi Data dapat dihitung nilai presisi, sensitifitas, dan akurasinya sebagai berikut:

No	Kelas	Hasil			
		Accuracy	Recall	Precision	F1-Score
0	Batik Bali	0,98	0,70	0,91	0,79
1	Batik Betawi	0,99	0,87	0,96	0,91
2	Batik Celup	0,99	1,00	0,80	0,89
3	Batik Cendrawasih	0,99	0,93	0,90	0,92
4	Batik Ceplok	0,98	0,77	0,85	0,81
5	Batik Ciamis	0,99	0,73	0,96	0,83
6	Batik Garutan	0,97	0,93	0,67	0,78
7	Batik Gentongan	0,97	0,70	0,75	0,72
8	Batik Kawung	0,99	0,80	0,92	0,86
9	Batik Keraton	0,97	0,90	0,63	0,74
10	Batik Lasem	0,99	0,97	0,83	0,89
11	Batik Megamendung	1,00	1,00	0,94	0,97
12	Batik Parang	0,99	0,93	0,88	0,90
13	Batik Pekalongan	0,98	0,93	0,76	0,84
14	Batik Priangan	0,97	0,43	0,81	0,57
15	Batik Sekar	0,96	0,73	0,59	0,66

16	Batik Sidoluhur	0,98	0,90	0,75	0,82
17	Batik Sidomukti	0,98	0,63	0,95	0,76
18	Batik Sogan	0,98	0,87	0,72	0,79
19	Batik Tambal	0,99	0,80	0,96	0,87



Pada gambar diatas menunjukkan bahwa pelatihan menggunakan model ResNet50 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 ResNet50 hingga ResNet152 menunjukkan bahwa percobaan menggunakan model ResNet50 memiliki nilai yang lebih tinggi daripada model yang lainnya. Perbandingan ini bisa dilihat pada tabel berikut.

Model	Accuracy	Precision	Recall	F1-score
ResNet50	0.89	0.89	0.89	0.88
ResNet101	0.82	0.83	0.82	0.82
ResNet152	0.82	0.84	0.82	0.82

Pada jaringan ResNet 50, 101 dan 152 layer, terdapat modifikasi pada blok bangunan koneksi pintas. Modifikasi tersebut adalah setiap fungsi residual ditumpuk 3 lapisan konvolusi dengan ukuran 1x1, 3x3, dan 1x1. Modifikasi ini disebut Deeper Bottleneck architecture, dimana lapisan terakhir 1x1 yang responsible untuk mengurangi dan meningkatkan dimensi dengan meninggikan lapisan konvolusi 3x3 input/output yang lebih kecil. Pada ResNet 50, 101, dan 152 layer digunakan pintasan identitas yang bebas parameter (parameter-free identity shortcut).

Pada ResNet 50 layer, untuk meningkatkan dimensi menggunakan projection shortcut pada pintasan identitas (identity shortcut) dan mengubah 2 layer block pada ResNet 34 layer dengan 3 layer bottleneck block. Sedangkan pada ResNet 101 dan

152 layer menggunakan lebih dari 3 layer block. Meskipun ResNet 152 layer kedalamannya meningkat secara signifikan, namun masih mempunyai kompleksitas yang lebih rendah dari ResNet50. Model yang dibuat menggunakan hasil ekstraksi fitur gambar dari ResNet50. Nilai input model diambil dari vektor terakhir dari proses ekstraksi fitur.

Meskipun arsitektur ResNet101 dan ResNet152 lebih kompleks dari ResNet50 namun jika dilihat dari hasil evaluasi model, ResNet101 memiliki performa yang lebih buruk. Hal ini diakibatkan oleh data latih yang relatif kecil apabila dilatih dengan arsitektur yang besar seperti ResNet-101, belum tentu menghasilkan akurasi yang lebih baik.

## 7. Kesimpulan

Dalam penelitian skripsi berjudul “Implementasi Image Classification Menggunakan Arsitektur Convolutional Neural Network”, peneliti telah berhasil merancang dan membangun model pelatihan sesuai dengan rancangan. Berikut adalah beberapa poin penting yang dapat disimpulkan dari penelitian ini:

1. Penelitian ini menggunakan arsitektur Convolutional Neural Network dan berhasil mencapai hasil sebesar 95%.
2. Model pelatihan yang mendapatkan akurasi tertinggi adalah model ResNet50 dengan optimizer RMSprop dan learning rate sebesar 0.0001, yang mencapai akurasi sebesar 88%.
3. Model paling optimal adalah model ResNet50 dengan optimizer RMSprop dan learning rate sebesar 0.0001. Model ini berhasil mengklasifikasikan 531 dari 600 gambar batik dengan kelas yang benar dalam waktu pelatihan selama 5458 detik.
4. Ditemukan bahwa semakin kecil nilai learning rate, semakin kecil kemungkinan terjadinya overfitting dan semakin baik akurasi yang didapatkan.
5. Ditemukan juga bahwa semakin tinggi penggunaan model pelatihan ResNet, semakin lama durasi yang dibutuhkan saat pelatihan, dan akurasi yang didapatkan tidak cukup baik. Hal ini disebabkan oleh jumlah layer yang semakin banyak.

Dengan demikian, penelitian ini memberikan kontribusi penting dalam bidang klasifikasi citra menggunakan arsitektur Convolutional Neural Network, khususnya dalam konteks klasifikasi gambar batik. Kesimpulan ini dapat menjadi referensi penting bagi penelitian selanjutnya dalam bidang ini.

## 8. Kesimpulan

Berikut adalah beberapa saran untuk pengembangan lebih lanjut dari penelitian yang

berjudul “Implementasi Image Classification Menggunakan Arsitektur Convolutional Neural Network”:

1. Menambah Citra pada Dataset: Menambahkan lebih banyak citra pada dataset di setiap kelas dapat membantu meningkatkan akurasi model. Semakin banyak citra yang diproses oleh mesin, semakin baik mesin tersebut dalam mempelajari dan mengklasifikasikan citra tersebut.
2. Merancang Layer Tambahan yang Lebih Sederhana: Merancang tambahan layer yang lebih sederhana dapat membantu mempercepat proses pelatihan. Dengan layer yang lebih sederhana, waktu yang dibutuhkan untuk pelatihan dapat dikurangi tanpa mengorbankan akurasi model.

### Daftar Pustaka

- Anggraini, W. (n.d.). DEEP LEARNING UNTUK DETEKSI WAJAH YANG BERHIJAB MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN TENSORFLOW. 2020.
- Falahkhi, B., Achmal, E. F., Rizaldi, M., Rizki, R., & Yudistira, N. (n.d.). Perbandingan Model AlexNet dan ResNet dalam Klasifikasi Citra Bunga Memanfaatkan Transfer Learning Comparison of AlexNet and ResNet Models in Flower Image Classification Utilizing Transfer Learning. 2022. <http://journal.ipb.ac.id/index.php/jika>
- Hadianto, N., Novitasari, H. B., & Rahmawati, A. (2019). KLASIFIKASI PEMINJAMAN NASABAH BANK MENGGUNAKAN METODE NEURAL NETWORK. *Jurnal Pilar Nusa Mandiri*, 15(2), 163–170. <https://doi.org/10.33480/pilar.v15i2.658>
- Hasan Mahmud, K., & al Faraby, S. (2019). Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network.
- Homepage, J., Roihan, A., Abas Sunarya, P., & Rafika, A. S. (2019). IJCIT (Indonesian Journal on Computer and Information Technology) Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. In *IJCIT (Indonesian Journal on Computer and Information Technology)* (Vol. 5, Issue 1).
- Kholik, A. (2021). KLASIFIKASI MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA TANGKAPAN LAYAR HALAMAN INSTAGRAM. *JDMSI*, 2(2), 10–20.
- Kusumanto, R. D., & Tomponu, A. N. (2011). PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB. In *Seminar Nasional Teknologi Informasi & Komunikasi Terapan*.
- Mawan, R. (2020). Klasifikasi motif batik menggunakan Convolutional Neural Network. *Jnanaloka*, 45–50. <https://doi.org/10.36802/jnanaloka.2020.v1-no1-45-50>
- Pengolahan..., A., Zaid Munantri, N., Sofyan, H., & Yanu, M. (2019). APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK IDENTIFIKASI UMUR POHON. In *TELEMATIKA* (Vol. 16, Issue 2).
- Triyani, E., Harsono, D., & Sulistyowati, R. (2019). IJCIT (Indonesian Journal on Computer and Information Technology). In *IJCIT (Indonesian Journal on Computer and Information Technology)* (Vol. 5, Issue 1).
- Tumewu, S. F., Setiabud, D. H., & Sugiarto, I. (2020). Klasifikasi Motif Batik menggunakan Metode Deep Convolutional Neural Network dengan Data Augmentation. *Jurnal Infra*, 8(2), 189–194.
- Wei, M., Wu, Q., Ji, H., Wang, J., Lyu, T., Liu, J., & Zhao, L. (2023). A Skin Disease Classification Model Based on DenseNet and ConvNeXt Fusion. *Electronics (Switzerland)*, 12(2). <https://doi.org/10.3390/electronics12020438>