

Implementasi Algoritma *Convolutional Neural Networks* (CNN) Untuk Klasifikasi Batik

Fuadz Hasyim¹, Kamil Malik², Fathur Rizal³, Yudistira⁴
^{1,2,3,4} Universitas Nurul Jadid, Probolinggo

Info Artikel	ABSTRAK
<p><u>Riwayat Artikel</u> Diterima: 10-11-2021 Disetujui: 19-11-2021</p> <p><u>Kata Kunci</u> Batik; <i>Convolutional Neural Networks</i>; Deep Learning;</p> <p>e-mail*</p> <p>*fathurrizal1993@gmail.com</p>	<p>Batik adalah salah satu budaya khas Indonesia dan sudah diakui sebagai warisan budaya Internasional oleh UNESCO (The United Nations Educational, Scientific and Cultural Organization) pada tanggal 2 Oktober 2009. Batik telah menjadi warisan budaya turun temurun di seluruh Indonesia khususnya di daerah Jawa. Saat ini ada ratusan motif kain batik dari seluruh penjuru Indonesia. Banyaknya pola batik di Indonesia mengakibatkan sulitnya masyarakat mengidentifikasi motif pada batik. Penelitian ini dapat mempermudah pengenalan pola batik. Salah satu teknologi kecerdasan buatan dengan sebutan artificial intelligence (AI) adalah pembelajaran mesin dengan menggunakan metode computer vision. Salah satu model pembelajaran mesin tersebut adalah jaringan syaraf tiruan (JST) dengan menggunakan banyak lapisan, sehingga dengan adanya model tersebut maka dapat lebih baik lagi performa komputasi dengan menggunakan teknik Deep Learning. Metode yang digunakan adalah <i>Convolutional Neural Networks</i> (CNN) dengan melakukan klasifikasi gambar pada batik berbasis komputer dengan memanfaatkan kecerdasan buatan (artificial intelligence). Hasil dari penelitian yang telah dilakukan pada pengujian terhadap 200 dataset dan 20 label diperoleh nilai akurasi yang tertinggi adalah “Batik Megamendung dan Batik Celup” dengan nilai akurasi 80% dan 60%, hasil accuracy yang diperoleh dari proses pelatihan model dari 200 epoch yang tertinggi adalah 90%.</p>

1. PENDAHULUAN

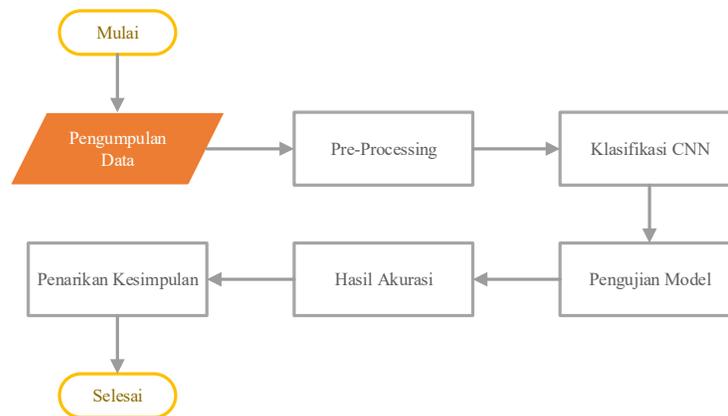
Batik adalah salah satu budaya khas Indonesia dan sudah diakui sebagai warisan budaya Internasional oleh UNESCO (*The United Nations Educational, Scientific and Cultural Organization*) pada tanggal 2 Oktober 2009[1]. Batik telah menjadi warisan budaya turun temurun di seluruh Indonesia khususnya di daerah Jawa[2]. Secara etimologis, Batik merupakan singkatan dari dua suku kata yang diambil dari kata *Amba* dan *Titik*. *Amba* yang berartikan menulis dan *Titik* ialah titik, jadi kepanjangan dari batik berarti menulis titik diatas kain. Nilai dalam budaya dan seni yang sangat tinggi, sehingga menjadikan batik sebagai produk bernilai ekonomis untuk era saat ini. Motif dan beragam hias batik lahir dan diciptakan dari proses kognitif manusia yang didapat dari alam dan sekitarnya. Definisi lain juga menjelaskan bahwa batik merupakan bentuk seni visual Indonesia yang menggunakan teknik menggambar tradisional pada kain[3]. Saat ini ada ratusan motif kain batik dari seluruh penjuru Indonesia, mulai dari batik tulis hingga batik digital (*print*). Banyaknya pola batik di Indonesia mengakibatkan sulitnya masyarakat mengetahui motif atau pola suatu batik.

Oleh sebab itu, penelitian ini mencoba mengidentifikasi suatu pola pada batik dengan menggunakan algoritma yang digunakan dalam kecerdasan buatan (*artificial intelligence*)[4]. Algoritma yang akan digunakan untuk mengidentifikasi atau klasifikasi batik dalam penelitian

ini menggunakan *Convolutional Neural Network* (CNN). CNN adalah algoritma *machine learning* yang merupakan pengembangan dari *Multilayer Perceptron* (MLP)[5]. Algoritma ini termasuk dalam algoritma *Deep Learning* yang dapat melakukan klasifikasi suatu gambar maupun suara. Algoritma CNN digunakan dalam penelitian ini karena dapat menangani permasalahan pada betuk gambar seperti rotasi, skala dan translasi. Algoritma CNN dalam penelitian ini digunakan untuk melakukan klasifikasi terhadap 20 jenis citra batik yang diambil dari <https://www.kaggle.com/dionisiusdh/indonesian-batik-motifs>.

2. METODE

Dataset citra batik yang digunakan dalam penelitian ini menggunakan dataset *public* yang diperoleh dari laman <https://www.kaggle.com/dionisiusdh/indonesian-batik-motifs>. Dataset citra batik yang dikumpulkan sebanyak 983 dari 20 jenis batik yang digunakan dalam penelitian ini. *Tools* yang digunakan untuk melakukan uji coba algoritma CNN terhadap data yang telah diperoleh menggunakan *Google Colab*. Tahapan dalam penelitian ini meliputi pengumpulan data, *pre-processing*, klasifikasi serta pengujian model CNN dan hasil akurasi klasifikasi citra. Beberapa tahapan tersebut dapat digambarkan seperti pada gambar berikut:



Gambar 1. Kerangka Alur Penelitian

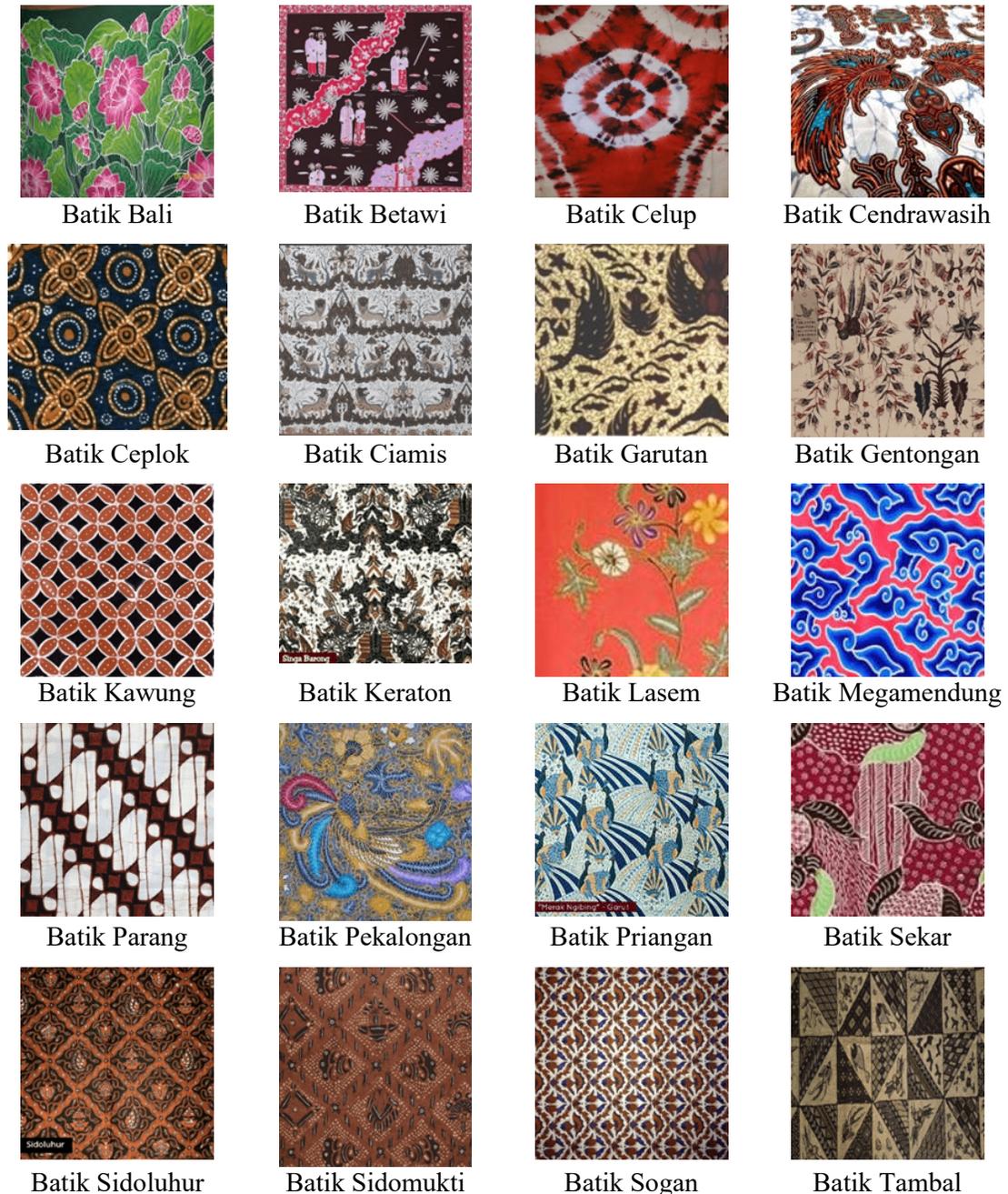
Tahap *pre-processing* digunakan untuk menyiapkan data citra batik agar dapat digunakan secara optimal. Tahap *pre-processing* ini dibagi menjadi beberapa bagian seperti membaca dataset yang sudah diupload pada *google drive* dan menyamakan resolusi citra batik dengan melakukan *resize* pada citra batik yang telah di unggah hal ini bertujuan agar citra batik yang digunakan memiliki resolusi yang sama[6]. Setelah dilakukan *resize* citra selanjutnya dilakukan normalisasi data sebelum dilakukan klasifikasi citra batik dan pengujian model.

Dataset citra batik yang telah dikumpulkan selanjutnya dibagi menjadi 2 kategori pada masing-masing citra batik, yakni citra yang digunakan sebagai data *training* dan data *testing*. Setelah dilakukan pengelompokan tersebut selanjutnya dilakukan pengujian model algoritma CNN pada citra batik yang telah di kelompok untuk memperoleh model serta hasil akurasi yang diperoleh dari algoritma CNN yang selanjutnya dapat dijadikan kesimpulan dari uji coba yang telah dilakukan.

3. HASIL DAN PEMBAHASAN

Dataset yang digunakan pada penelitian ini adalah citra Batik Indonesia. Data gambar tersebut diperoleh dengan cara mengunduh dari link yang telah disebut diatas. Jumlah citra yang

terdapat pada dataset adalah 931 gambar dengan format jpg, dengan resolusi yang berbeda-beda, yang terbagi menjadi 20 kelas yakni sebagai berikut:



Gambar 2. Kumpulan Dataset Citra Batik

Proses selanjutnya adalah memanggil dataset yang telah di upload ke *google drive*. Fungsi dari pemanggilan data ini ialah agar dapat mengetahui data apa saja yang akan kita olah atau kita proses pada metode *Convolutional Neural Networks (CNN)*. Kode memanggil dataset terlihat pada baris kode berikut.

```
# memasukan variabel yang sudah dikumpulkan pada looping di atas menjadi sebuah dataframe agar rapih  
df = pd.DataFrame({"path":full_path, 'file_name':file_name, "tag":tag})  
df.groupby(['tag']).size()
```

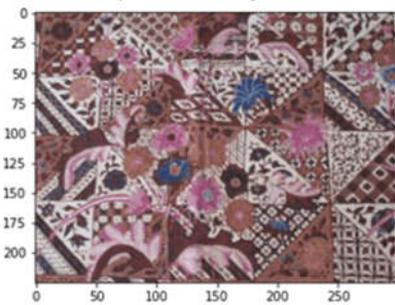
Hasil dari baris kode diatas dijalankan, maka akan menghasilkan informasi berikut:

```

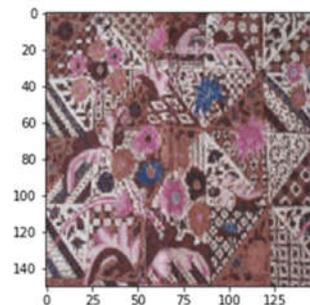
tag
batik bali          69
batik betawi       62
batik celup        51
batik cendrawasih  67
batik ceplok       51
batik ciamis       53
batik garutan      48
batik gentongan    48
batik kawung       66
batik keraton      50
batik lasem        70
batik megamendung  66
batik parang       70
batik pekalongan   59
batik priangan     47
batik sekar        67
batik sidoluhur   50
batik sidomukti    46
batik sogan        50
batik tambal       71
dtype: int64

```

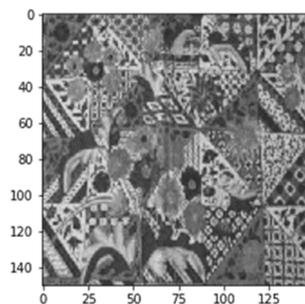
Setelah mengetahui jumlah data citra pada masing-masing citra batik yang telah diunggah selanjutnya dilakukan tahap *cropping* dan *resize* citra dengan ukuran 150 x150 pixel. Setelah tahapan tersebut selesai selanjutnya melakukan konversi citra batik ke grayscale. Adapun tahapan tersebut yakni sebagai berikut:



Gambar 3. Citra Asli



Gambar 4. Citra *Cropping & Resize*



Gambar 5. Citra *Grayscale*

Pada model CNN yang akan dibuat menggunakan API Keras yang berjalan diatas *Tensorflow*. Model yang dibuat berbentuk *Sequential*, penelitian ini menggunakan ukuran *kernel* yaitu 3x3. Ada beberapa susunan layer yang ada di dalam model *Sequential* ini. Pada lapis pertama ada layer konvolusi, pada layer konvolusi ini semua nilai *pixel* gambar berbentuk vektor dimana fungsinya untuk mengekstraksi atribut citra. Pada penelitian ini menggunakan 4 konvolusi layer dengan *neuron* 32, 64, 128, dan 512. Pada lapis kedua ada max pooling layer, max pooling selalu berada setelah konvolusi layer, max pooling layer berfungsi untuk mereduksi resolusi gambar agar proses ketika pelatihan lebih cepat. Setelah lapisan konvolusi, ada lapisan *hidden layer*. Pada layer ini, ada 3 lapisan *hidden layer* yang digunakan dengan fungsi aktivasi yang umum digunakan untuk klasifikasi gambar yaitu fungsi aktivasi relu dengan variasi *perceptron* 128, 512, dan 20. Pada lapis hidden layer yang terakhir dengan

perceptron 8 menggunakan fungsi aktivasi *softmax* untuk klasifikasi banyak kelas atau *multiclass*. Pada baris kode berikut menunjukkan arsitektur pemodelan CNN.

```
[45] model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150,3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(512, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(20, activation='softmax')
])
```

Proses *training* terhadap model dilakukan dengan memanggil fungsi *fit()* dan mengisi parameter pada fungsi *fit* tersebut. Parameter tersebut terdiri dari iterasi atau epoch dari pelatihan yang nantinya akan ditampilkan untuk melihat pergerakan proses pelatihan.

```
# latih model dengan model.fit
history = model.fit(
    train_generator, # base data latih
    steps_per_epoch=train_generator.samples // BATCH_SIZE, # berapa batch yang akan dieksekusi pada setiap epoch
    # steps_per_epoch=6,
    epochs=200, # tambahkan epochs jika akurasi model belum optimal
    validation_data=validation_generator, # menampilkan akurasi pengujian data validasi
    validation_steps=validation_generator.samples // BATCH_SIZE, # berapa batch yang akan dieksekusi pada setiap epoch
    # shuffle=True,
    verbose=2) #menampilkan progress bar dimana kita dapat melihat proses pelatihan.
```

Seperti yang terlihat pada baris kode di atas pada baris pertama melatih model dengan memanggil fungsi *fit()* dan ditampung pada *variable history*. Pada baris kedua *train_generator* adalah semua data gambar dari data latih. Pada baris ketiga, parameter *step_per_epoch* adalah jumlah dari data latih dibagi dengan *batch_size* atau jumlah file per *epoch*. Pada baris ke empat, *parameter epochs* merupakan banyaknya iterasi pelatihan yang akan dilakukan. Pada baris ke lima, parameter *validation_data*, diisi dengan data validasi yang akan memvalidasi data latih saat proses pelatihan. Pada baris ke enam *validation_steps* adalah jumlah dari data validasi dibagi dengan *batch_size* atau jumlah file per *epoch*. Pada baris ke tujuh, *verbose* merupakan parameter untuk melihat proses kemajuan per epoch saat pelatihan. Pada baris terakhir, *callbacks* merupakan fungsi untuk memantau kinerja model, pada *callbacks* disini yang di pantau adalah akurasi, jika akurasi mencapai hasil yang sudah ditentukan maka proses pelatihan akan dihentikan secara otomatis sebelum mencapai batas *epoch*.

```
Epoch 1/200
14/14 - 82s - loss: 3.5732 - accuracy: 0.0565 - val_loss: 2.9848 - val_accuracy: 0.0625
Epoch 2/200
14/14 - 78s - loss: 2.9981 - accuracy: 0.0750 - val_loss: 2.9640 - val_accuracy: 0.1172
Epoch 3/200
14/14 - 77s - loss: 3.0221 - accuracy: 0.0669 - val_loss: 2.9623 - val_accuracy: 0.0938
Epoch 4/200
14/14 - 74s - loss: 2.9612 - accuracy: 0.0830 - val_loss: 2.9346 - val_accuracy: 0.0781
Epoch 5/200
14/14 - 73s - loss: 2.9384 - accuracy: 0.0911 - val_loss: 2.8392 - val_accuracy: 0.1094
```

Proses Awal Pelatihan Per-epoch

```

14/14 - 74s - loss: 0.3173 - accuracy: 0.8870 - val_loss: 1.8821 - val_accuracy: 0.6094
Epoch 191/200
14/14 - 74s - loss: 0.3423 - accuracy: 0.8881 - val_loss: 1.1593 - val_accuracy: 0.6797
Epoch 192/200
14/14 - 74s - loss: 0.3470 - accuracy: 0.8939 - val_loss: 1.1843 - val_accuracy: 0.6875
Epoch 193/200
14/14 - 73s - loss: 0.5357 - accuracy: 0.8339 - val_loss: 0.8869 - val_accuracy: 0.7734
Epoch 194/200
14/14 - 74s - loss: 0.3496 - accuracy: 0.8697 - val_loss: 0.8938 - val_accuracy: 0.7578
Epoch 195/200
14/14 - 73s - loss: 0.3719 - accuracy: 0.8812 - val_loss: 1.1758 - val_accuracy: 0.7578
Epoch 196/200
14/14 - 74s - loss: 0.3548 - accuracy: 0.8772 - val_loss: 1.5708 - val_accuracy: 0.6484
Epoch 197/200
14/14 - 74s - loss: 0.6299 - accuracy: 0.8337 - val_loss: 0.9534 - val_accuracy: 0.7578
Epoch 198/200
14/14 - 73s - loss: 0.2981 - accuracy: 0.8973 - val_loss: 0.9505 - val_accuracy: 0.7578
Epoch 199/200
14/14 - 72s - loss: 0.4870 - accuracy: 0.8431 - val_loss: 0.9292 - val_accuracy: 0.8438
Epoch 200/200
14/14 - 72s - loss: 0.2968 - accuracy: 0.8973 - val_loss: 0.7229 - val_accuracy: 0.7812

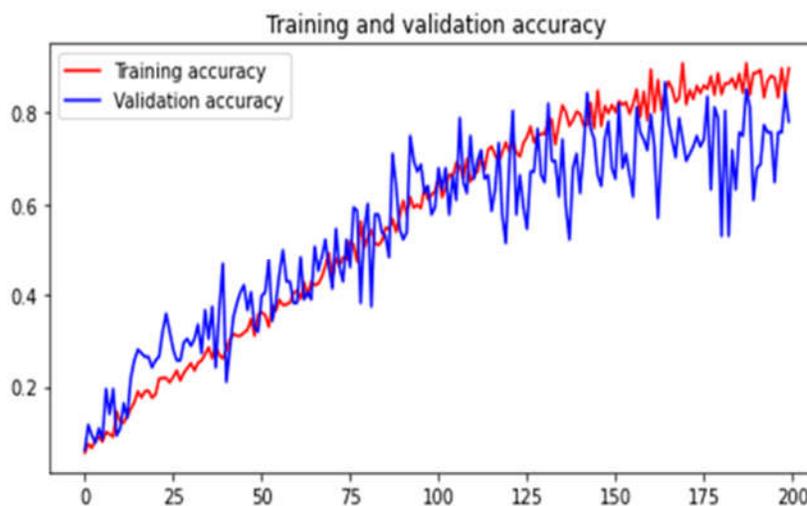
```

Proses Akhir Pelatihan Per-epoch

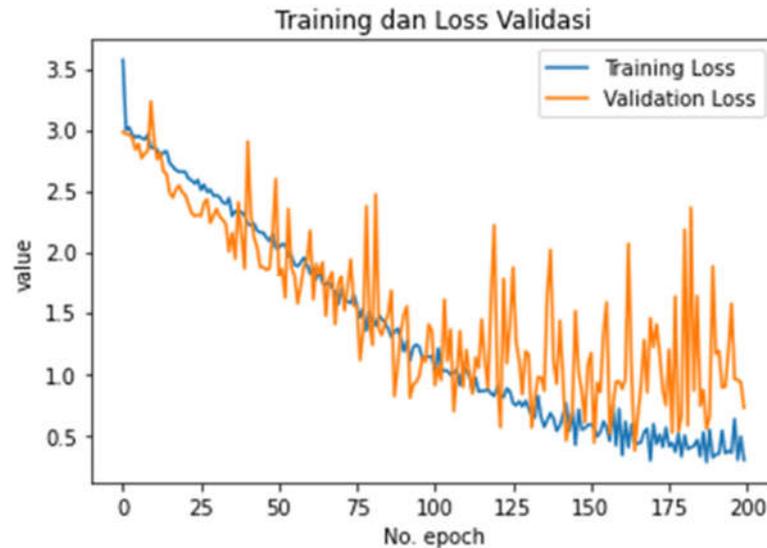
Terlihat pada baris kode diatas bahwa iterasi/epoch yang digunakan yaitu sebanyak 200. Penjelasan nilai loss, accuracy, val_loss dan val_accuracy adalah sebagai berikut:

1. *Loss* menyatakan nilai dari *loss function*, terlihat bahwa pada baris kode diatas *epoch* kesatu *loss* mendapatkan nilai 3.5732 yang cukup besar kemudian nilai itu menurun untuk setiap *epoch*.
2. *Accuracy* merupakan nilai akurasi dari data latih, terlihat pada baris kode diatas akurasi yang didapat 0.0565 atau bisa dibilang 5%, hal ini diharapkan terus meningkat seiring bertambahnya *epoch*.
3. *Val_Loss*, menyatakan nilai dari *loss function* pada data validasi, terlihat bahwa pada baris kode diatas *epoch* kesatu *loss* mendapatkan nilai 2.9848 kemudian nilai itu fluktuatif untuk setiap *epoch*.
4. *Val_accuracy* sesuai namanya, *val* merupakan nilai akurasi dari data validasi, terlihat pada baris kode diatas akurasi yang didapat 0.0625 atau bisa dibilang 6%, hal ini diharapkan terus meningkat seiring bertambahnya *epoch*.

Berikut adalah grafik hasil dari proses dari *training accuracy* di tunjukkan pada gambar 6, dan *training loss* di tunjukkan pada gambar 7.



Gambar 6. Training Accuracy.



Gambar 7. Training Loss

Tahap uji coba ini dilakukan dengan cara memasukkan satu persatu citra batik sebanyak 200 ke program yang telah dibuat untuk diproses sehingga diketahui jenis batik yang telah dimasukkan. Berikut adalah hasil dari pengujian yang telah dilakukan terhadap 200 dataset citra batik, ditunjukkan pada tabel 1 dibawah.

Tabel 1. Hasil Pengujian Citra Batik menggunakan CNN

Label / Jenis Batik	Jumlah Data	Prediksi Benar	Prediksi Salah	Akurasi
Batik Bali	10	2	8	20%
Batik Betawi	10	3	7	30%
Batik Celup	10	6	4	60%
Batik Cendrawasih	10	1	9	10%
Batik Ceplok	10	2	8	20%
Batik Ciamis	10	4	6	40%
Batik Garutan	10	1	9	10%
Batik Gentongan	10	2	8	20%
Batik Kawung	10	0	10	0%
Batik Keraton	10	5	5	50%
Batik Lasem	10	2	8	20%
Batik Megamendung	10	8	2	80%
Batik Parang	10	3	7	30%
Batik Pekalongan	10	5	5	50%
Batik Priangan	10	0	10	0%
Batik Sekar	10	2	8	20%
Batik Sidoluhur	10	1	9	10%
Batik Sidomukti	10	0	10	0%
Batik Sogan	10	1	9	10%
Batik Tambal	10	3	7	30%
Total	200	51	149	25,5%

Persentase klasifikasi jenis batik ini dilakukan dengan cara mencocokkan motif data citra batik *testing* dengan data citra batik *training*. Untuk memperoleh nilai akurasi klasifikasi jenis batik dilakukan dengan menggunakan persamaan berikut:

Jumlah prediksi benar = 51 citra batik

Jumlah total prediksi = 200 citra batik

$$Akurasi = \frac{Jumlah\ prediksi\ benar}{Jumlah\ total\ prediksi}$$

$$Akurasi = \frac{51}{200} \times 100\% = 25,5\%$$

Adapun hasil klasifikasi yang dilakukan dengan 200 data uji seperti batik bali, batik betawi, batik celup, batik cendrawasih, batik ceplok, batik ciamis, batik garutan, batik gentongan, batik kawung, batik keraton, batik lasem, batik megamendung, batik parang, batik pekalongan, batik priangan, batik sekar, batik sidoluhur, batik sidomukti, batik sogan, batik tambal.

Pada tahap uji coba citra batik sebagai dataset yang dilakukan mendapat hasil dengan tingkat akurasi yang paling besar adalah data uji citra batik megamendung sebesar 80% dan batik ceplok 60% karena batik megamendung dan batik celup memiliki motif yang sama hanya berbeda warna serta untuk hasil uji citra batik secara keseluruhan memperoleh persentase sebesar 25,5%.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian yang telah dilakukan dalam penelitian ini, maka dapat disimpulkan bahwa “Implementasi Algoritma Convolutonal Neural Network (CNN) Untuk Klasifikasi Batik“ sudah berhasil diterapkan. Hasil dari pengujian terhadap 200 dataset dan 20 label diperoleh nilai akurasi yang tertinggi adalah “Batik Megamendung dan Batik Celup” dengan nilai akurasi 80% dan 60%, hasil accuracy yang diperoleh dari proses pelatihan model dari 200 epoch yang tertinggi adalah 90%, serta tingkat akurasi secara keseluruhan memperoleh persentase sebesar 25,5%. Adapun saran dari penelitian yang telah dilakukan yakni:

1. Menambah lebih banyak dataset per-kelasnya, karena deeplearning semakin banyak dataset maka semakin banyak fitur yang bisa ditangkap.
2. Meningkatkan ukuran dimensi gambar, agar model dapat meningkatkan hasil nilai akurasi
3. Diharapkan pada peneitian selanjutnya membuat model yang lebih optimal sehingga dapat mengurangi nilai *loss* yang cukup tinggi.

6. DAFTAR PUSTAKA

- [1] W. Rofianto, M. G. Haque, D. Sunarsi, A. Purwanto, and H. S. Kahpi, “Cultural Product Branding, Antecedents, and its Implications: A Study on the Context of Indonesian Batik,” vol. 2, no. 2, p. 10.
- [2] R. Andrian, M. A. Naufal, B. Hermanto, A. Junaidi, and F. R. Lumbanraja, “k-Nearest Neighbor (k-NN) Classification for Recognition of the Batik Lampung Motifs,” *J. Phys.: Conf. Ser.*, vol. 1338, no. 1, p. 012061, Oct. 2019, doi: 10.1088/1742-6596/1338/1/012061.
- [3] D. A. Septhesia, “Penerapan Brand Batique Terhadap Tren Fashion Generasi Milenial Sebagai Alternatif Melestarikan Batik Indonesia,” *Prosiding Konferensi Mahasiswa Desain Komunikasi Visual 2020*, p. 8.
- [4] J. Han, *Data Mining: Concepts and Techniques*, Second Edition. San Francisco: Morgan Kaufmann, 2006.
- [5] I. W. Suartika E. P, A. Y. Wijaya, and R. Soelaiman, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” *JTITS*, vol. 5, no. 1, Mar. 2016, doi: 10.12962/j23373539.v5i1.15696.
- [6] F. Rizal, A. Wijaya, and U. R. Hidayat, “Penerapan Algoritma Backpropagation Untuk Klasifikasi Jenis Buah Rambutan Berdasarkan Fitur Tekstur Daun,” *jatim*, vol. 1, no. 2, pp. 1–8, Oct. 2020, doi: 10.31102/jatim.v1i2.969.