



Department of Digital Business

**Journal of Artificial Intelligence and Digital Business (RIGGS)**

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 4 No. 4 (2025) pp: 1515-1532

P-ISSN: 2963-9298, e-ISSN: 2963-914X

---

## Klasifikasi Motif Batik Jambi Menggunakan *Convolutional Neural Network*

Fadli Pratama, M.Theo Ari Bangsa, Andreo Yudertha

Sistem Informasi, Sains dan Teknologi, Universitas Islam Negeri Sulthan Thaha Saifuddin Jambi

[fadliprtaaaa@gmail.com](mailto:fadliprtaaaa@gmail.com), [theoaribangsa@uinjambi.ac.id](mailto:theoaribangsa@uinjambi.ac.id), [andreo@uinjambi.ac.id](mailto:andreo@uinjambi.ac.id)

### Abstrak

Batik merupakan budaya yang sangat melekat di Indonesia dan dapat ditemukan di seluruh Nusantara dengan gaya unik masing-masing daerah, termasuk batik Jambi. Batik Jambi memiliki kekhasan pada motif, warna, serta filosofi yang terkandung di dalamnya. Namun, pemahaman masyarakat, khususnya generasi muda, terhadap ragam motif batik Jambi masih terbatas. Oleh karena itu, penelitian ini bertujuan mengembangkan sistem klasifikasi otomatis untuk mengidentifikasi motif batik Jambi menggunakan kecerdasan buatan, khususnya *Convolutional Neural Network (CNN)*. Arsitektur yang digunakan adalah VGG16, salah satu model CNN yang terbukti efektif untuk pengenalan citra. Penelitian ini melibatkan tahap pengumpulan dataset yang terdiri dari 1215 citra batik Jambi yang terbagi ke dalam 15 motif berbeda seperti Angso Duo, Batanghari, dan Bungo Melati. Tahapan penelitian mencakup augmentasi data untuk memperkaya variasi citra, pelatihan model, dan evaluasi performa sistem. Hasil penelitian menunjukkan bahwa model final VGG16 mampu mengidentifikasi motif batik Jambi dengan sangat baik, ditunjukkan oleh akurasi sebesar 97,43%, validation akurasi 99,11%, serta rata-rata precision, recall, dan F1-score sebesar 99%. Nilai evaluasi yang berada di atas 95% membuktikan bahwa model mampu melakukan klasifikasi secara konsisten dan akurat. Dengan adanya sistem ini, diharapkan masyarakat Jambi dan generasi muda dapat lebih mengenal dan memahami motif batik daerahnya, sekaligus membantu upaya pelestarian budaya. Teknologi kecerdasan buatan dapat menjadi sarana penting dalam mendukung pelestarian dan digitalisasi warisan budaya Indonesia.

*Kata kunci:* Klasifikasi, Motif Batik Jambi, CNN

### 1. Latar belakang

Batik adalah budaya yang sangat melekat di Indonesia, sehingga dapat menemukan batik di seluruh Nusantara dengan gaya unik mereka. Pada tanggal 2 Oktober 2009, UNESCO memasukkan Batik Indonesia ke dalam Daftar Representatif sebagai Budaya Warisan Manusia, mengakui batik sebagai warisan budaya dunia. Namun, Presiden Indonesia Joko Widodo menyatakan bahwa UNESCO sedang mengevaluasi pengakuan warisan budaya dunia tersebut pada hari batik nasional 2 Oktober 2019. Presiden mengimbau masyarakat Indonesia untuk mempertahankan batik.[1]

Batik Jambi adalah salah satu budaya nasional yang harus dilestarikan agar tidak hilang. Ini karena motif-motif batik Jambi mengandung nilai-nilai religius, nasehat, kebiasaan, dan bahkan sejarah daerah Jambi. Oleh karena itu, informasi tentang bentuk motif dan maknanya sangat penting untuk membantu masyarakat Jambi mendapatkan atau memahami informasi tersebut.[2]

Beberapa motif batik Jambi terinspirasi dari alam jambi, seperti *flora* dan *fauna*. Namun, corak dan pewarnaannya berbeda dari batik lain di Indonesia. Bentuk dan pewarnaan batik Jambi unik karena motifnya yang tidak berangkai (ceplok-ceplok) dan berdiri sendiri. Pemberian nama pada motif batik Jambi, diberikan pada setiap satu bentuk motif, seperti motif Bungo Melati, motif Bungo Tanjung, dan sebagainya. Jadi bukan diberikan pada suatu rangkaian bentuk dari berbagai unsur atau elemen yang telah di desain sedemikian rupa yang telah menjadi satu kesatuan yang utuh baru di beri nama.[2]

Akan tetapi, walaupun orang Jambi cukup gemar dan sering mengenakan batik, tak sedikit juga yang masih kurang kenal dengan motif yang berada pada produk batik yang digunakan. Salah satu tindakan konkret untuk melestarikan batik adalah dengan mengenal motif-motif pada batik Jambi. Banyaknya pola batik di Jambi mengakibatkan sulitnya untuk mengidentifikasi motif, khususnya bagi masyarakat awam. Adanya suatu metode untuk mempermudah pengenalan pola batik tentu memiliki banyak manfaat. Salah satu metode yang bisa

digunakan adalah dengan melakukan klasifikasi dengan tahapan mengumpulkan data gambar dan memperbanyak dengan *augmentasi* data, setelah mendapatkan *dataset*, data akan diolah *mechine learning* agar dapat mengklasifikasikan gambar ke dalam kategori tertentu menggunakan metode *Convulotional Neural Network* yang akan menampilkan akurasi dari data tersebut.

Dalam penelitian ini menggunakan arsitektur model VGG16 yang di usulkan oleh K.Simonyan dan A. Zisserman di *Universitas Oxford*. Model ini dapat memperbaiki konvolusi yang sangat kecil dalam jaringan Terdapat penelitian klasifikasi batik yang menggunakan model VGG16 untuk ekstraksi ciri. Oleh karena itu model VGG16 dapat mengklasifikasikan gambar batik dengan sangat baik.[3] . Biasanya, proses klasifikasi data melibatkan dua fase: yang pertama adalah fase pembelajaran, di mana algoritma klasifikasi mengembangkan pengklasifikasi dengan memeriksa kumpulan data yang ada, yang umumnya disebut sebagai set pelatihan, yang mencakup label untuk setiap bagian data. Fase kedua adalah fase pengujian, di mana pengklasifikasi yang dibuat selama fase pembelajaran digunakan untuk mengklasifikasikan atau memperkirakan kumpulan data baru dengan nilai yang tidak diketahui untuk variabel dependen atau label.[4] Model ini mengambil gambar masukan dua dimensi dan mengubah fitur-fiturnya menjadi *matriks* kolom satu dimensi. MLP memiliki n-lapisan (kotak merah dan biru), dengan masing-masing lapisan.[5]. pembelajaran mesin (ML). Untuk membagi antara data latihan dan data uji, metode validasi holdout dan cross-fold dapat digunakan. Validasi sangat penting untuk dilakukan karena tujuannya adalah agar setiap data memiliki kesempatan untuk dilatih dan diuji.[6]. Dari hasil pemodelan dengan optimasi terbaik, kemudian dilakukan pengujian untuk mengklasifikasikan data test yang juga merupakan citra batik jambi, penentuan dari suatu performa baik atau tidaknya dalam suatu klasifikasi dapat di ukur dalam parameternya, yaitu tingkat akurasi, *recall*, dan presisi.[7]

Berdasarkan permasalahan di atas, terdapat kebutuhan perangkat beserta metode yang dapat digunakan untuk mengklasifikasikan motif batik jambi. Identifikasi terhadap batik jambi secara detail memerlukan pengolahan dan analisis mendalam dengan menggunakan metode *convulotional neural network*.

## 2. Metodologi Penelitian

### 2.1 Metode Analisis Data

Pada penelitian ini digunakan metode untuk mengklasifikasi gambar motif batik jambi menggunakan *Convolutional Neutral Network* (CNN) yang merupakan salah satu metode yang terkenal mampu mengklasifikasi data gambar dengan baik. Proses untuk mengolah algoritma CNN menggunakan bahasa pemrograman *python* dalam *machine learning* layanan *google colab*. [8] Adapun cara kerja yang dilakukan adalah dengan mengenali objek atau citra sebagai *input* dan *output* yang diharapkan adalah tingkat akurasi pengenalan objek tersebut.

### 2.2 Tahapan Penelitian

Proses penelitian melibatkan serangkaian Langkah sistematis yang perlu diikuti. Tahapan-tahapan dalam penelitian ini adalah sebagai berikut:



Gambar 1 Tahapan Penelitian

## 2.3 Tahapan Pembuatan Program

### 2.3.1 Pengumpulan Data

Data yang dibutuhkan dalam penelitian klasifikasi ini Menggunakan data *primer* dan *sekunder*. Data *primer* yang akan di dapatkan dengan foto langsung sumber data di sanggar baik jambi, sedangkan data *sekunder* yang akan diperoleh dari berbagai sumber internet maupun sosial media. *Dataset* akan memiliki banyak gambar sehingga menghasilkan kualitas data yang baik agar dapat digunakan dalam proses klasifikasi. [9]

### 2.3.2 Praproses Data

Dalam sebuah penelitian yang berfokus pada klasifikasi suata data, praproses data merupakan langkah penting yang dilakukan. Praproses data memastikan bahwa data yang akan diproses memiliki kualitas yang sah dan baik. Untuk klasifikasi CNN, praproses data dilakukan dengan memberi label pada data dan mengubah dimensi gambar agar sama. Data yang telah diproses dibagi menjadi dua bagian: data pelatihan dan data pengujian. Data pelatihan digunakan untuk proses pelatihan data pada klasifikasi, sedangkan data pengujian digunakan untuk proses pengujian data pada klasifikasi.[10]

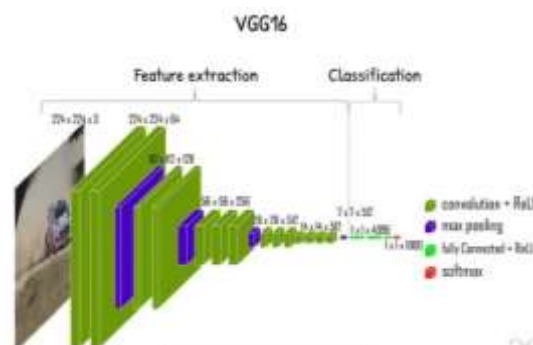
Menurut Perez & Wang (2017) *augmentasi* data adalah suatu proses dalam pengolahan data gambar, *augmentasi* merupakan proses mengubah atau memodifikasi gambar sedemikian rupa sehingga komputer akan mendeteksi bahwa gambar yang diubah adalah gambar yang berbeda, namun manusia masih dapat mengetahui bahwa gambar yang diubah tersebut adalah gambar yang sama. *Augmentasi* berguna untuk mendapatkan data gambar tambahan dengan cara *rotation*, *flipping* dan *cropping* yang dapat membuat model CNN akurasi yang tinggi.[11]

### 2.3.3 Perancangan Model CNN

Perancangan model CNN adalah bagian dari proses membuat model yang akan digunakan dalam proses klasifikasi. Proses ini termasuk menentukan ukuran jumlah lapisan, menentukan struktur CNN, menentukan filter yang akan digunakan, menentukan ukuran kernel, menentukan arsitektur CNN, dan menentukan ukuran pooling selama proses klasifikasi.[12]

Setelah data dibuat, langkah selanjutnya adalah melatih model CNN. Secara umum CNN terdiri dari dua fase: fase pembelajaran fitur dan fase klasifikasi. Gambar yang dimasukkan pada model CNN menggunakan gambar berukuran 224 x 224 x 3 Yang ketiga adalah gambar dengan tiga saluran: merah, hijau, dan biru (RGB) Kemudian, gambar masukan diproses terlebih dahulu melalui proses konvolusi dan *pooling* pada tahap pembelajaran fitur. Banyaknya proses konvolusi pada desain ini terdiri dari dua lapisan konvolusi. Setiap konvolusi memiliki jumlah filter yang berbeda dan ukuran kernel yang berbeda. Selanjutnya dilakukan dropout untuk memulai proses perataan, yaitu mengubah peta fitur yang diperoleh dari *pooling layer* menjadi *format vektor*. Proses ini biasa disebut sebagai tahap lapisan yang terhubung sepenuhnya.[13]

Berikut adalah rancangan dari arsitektur CNN pada penelitian ini yang dapat dilihat pada Gambar.2.



Gambar 2 Arsitektur CNN

Berdasarkan arsitektur tersebut dijelaskan bahwa arsitektur CNN memiliki dua fase yaitu *Feature Extraction* dan *Classification*. *Feature Extraction* merupakan teknik yang memungkinkan suatu sistem secara otomatis

menentukan representasi suatu gambar menjadi fitur dalam bentuk nilai numerik yang mewakili gambar tersebut. Fase *Classification* merupakan fase dimana hasil pembelajaran fitur digunakan untuk pemrosesan klasifikasi berdasarkan *subclass* yang telah ditentukan. Berdasarkan gambar ini, proses konvolusi pertama menggunakan 3 *filter* dan *kernel matriks* 3x3. Selanjutnya dilakukan *padding "same"*, kemudian melakukan proses *MaxPooling* dengan ukuran pooling 2x2 dan *Steride* 2. Kemudian melakukan konvolusi sebanyak enam kali. Kelas citra mengklasifikasikan berdasarkan nilai dari *neuron* pada lapisan terakhir dengan menggunakan fungsi aktivasi *softmax*.

### 2.3.4 Rancangan Pengujian Model

Pengujian dilakukan untuk mengevaluasi model CNN. Tahap pengujian model yang telah dilakukan pada tahap pelatihan. Dalam penelitian ini, 15 kelas motif batik jambi yang akan di uji. Tujuan dari uji ini adalah untuk mengetahui apakah model mampu mengklasifikasikan gambar dengan baik. Untuk melakukan evaluasi model ini, fitur model *Tensorflow* digunakan. evaluasi dan juga menggunakan metode *confusion matrix* untuk memberikan pemeriksaan model yang lebih rinci. Rancangan *confusion matriks* penelitian ini ditunjukkan pada tabel 1.

Tabel 1 Akurasi Matriks

Matriks		Prediksi			
		Motif Batanghari (B)	Motif Angso Duo (A)	Motif Daun Keladi (D)	
Aktual	Motif Batanghari (B)	BB	BA	BD	False Negatif (FN)
	Motif Angso Duo (A)	AB	AA	AD	
	Motif Daun Keladi (D)	DB	DA	DD	True Positif (TP)
					False Positif (FP)

*Confusion Matriks* dapat di gunakan untuk mengetahui nilai akurasi, *precision*, dan *recall* dari model klasifikasi motif batik jambi.

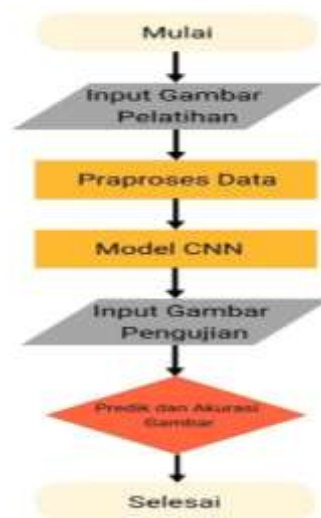
$$\text{Akurasi} = \frac{TP \text{ Total}}{\text{Total Dataset}}$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{TP}{\text{Total Prediksi}}$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{TP}{\text{Total Aktual}}$$

Jumlah prediksi yang sesuai dengan aktual dari kelasnya sendiri, seperti motif batik batanghari yang diantisipasi sebagai motif batik batanghari, TN adalah jumlah data bukan kelas yang diantisipasi bukan kelas, seperti motif batik batanghari yang tidak diantisipasi sebagai motif batik batanghari, FN adalah jumlah prediksi yang diantisipasi bukan kelas yang seharusnya diantisipasi kelasnya sendiri, seperti motif batik batanghari yang diantisipasi sebagai motif batik batanghari, FP merupakan jumlah prediksi yang dianggap kelasnya sendiri yang seharusnya bukan termasuk kelasnya sendiri seperti motif batik selain batanghari diprediksi sebagai motif batik Batanghari.[14]

### Flowchart Program



Gambar 3 Flowchart Program

Berdasarkan gambar 3 flowchart program terdiri dari proses Input citra, praproses data, model CNN, input gambar Kemudian melakukan hasil uji coba untuk mengetahui hasil akurasi dan predik gambar dari training and validation accuracy dan training and validation loss.

### 3. Hasil dan pembahasan

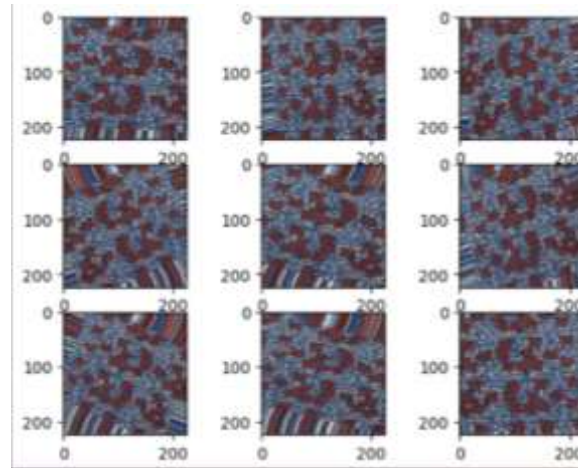
#### 3.1 Augmentasi Data

Data *augmentation* digunakan untuk memperbanyak data. Pada penelitian ini, Proses *augmentation* dilakukan menggunakan Google Colab dengan *dataset* dari google drive.

```
1 from google.colab import drive
2 drive.mount('/content/drive')
3
4 import numpy as np
5 import random as rd
6 import matplotlib.pyplot as plt
7
8 from keras.preprocessing.image import ImageDataGenerator
9 save_dir = '/content/drive/MyDrive/aiartifacts_data/labels/'
10
11 train_datagen = ImageDataGenerator(
12     rotation_range = 180,
13     zoom_range = 0.2,
14     width_shift_range = 0.2,
15     height_shift_range = 0.2,
16     shear_range = 0.2,
17     mode = 'nearest',
18     fill_mode = 'nearest')
19
20 state_generator = train_datagen.flow_from_directory(
21     '/content/drive/MyDrive/aiartifacts_data/labels',
22     target_size = (224, 224),
23     batch_size=32,
24     class_mode='categorical',
25     save_dir=save_dir)
26 img = train_generator.flow()[0][0]
27
28 for i in range(5):
29     plt.subplot(2, 2, i + 1)
30     augmented_image = train_datagen.random_transform(img)
31     plt.imshow(augmented_image)
```

Gambar 4 Code Augmentasi Data

dari code di atas akan menghasilkan gambar *augmentasi* seperti pada gambar 4.5.



Gambar 5. Hasil Augmentasi Data

### 3.2 Preprocessing Data

Pada tahap ini peneliti membuat sebuah folder di google drive yang bernama klasifikasi\_batik, di dalam folder tersebut ada 3 folder yaitu bahan, train, dan val. Pada folder bahan memiliki 100% dataset motif batik jambi dan selebihnya data kosong, dari data tersebut akan di split validation 80% data train dan 20% data val dengan code di bawah ini.

```

1 !pip install split-folders
2 import os
3 import shutil
4 base_dir = '/content/drive/My Drive/klasifikasi_batik/'
5 bahan_dir = os.path.join(base_dir, 'bahan')
6 outputbase_dir = '/content/drive/My Drive/klasifikasi_batik/'
7 splitfolders.split(bahan_dir, output=outputbase_dir, seed=1237, ratio=(.8, .2), group_prefix=None)
8 train_dir = os.path.join(outputbase_dir, 'train')
9 validation_dir = os.path.join(outputbase_dir, 'val')
    
```

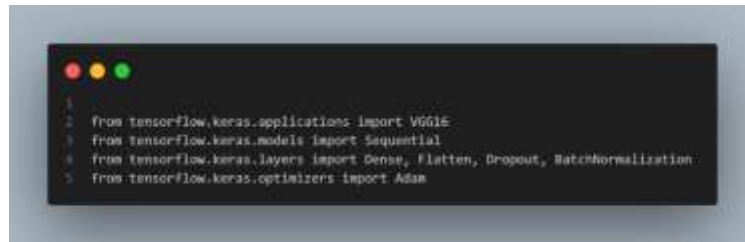
Gambar 6 Code Split Validation

dari code tersebut mendapatkan 966 gambar 15 kelas di data train dan 249 gambar 15 kelas di data val.

Motif batik	Bahan	Train	Val
Angso Duo	164	131	33
Batanghari	134	107	27
Bungo Kaco Piring	40	32	8
Bungo Melati	68	54	14
Bungo Tanjung	16	12	4
Bungo Teratai	8	6	2
Daun Keladi	146	116	30
Durian Pecah	125	100	25
Kapak Lepas	56	44	12
Kapal Sanggat	116	92	24
Kuau Berhias	144	115	29
Merak Ngeram	15	12	3
Relung Kangkung	9	7	2
Riang – Riang	38	30	8
Tampuk Manggis	136	108	28
<b>Total</b>	<b>1215</b>	<b>966</b>	<b>249</b>

### 3.3 Perancangan Model CNN

Setelah melalui berapa tahapan, maka selanjutnya proses perancangan model, pada penilitan ini menggunakan arsitektur VGG16.



```
1  
2 from tensorflow.keras.applications import VGG16  
3 from tensorflow.keras.models import Sequential  
4 from tensorflow.keras.layers import Dense, Flatten, Dropout, BatchNormalization  
5 from tensorflow.keras.optimizers import Adam
```

Gambar 7 Perancangan VGG16

Kode ini membangun sebuah model *deep learning* yang dapat digunakan untuk mengklasifikasikan gambar menjadi 15 kelas. Model ini memanfaatkan kekuatan VGG16 sebagai ekstraktor fitur dan menambahkan lapisan-lapisan *fully connected* untuk melakukan klasifikasi.

### 3.4 Pengujian

Pada tahap pengujian, peneliti melakukan 5 kali percobaan dengan tujuan membandingkan penggunaan arsitektur VGG16 dan hanya menggunakan sequential untuk mendapatkan akurasi yang tinggi.

#### 3.4.1 Pengujian 1 (Sequential dengan 2 kelas data)

Pada pengujian ini menggunakan 2 kelas motif batik jambi yaitu, motif angso duo dan motif batanghari, yang memiliki data train 238 dan data val 60, dengan *code preprocessing* dibawah ini.



```
1 from tensorflow.keras.preprocessing.image import ImageDataGenerator  
2  
3 train_datagen = ImageDataGenerator(  
4     rescale=1/255,  
5     rotation_range=20,  
6     width_shift_range=0.1,  
7     height_shift_range=0.1,  
8     shear_range=0.2,  
9     zoom_range=0.2,  
10    horizontal_flip=True,  
11    brightness_range=(0.2, 1.2),  
12    fill_mode='nearest')  
13  
14 test_datagen = ImageDataGenerator(  
15     rescale=1/255,  
16     rotation_range=20,  
17     width_shift_range=0.1,  
18     height_shift_range=0.1,  
19     shear_range=0.2,  
20     zoom_range=0.2,  
21     horizontal_flip=True,  
22     brightness_range=(0.2, 1.2),  
23     fill_mode='nearest')  
24  
25 train_generator = train_datagen.flow_from_directory(  
26     train_dir,  
27     target_size=(224, 224),  
28     batch_size=32,  
29     class_mode='categorical')  
30  
31 test_generator = test_datagen.flow_from_directory(  
32     validation_dir,  
33     target_size=(224, 224),  
34     batch_size=32,  
35     class_mode='categorical')
```

Gambar 8 Code Pengujian 1

Pada gambar 4.8 ini untuk mempersiapkan data gambar motif batik jambi yang ingin latih. Setelah melakukan *preprocessing* data selanjutnya akan melakukan rancangan model *convuolotional neural network* dengan *code* dibawah ini.

```
1 model = tf.keras.models.Sequential([
2     tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(224, 224, 3)), # Changed 'layers' to 'layers'
3     tf.keras.layers.MaxPooling2D(2, 2),
4     tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
5     tf.keras.layers.MaxPooling2D(2,2),
6     tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
7     tf.keras.layers.MaxPooling2D(2,2),
8     tf.keras.layers.Flatten(),
9     tf.keras.layers.Dense(200, activation='relu'),
10    tf.keras.layers.Dropout(0.5, seed=112),
11    tf.keras.layers.Dense(500, activation='relu'),
12    tf.keras.layers.Dropout(0.5, seed=112),
13    tf.keras.layers.Dense(2, activation='sigmoid'),])
14
```

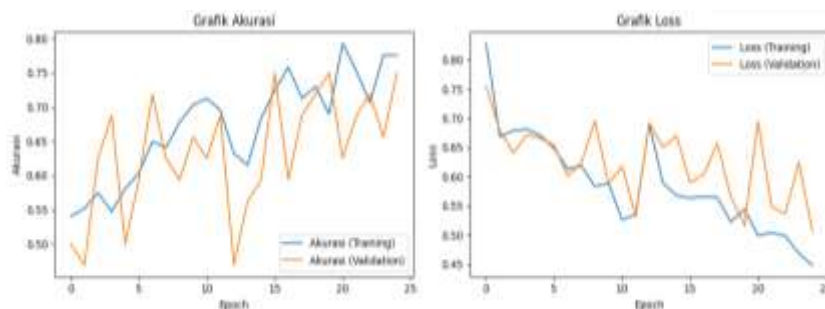
Gambar 9 Code model layer

pada code gambar 4.9 menggunakan model Sequential, Model ini dirancang untuk melakukan klasifikasi gambar biner menggunakan arsitektur *convolutional neural network* (CNN). Penggunaan layer konvolusi, *pooling*, dan *fully connected* memungkinkan model untuk belajar representasi fitur tingkat tinggi dari gambar dan melakukan klasifikasi dengan akurasi yang baik. pada model ini melakukan convulotional layer sebanyak 3 kali, *maxpooling* sebanyak 3 kali, danse layer sebanyak 3 kali, dropout layer sebanyak 2 kali, dan output layer dengan Mengambil output dari Dropout Layer 2 dan menghubungkannya dengan 2 neuron. Setiap neuron mewakili satu kelas.

```
1 import tensorflow as tf
2
3 class myCallback(tf.keras.callbacks.Callback):
4     def on_epoch_end(self, epoch, logs={}):
5         if logs.get('accuracy') >= 0.95:
6             print("\nAkurasi telah mencapai >95!")
7
8 callbacks = myCallback()
9
10
11 history = model.fit(train_generator,
12                    steps_per_epoch=6,
13                    epochs=25,
14                    validation_data=test_generator,
15                    validation_steps = 1,
16                    verbose = 1,
17                    callbacks=[callbacks] )
18
```

Gambar 10 Code Melatih Model

pada pengujian ini menggunakan *epochs = 25*, *steps per epoch = 6*, *validation steps = 1*, dan *verbose = 1*.mendapatkan akurasi 77.59% dan *validation akurasi 75.00%*. dengan grafik .



Gambar 11 Grafik Pengujian 1

### 3.4.2 Pengujian 2 (VGG16 dengan 2 kelas data)

Pada pengujian kedua ini sama seperti pengujian pertama memiliki 2 kelas motif batik jambi yaitu, motif angsu duo dan motif Batanghari, yang memiliki data train 238 dan data val 60. Dengan menggunakan rancangan model *Convolutional Neural Network* arsitektur VGG16.

```
1 # Parameter
2 IMAGE_SIZE = (128, 128)
3 BATCH_SIZE = 32
4 NUM_CLASSES = 2 # Ganti dengan jumlah kelas Anda
5 EPOCHS = 50 # Meningkatkan jumlah epoch
6
7 # Memuat dan mempersiapkan data
8 train_datagen = ImageDataGenerator(
9     rescale=1./255,
10    rotation_range=40, # Meningkatkan rentang rotasi
11    width_shift_range=0.2, # Meningkatkan rentang pergeseran lebar
12    height_shift_range=0.2, # Meningkatkan rentang pergeseran tinggi
13    shear_range=0.2, # Meningkatkan rentang shear
14    zoom_range=0.2, # Meningkatkan rentang zoom
15    horizontal_flip=True,
16    fill_mode='nearest'
17)
18
19 test_datagen = ImageDataGenerator(rescale=1./255)
20
```

Gambar 12 Persiapan Data Pengujian 2

Pada gambar 4.12 ini mempersiapkan untuk meningkatkan kinerja model pada data yang belum pernah dilihat sebelumnya dengan mempersiapkan data gambar untuk pelatihan model *deep learning* dengan melakukan *augmentasi* data yang lebih agresif pada data pelatihan dibandingkan dengan data pengujian.

```
1 # Memuat gambar yang akan
2 train_dir = 'data/pelatihan_dog' # Ganti dengan path folder data
3 validation_dir = 'data/pengujian_dog' # Ganti dengan path folder data
4
5 # Membuat generator untuk data pelatihan
6 train_generator = ImageDataGenerator(
7     rescale=1./255,
8     rotation_range=40,
9     width_shift_range=0.2,
10    height_shift_range=0.2,
11    shear_range=0.2,
12    zoom_range=0.2,
13    horizontal_flip=True,
14    fill_mode='nearest'
15)
16
17 # Membuat generator untuk data pengujian
18 validation_generator = ImageDataGenerator(
19     rescale=1./255,
20    rotation_range=0,
21    width_shift_range=0,
22    height_shift_range=0,
23    shear_range=0,
24    zoom_range=0,
25    horizontal_flip=False,
26    fill_mode='nearest'
27)
28
29 # Memuat model VGG16 yang sudah dilatih
30 model = load_model('vgg16_weights.h5')
31
32 # Mengubah lapisan yang akan dilatih
33 for layer in model.layers[:10]: # Ganti dengan jumlah layer yang
34     layer.trainable = False
35
36 # Menetapkan model
37 model.compile(loss='categorical_crossentropy',
38             optimizer='adam',
39             metrics=['accuracy'])
40
41 # Melatih model
42 model.fit(train_generator.flow(train_dir, class_indices),
43         validation_data=(validation_generator.flow(validation_dir), class_indices),
44         epochs=EPOCHS,
45         callbacks=[callbacks])
46
```

Gambar 13 Membangun Model Pengujian 2

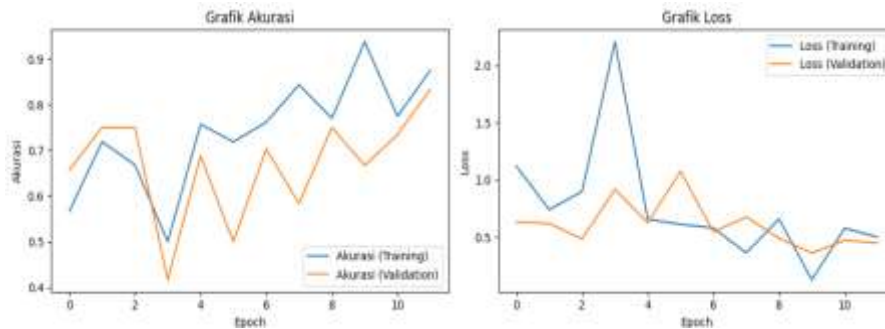
Pada gambar 4.13 line 1-17 path data pada pengujian ini, line 20 membuat model VGG16, line 23-24 mengatur lapisan yang yang dapat dilatih, dan line 27-37 membangun model VGG16. Tujuan dari gambar di atas untuk membangun model klasifikasi gambar yang akurat dengan memanfaatkan kekuatan model pra-latih VGG16 dan teknik *augmentasi* data. Model ini dapat digunakan untuk mengklasifikasikan gambar ke dalam beberapa kelas berdasarkan fitur-fitur visual yang telah dipelajarinya.

```
1 # Mengimpor the keras
2 from keras.preprocessing.image import ImageDataGenerator # Memuat library yang akan digunakan
3 from keras.models import Sequential # Memuat library yang akan digunakan
4 from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Softmax, Dropout # Memuat library yang akan digunakan
5
6 # Membuat model VGG16
7 model = Sequential()
8 model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(128, 128, 3)))
9 model.add(MaxPooling2D(pool_size=(2, 2), activation='relu'))
10 model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
11 model.add(MaxPooling2D(pool_size=(2, 2), activation='relu'))
12 model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
13 model.add(MaxPooling2D(pool_size=(2, 2), activation='relu'))
14 model.add(Conv2D(256, kernel_size=(3, 3), activation='relu'))
15 model.add(MaxPooling2D(pool_size=(2, 2), activation='relu'))
16 model.add(Conv2D(512, kernel_size=(3, 3), activation='relu'))
17 model.add(MaxPooling2D(pool_size=(2, 2), activation='relu'))
18 model.add(Flatten())
19 model.add(Dense(4096, activation='relu'))
20 model.add(Dense(4096, activation='relu'))
21 model.add(Dense(1000, activation='softmax'))
22 model.add(Dropout(0.5))
23 model.add(Dense(1000, activation='softmax'))
24
25 # Kompilasi model
26 model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
27
28 # Melatih model
29 train_datagen = ImageDataGenerator(
30     rescale=1./255,
31     rotation_range=40,
32     width_shift_range=0.2,
33     height_shift_range=0.2,
34     shear_range=0.2,
35     zoom_range=0.2,
36     horizontal_flip=True,
37     fill_mode='nearest'
38)
39
40 validation_datagen = ImageDataGenerator(rescale=1./255)
41
42 # Melatih model
43 model.fit(train_datagen.flow(train_dir, class_indices),
44         validation_data=(validation_datagen.flow(validation_dir), class_indices),
45         epochs=EPOCHS,
46         callbacks=[callbacks])
47
```

Gambar 14 Code Latih Model

Pada gambar 4.13 line 1-4 untuk memberikan instruksi kepada model tentang bagaimana cara belajar dari data. Model akan berusaha meminimalkan nilai *loss function* dengan menyesuaikan bobotnya, dan akurasi akan digunakan untuk memantau kemajuan pelatihan. Line 6-19 bagian kode ini berfokus pada *callbacks* yang digunakan dalam proses pelatihan model *deep learning*. *Callbacks* adalah mekanisme yang memungkinkan kita untuk melakukan tindakan tertentu selama proses pelatihan berlangsung. Tindakan ini bisa berupa menyimpan model terbaik, jika sudah menyentuh 90% training akan berhenti otomatis. Line 21-29 kode ini merupakan langkah akhir dalam proses pelatihan model *deep learning*. Melalui proses pelatihan ini, model diharapkan dapat belajar dari data dan mencapai kinerja yang baik dalam melakukan tugas klasifikasi gambar.

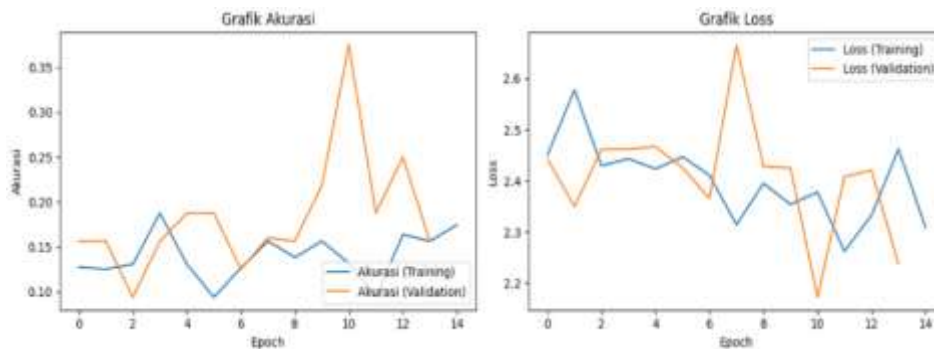
Pada penelitian ini menggunakan *epoch* = 10, *steps per epoch* = 8, mendapatkan akurasi 87.50% dan *validation* akurasi 84.00%. dengan grafik.



Gambar 15 Grafik Pengujian 2

### 3.4.3 Pengujian 3 (Sequential dengan 15 kelas data)

Pada pengujian ini rancangan model *Convolutional Neural Network* sama seperti pengujian 1, hanya membedakan jumlah data, *epoch*, dan *step per epoch*. Jumlah data train 966 dalam 15 kelas, data val 249 dalam 15 kelas, *epoch* 15, dan *steps per epoch* 30 .pada pengujian ini mendapatkan akurasi 17.45% dan *validation* akurasi 15.62%, dengan grafik.

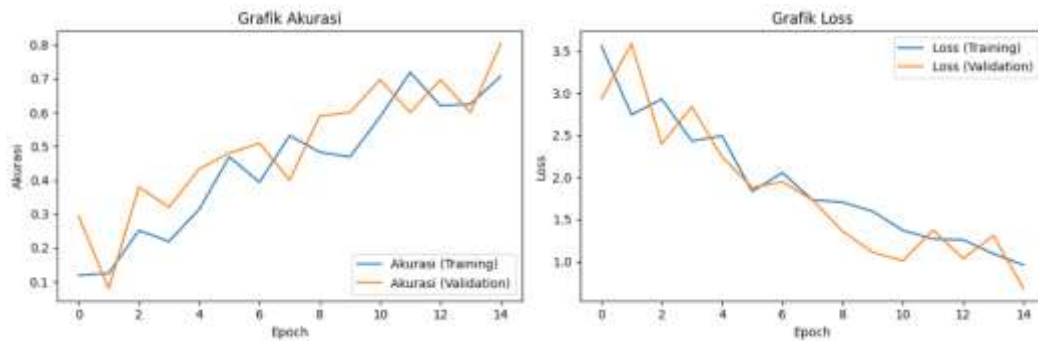


Gambar 16 Grafik Pengujian 3

### 3.4.4 Pengujian 4 (VGG16 dengan 15 kelas data)

Pada pengujian ini rancangan model *Convolutional Neural Network* sama seperti pengujian 2 hanya membedakan jumlah data, *epoch*, dan *step per epoch*. Jumlah data *train* 966 dalam 15 kelas dan data val 249 dalam 15 kelas.

Dengan jumlah data *train* 966 dalam 15 kelas, data val 249 dalam 15 kelas, *epochs* 15 dan *steps per epochs* 30 mendapatkan akurasi 70.66% dan *validation* akurasi 80.36%. dengan grafik.



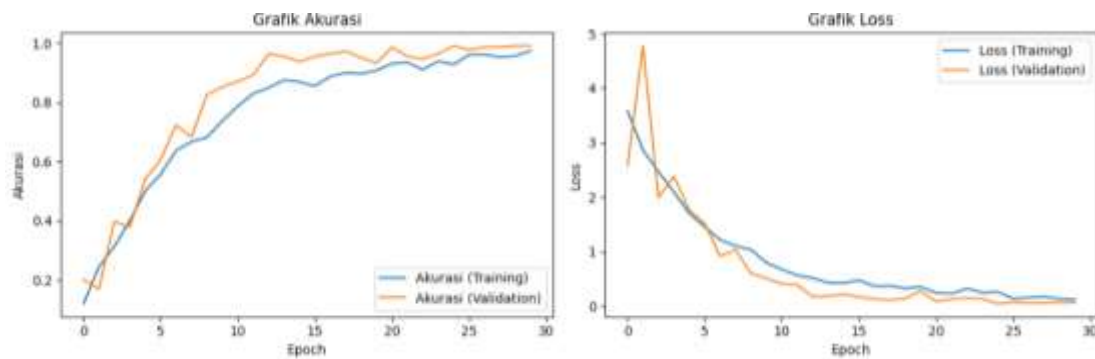
Gambar 17 Grafik Pengujian 4

### 3.4.5. Pengujian Final (VGG16 dengan 15 kelas data & 30 Epoch)

Pada pengujian ini rancangan model *Convolutional Neural Network* sama seperti pengujian 2 dan jumlah data sama seperti pengujian 4, hanya membedakan *epoch*, dan *step per epoch*. Jumlah data *train* 966 dalam 15 kelas dan data *val* 249 dalam 15 kelas. Dalam penelitian ini menggunakan arsitektur VGG16 *epoch* 30, *steps per epoch* 30 mendapatkan akurasi 97.43% dan *validation* akurasi 99.11%.

### 3.5 Hasil Pengujian Model Final

Sesuai dengan skenario sebelumnya, percobaan dilakukan pada bagian ini dengan membagi seluruh data dengan 80% data *train* dan 20% data *validation* sesuai dengan model yang telah dibuat. Hasil berikut adalah grafik akurasi pelatihan dan validasinya.



Gambar 18 Grafik Pengujian Final

#### Training Accuracy

Dengan 30 epochnya dan 30 step per epochnya, akurasi pelatihan meningkat setiap epochnya. Proses pelatihan tertinggi mencapai 97.43% pada epochnya ketiga puluh, dan akurasi terendah mencapai 11.99% pada epochnya pertama.

#### Validation Accuracy

Akurasi dari *validation* dengan 30 *epochs* dan 30 *steps per epochs* mengalami peningkatan setiap *epochnya*. Dari proses *validation* yang dilakukan mendapatkan akurasi pelatihan tertinggi diperoleh di *epoch* ke-30, yaitu sebesar 99.11% dan akurasi terendah terjadi pada *epoch* pertama, yaitu sebesar 20.09%.

Akurasi Validasi sudah termasuk dalam pengujian. Pengujian ini menggunakan data uji sebanyak 1215 citra dengan jumlah setiap kelas berbeda-beda. Seperti yang telah dijelaskan sebelumnya, jumlah data uji pada tiap kelas ini didapatkan dari hasil pembagian data training dan *validation* dengan skenario 80:20. Sebaran data uji pada tiap kelas ditunjukkan pada tabel berikut:

**Tabel 3. Jumlah Data Validation Setiap Kelas**

<b>Motif batik</b>	<b>Validation</b>
Angso Duo	33
Batanghari	27
Bungo Kaco Piring	8
Bungo Melati	14
Bungo Tanjung	4
Bungo Teratai	2
Daun Keladi	30
Durian Pecah	25
Kapak Lepas	12
Kapal Sangat	24
Kuau Berhias	29
Merak Ngeram	3
Relung Kangkung	2
Riang – Riang	8
Tampuk Manggis	28
<b>Total</b>	<b>249</b>

Dari pengujian data tersebut, juga didapatkan confusion matrix sebagai berikut:

**Tabel 4 Tabel Confusion Matrix**

<b>Class</b>	<b>TP</b>	<b>FP</b>	<b>TN</b>	<b>FN</b>
Angso Duo	33	1	215	0
Batanghari	24	0	222	3
Bungo Kaco Piring	8	0	241	0
Bungo Melati	14	1	234	1
Bungo Tanjung	4	0	245	0
Bungo Teratai	2	0	247	0
Daun Keladi	30	0	219	0
Durian Pecah	25	0	224	0
Kapak Lepas	12	0	237	0
Kapal Sangat	24	0	225	0
Kuau Berhias	29	1	219	1
Merak Ngeram	3	0	246	0
Relung Kangkung	2	0	247	0
Riang-Riang	8	0	241	0
Tampuk Manggis	28	0	221	0

Dari hasil confusion matrix tersebut, didapatkan hasil perhitungan *accuracy*, *precision*, *recall*, dan *f1 score* sebagai berikut:

	precision	recall	f1-score	support
angso_duo	0.97	1.00	0.99	33
batanghari	1.00	0.89	0.94	27
bungo_kaco_piring	1.00	1.00	1.00	8
bungo_melati	0.93	1.00	0.97	14
bungo_tanjung	1.00	1.00	1.00	4
bungo_taratai	1.00	1.00	1.00	2
daun_keladi	1.00	1.00	1.00	30
durian_pecah	1.00	1.00	1.00	25
kapak_lepas	1.00	1.00	1.00	12
kapal_sangat	1.00	1.00	1.00	24
kuau_berhias	0.97	1.00	0.98	29
merak_ngeras	1.00	1.00	1.00	3
relung_kangkung	1.00	1.00	1.00	2
riang-riang	1.00	1.00	1.00	8
tampuk_manggis	1.00	1.00	1.00	28
accuracy			0.99	249
macro avg	0.99	0.99	0.99	249
weighted avg	0.99	0.99	0.99	249

Gambar 19 Classification Precision, Recall dan F1-Score

Dengan menggunakan sebanyak 249 data uji, dapat disimpulkan bahwa hasil prediksi model sangat baik pada proses uji, seperti yang ditunjukkan dalam tabel 4.3 dan Gambar 4.9. Akurasi dari model dengan ukuran gambar 224 x 224 piksel, dengan 30 epoch, mencapai 99%. Berikut adalah contoh perhitungan untuk mendapatkan nilai precision, recall, dan f1 pada kelas "Angso\_Duo":

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{33}{33+1} = \frac{33}{34} = 0.97$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{33}{33+0} = \frac{33}{33} = 1.00$$

$$\text{F1-Score} = 2x \left( \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) = 2x \left( \frac{0.97 \times 1.00}{0.97 + 1.00} \right) = 2x \left( \frac{0.97}{1.97} \right) = 0.98$$

### 3.6 Perbandingan Setiap Pengujian

Setiap pengujian dalam penelitian ini adanya perbedaan dalam segi arsitektur maupun jumlah *epoch* dan sebagainya, berikut perbedaanya:

#### Pengujian 1 & Pengujian 2

Pada pengujian 1 & 2 sama menggunakan 2 kelas, kelas angso duo dan Batanghari, pengujian 1 menggunakan model sequential dengan 3 lapisan konvolusi, 3 lapisan pooling, 3 lapisan fully connected, dan 2 lapisan dropout, menggunakan *epochs* 25 dan steps per *epochs* 6 mendapatkan akurasi 77.59% dan *validation* akurasi 75.00%.



Gambar 20 Hasil Epochs Akurasi Tertinggi Pengujian 1

Pada pengujian 2 menggunakan model VGG16 dengan menggunakan *epochs* 10 dan steps per *epochs* 8 mendapatkan akurasi 87.50% dan *validation* 84.00%.

Jadi, pada pengujian 1 & 2 dengan jumlah data yang sama dan hanya membedakan jenis modelnya, pengujian 1 menggunakan sequantial sedangkan pengujian 2 menggunakan model VGG16 dan juga jumlah *epochs* yang berbeda dari pengujian ini dapat kita lihat akurasi yang lebih tinggi adalah pengujian 2 karena menggunakan arsitektur model VGG16.

### 3.7 Pengujian 3 & Pengujian 4

Pada pengujian 3 & 4 sama-sama menggunakan 15 kelas motif batik jambi, pengujian 3 menggunakan model sequential dengan 3 lapisan konvolusi, 3 lapisan *pooling*, 3 lapisan *fully connected*, dan 2 lapisan dropout, menggunakan *epochs* 15 dan steps per *epochs* 30 mendapatkan akurasi 17.45% dan *validation* akurasi 15.62%.



**Gambar 21 Hasil Epochs Akurasi Tertinggi Pengujian 3**

Pada pengujian 2 menggunakan model VGG16 dengan menggunakan *epochs* 15 dan steps per epochs 30 mendapatkan akurasi 84.37% dan validation 90.62%.

Jadi, pada pengujian 3 & 4 dengan jumlah data, epochs, steps per epoch yang sama dan hanya membedakan jenis modelnya, pengujian 3 menggunakan sequential sedangkan pengujian 4 menggunakan model VGG16, dari pengujian ini dapat kita lihat akurasi yang lebih tinggi adalah pengujian 4 karena menggunakan arsitektur model VGG16. Berikut tabel perbandingan:

**Tabel 5. Perbandingan**

Sequential	VGG16	Kesimpulan
Pada pengujian 1 menggunakan 2 kelas data, sedangkan pengujian 3 menggunakan 15 kelas data, pengujian 1 & 3 menggunakan 3 lapisan konvolusi, 3 lapisan pooling, 3 lapisan fully connected, dan 2 lapisan dropout, menggunakan <i>epochs</i> 25 dan steps per epochs 6 mendapatkan akurasi pada 2 kelas data 77.59% dan validation akurasi 75.00%, akurasi 15 kelas data 17.45% dan validation akurasi 15.62%	pada pengujian 2 menggunakan 2 kelas data, sedangkan pengujian 4 menggunakan 15 kelas data, pengujian 2 & 4 menggunakan arsitektur VGG16 mendapatkan akurasi pada pengujian 2, 87.50% dan validation akurasi 84.00%, akurasi 15 kelas data 84.37% dan validation akurasi 90.62%.	Arsitektur VGG16 lebih baik daripada model Sequential untuk tugas klasifikasi gambar, terutama pada dataset dengan jumlah kelas yang banyak. Hal ini menunjukkan bahwa arsitektur VGG16 memiliki kemampuan generalisasi yang lebih baik dan lebih cocok untuk menangani kompleksitas data yang lebih tinggi.  Jumlah kelas data sangat mempengaruhi kinerja model. Semakin banyak kelas data, semakin sulit bagi model untuk melakukan klasifikasi dengan akurasi yang tinggi. Model Sequential yang sederhana tidak mampu mengatasi kompleksitas dataset 15 kelas, sedangkan VGG16 yang memiliki arsitektur yang lebih dalam dan kompleks dapat menangani dataset tersebut dengan lebih baik.

Pada tabel diatas adalah perbandingan pengujian menggunakan sequential dengan arsitektur VGG16. Berikut adalah tabel akurasi setiap pengujian dalam penelitian ini:

**Tabel 6. Akurasi Setiap Pengujian**

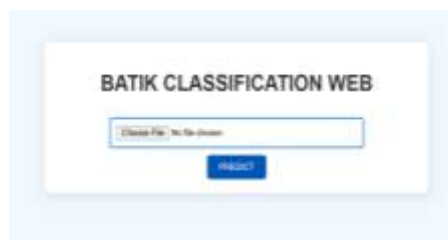
Pengujian	Jumlah Data	Jenis Model	Epoch & Step per epoch	Akurasi	Validation Akurasi	Runtime	Waktu Training
1	Train (236) Val (60)	Sequential	Epoch (25) Per Epoch (6)	77.59%	75.00%	T4 GPU	1 Menit 43 Detik
2	Train (236) Val (60)	VGG16	Epoch (10) Per Epoch (8)	87.50%	84.00%	T4 GPU	4 Menit 31 Detik
3	Train (966) Val (249)	Sequential	Epoch (15) Per Epoch (30)	17.45%	15.62%	v2-8 TPU	21 Detik

4	Train (966) Val (249)	VGG16	Epoch (15) Per Epoch (30)	70.66%	80.36%.	T4 GPU	15 Detik
Final	Train (966) Val (249)	VGG16	Epoch (30) Per Epoch (30)	97.43%	99.11%.	v2-8 TPU	2 Jamb 35 Menit 22 Detik

Pada tabel 4.3 kita bisa melihat pada penelitian ini mendapatkan akurasi dan validation yang tinggi itu menggunakan arsitektur model VGG16.

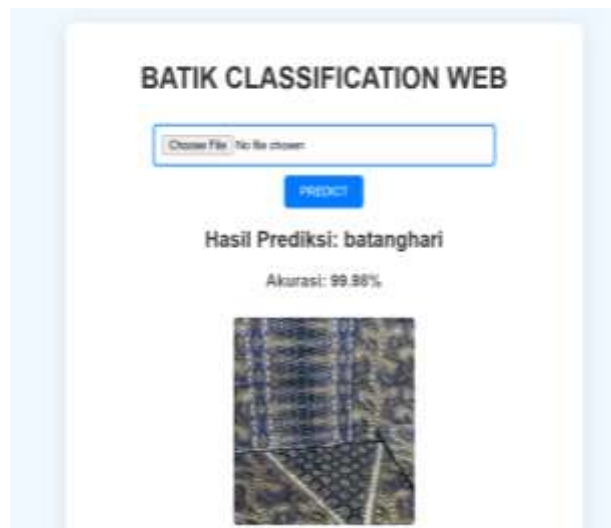
### 3.8 Hasil Klasifikasi Gambar dan Testing Menggunakan Website

Pada titik ini, pengujian pendeteksian gambar dilakukan. Pengujian akhir untuk mengklasifikasi motif batik jambi dilakukan dengan menggunakan aplikasi berbasis web yang terintegrasi dengan Model VGG16. Aplikasi ini dijalankan pada URL berikut: <http://127.0.0.1:5000/>, dan halaman utamanya adalah Flask, microframework Python yang digunakan untuk mengembangkan situs web ini.



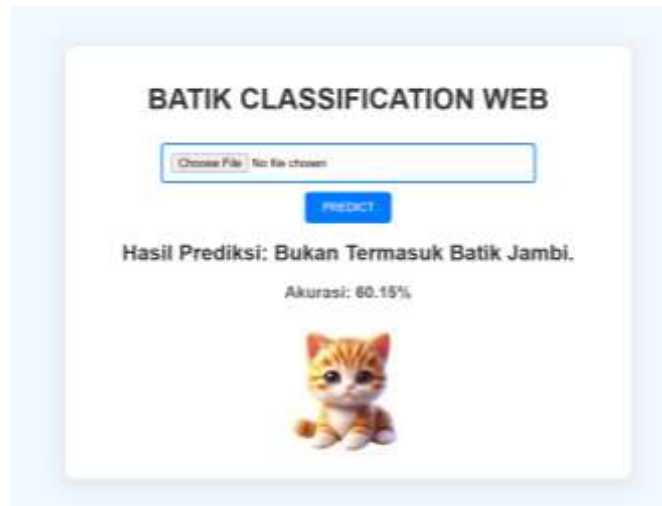
**Gambar 22 Halaman Website**

Pada halaman utama ini, user dapat mengupload gambar dari penyimpanan local. Setelah itu, sistem akan mengirimkan file gambar tersebut ke model untuk diproses dalam klasifikasi motif batik jambi. Setelah gambar berhasil diproses oleh Model VGG16, maka akan dihasilkan output. Output yang dihasilkan berupa kelas dengan akurasi tertinggi yang akan ditampilkan pada halaman yang baru seperti berikut:



**Gambar 23 Hasil Prediksi Motif Batik Jambi**

Dapat mengisi form dengan memasukkan link gambar yang ditemukan di internet. Setelah file gambar dimasukkan, pengguna dapat menekan tombol "prediksi". Sistem akan mengirimkan file gambar ke model untuk diproses untuk mendeteksi jenis sampah. Setelah gambar diproses dengan benar oleh model, output akan dihasilkan.



**Gambar 24 Hasil Prediksi diluar motif batik jambi**

Ini dapat terjadi karena model jaringan saraf kontemporer memiliki kemampuan prediktif yang luar biasa, tetapi mereka tidak dapat mengetahui kapan prediksi mereka melakukan kesalahan [15]. Menurut [15], metode deteksi keluar dari distribusi dapat digunakan untuk menyelesaikan masalah ini. Mengidentifikasi objek yang tidak termasuk dalam kelas yang telah ditetapkan sebelumnya akan menjadi masalah yang dapat diatasi model dengan menggunakan metode ini.

Selanjutnya dilakukan pengujian menggunakan data uji baru (diluar dari dataset) dengan skenario data sebanyak 45 citra gambar dengan tiap kelas masing masing berjumlah 3 gambar motif batik jambi sesuai dengan kelasnya.

**Tabel 7. Confusion Matrix 3 Citra**

Class	TP	FP	TN	FN
Angso Duo	3	2	40	0
Batanghari	2	0	42	1
Bungo Kaco Piring	3	0	42	0
Bungo Melati	3	0	42	0
Bungo Tanjung	3	0	42	0
Bungo Teratai	3	0	42	0
Daun Keladi	3	0	42	0
Durian Pecah	3	0	42	0
Kapak Lepas	3	0	42	0
Kapal Sanggat	3	0	42	0
Kuau Berhias	2	0	42	1
Merak Ngeram	3	0	42	0
Relung Kangkung	3	0	42	0
Riang-Riang	3	0	42	0
Tampuk Manggis	3	0	42	0

Dari hasil confusion matrix tersebut, didapatkan hasil perhitungan accuracy, precision, recall, dan f1 score sebagai berikut:

**Tabel 8. Classification Report Precision, Recall dan F1-score 3 Citra**

Motif Baitk Jambi	precision	recall	f1-score	support
angso_duo	0.60	1.00	0.75	3
batanghari	1.00	0.67	0.80	3
bungo_kaco_piring	1.00	1.00	1.00	3
bungo_melati	1.00	1.00	1.00	3
bungo_tanjung	1.00	1.00	1.00	3
bungo_taratai	1.00	1.00	1.00	3
daun_keladi	1.00	1.00	1.00	3
durian_pecah	1.00	1.00	1.00	3
kapak_lepas	1.00	1.00	1.00	3
kapal_sangat	1.00	1.00	1.00	3
kuau_berhias	1.00	0.67	0.80	3
merak_ngeram	1.00	1.00	1.00	3
relung_kangkung	1.00	1.00	1.00	3
riang-riang	1.00	1.00	1.00	3
tampuk_manggis	1.00	1.00	1.00	3
	0.97	0.95	0.96	45

Berdasarkan tabel 17, dapat disimpulkan bahwa, dengan menggunakan sebanyak 45 data uji akurasi uji, yaitu sebesar 95%, hasil prediksi model cukup baik selama proses uji. Perhitungan berikut untuk mendapatkan nilai akurasi, recall, dan skor f1 pada kelas "Batanghari":

$$\text{Akurasi} = \frac{TP+TN}{(TP+TN+FP+FN)} = \frac{44+42}{(44+42+2+2)} = \frac{86}{90} = 0.95$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{2}{2+0} = 1.00$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{2}{2+1} = \frac{2}{3} = 0.66$$

Dari perhitungan di atas, dapat disimpulkan bahwa rasio prediksi benar positif adalah 1 dibandingkan dengan keseluruhan hasil prediksi yang benar positif; rasio prediksi benar positif juga adalah 0.66 dibandingkan dengan keseluruhan data yang benar positif. Selain itu, kami dapat mengetahui dari tiga data Val kelas "Batanghari".

#### 4. Kesimpulan

Berdasarkan penelitian ini, pengujian final menggunakan data 1215 citra, data train 966 citra dan data val 249 citra dalam 15 kelas motif batik jambi menggunakan arsitektur VGG16 dengan epoch 30 dan steps per epoch 30 mendapatkan akurasi 97.43% dan validation akurasi 99.11%. Akurasi matrix pada motif angso duo Precision 0.97, Recall 1.00 dan F1-Score 0.98. Perbandingan dalam pengujian yang hanya menggunakan sequential dan menggunakan arsitektur VGG16 dapat di lihat pada pengujian 3 & 4, dengan jumlah data, epochs, steps per epoch yang sama dan hanya membedakan jenis modelnya, pengujian 3 menggunakan sequential mendapatkan akurasi 17.45% dan validation akurasi 15.62% sedangkan pengujian 4 menggunakan model VGG16 mendapatkan akurasi 70.66% dan validation akurasi 80.36%, pengujian ini dapat kita lihat akurasi yang lebih tinggi adalah pengujian 4 karena menggunakan arsitektur model VGG16. Percobaan pengujian di luar dataset, pada pengujian ini menggunakan 3 citra dalam setiap kelasnya, akurasi matrix dalam motif Batanghari Precision 0.95 Recall 1.00 F1-Score 0.66.

## Referensi

- [1] S. F. Tumewu, D. H. Setiabud, and I. Sugiarto, "Klasifikasi Motif Batik menggunakan Metode Deep Convolutional Neural Network dengan Data Augmentation," *J. Infra*, vol. 8, no. 2, pp. 189–194, 2020.
- [2] J. Surya, R. Wahyuning Astuti, and P. Priyanto, "Media Informasi Pengenalan Makna Dan Motif Batik Jambi Berbasis Android Pada Sanggar Batik Olak Kemang," *J. Akad.*, vol. 14, no. 2, pp. 81–87, 2022, doi: 10.53564/akademika.v14i2.865.
- [3] A. E. Minarno, "Klasifikasi Jenis Batik Menggunakan Algoritma Convolutional Neural Network," *J. Repos.*, vol. 3, no. 2, pp. 199–206, 2021, doi: 10.22219/repositor.v3i2.1201.
- [4] S. Wahyuni Kalumbang, "Perbandingan Regresi Logistik, Klasifikasi Naive Bayes, Dan Random Forest (Comparison the Logistic Regression, Naive Bayes Classification, and Random Forest)," *J. Mat. Thales*, vol. 03, no. 02, pp. 1–13, 2021.
- [5] S. Ariessaputra, V. H. Vidiyari, S. M. Al Sasongko, B. Darmawan, and S. Nababan, "Classification of Lombok Songket and Sasambo Batik Motifs Using the Convolution Neural Network (CNN) Algorithm," *Int. J. Informatics Vis.*, vol. 8, no. 1, pp. 38–44, 2024, doi: 10.62527/joiv.8.1.1386.
- [6] R. Oktafiani, A. Hermawan, and D. Avianto, "Pengaruh Komposisi Split data Terhadap Performa Klasifikasi Penyakit Kanker Payudara Menggunakan Algoritma Machine Learning," *J. Sains dan Inform.*, no. August, pp. 19–28, 2023, doi: 10.34128/jsi.v9i1.622.
- [7] L. A. Andika, H. Pratiwi, and S. S. Handajani, "Klasifikasi Penyakit Pneumonia Menggunakan Metode Convolutional Neural Network Dengan Optimasi Adaptive Momentum," *Indones. J. Stat. Its Appl.*, vol. 3, no. 3, pp. 331–340, 2019, doi: 10.29244/ijsa.v3i3.560.
- [8] D. Gede, T. Meranggi, N. Yudistira, and Y. Arum, "Batik Classification Using Convolutional Neural Network with Data Improvements," vol. 6, no. March, pp. 6–11, 2022.
- [9] G. D. A. N. Cnn, A. R. Dani, and I. Handayani, "KLASIFIKASI MOTIF BATIK YOGYAKARTA MENGGUNAKAN METODE GLCM DAN CNN," vol. 10, no. 2, pp. 142–156, 2024.
- [10] T. Bariyah, M. A. Rasyidi, U. Internasional, and S. Indonesia, "Convolutional Neural Network Untuk Metode Klasifikasi Multi-Label Pada Motif Batik," vol. 20, no. 1, pp. 155–165, 2021.
- [11] R. Mawan, "Klasifikasi motif batik menggunakan Convolutional Neural Network," *Jnanaloka*, pp. 45–50, 2020, doi: 10.36802/jnanaloka.2020.v1-no1-45-50.
- [12] M. N. Azmi, T. Informatika, P. Lor, and J. Tengah, "ANALISIS PENGARUH DIMENSI GAMBAR PADA KLASIFIKASI MOTIF BATIK DENGAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK," vol. 6, no. 2, pp. 190–198, 2023.
- [13] S. Arifin, J. Letjend, P. Soemarto, and K. P. Utara, "Klasifikasi Motif Batik Menggunakan Metode Convolutional Neural Network ( CNN ) Dengan Multi Class Clasification," vol. 10, no. 1, pp. 30–38, 2024.
- [14] A. Prayoga, Maimunah, P. Sukmasetya, Muhammad Resa Arif Yudianto, and Rofi Abul Hasani, "Arsitektur Convolutional Neural Network untuk Model Klasifikasi Citra Batik Yogyakarta," *J. Appl. Comput. Sci. Technol.*, vol. 4, no. 2, pp. 82–89, 2023, doi: 10.52158/jacost.v4i2.486.
- [15] T. DeVries and G. W. Taylor, "Learning Confidence for Out-of-Distribution Detection in Neural Networks," 2018, [Online]. Available: <http://arxiv.org/abs/1802.04865>