

# **SISTEM PENGENALAN TEMPAT WISATA KABUPATEN TUBAN MENGUNAKAN *DEEP LEARNING* DENGAN METODE *CONVOLUTIONAL NEURAL NETWORK (CNN)***

**Yella Ariska Safitri**

Universitas 17 Agustus 1945 Surabaya, Jl. Semolowaru No.45, Menur Pumpungan,  
Kec. Sukolilo, Kota SBY, Jawa Timur 60118, +62-31-5931800, [humas@untag-sby.ac.id](mailto:humas@untag-sby.ac.id).

## **Abstract**

*Tourist objects are places where people vacation with family or alone. This place can also be used as a good business area if accompanied by proper management. The purpose of this research is to create a tourist attraction recognition system that can perform image searches. The method used is Deep Learning with Convolutional Neural Network (CNN) algorithm. The application of the CNN method is carried out using the Python programming language using a dataset which is grouped into two parts, training data and test data. In the training process, the data used are 720 images consisting of 12 classes. As for the test, the data used are 10 images for each class, so that in total there are 120 images. The data is taken from one point of view so that the image that can be identified is also only from that point of view. After the test is carried out, accuracy calculations and conclusions were drawn. From the test results, the precision value is 66.7%, recall is 79,5%, F1 score is 72,5% and accuracy is 66.7%.*

**Keywords** : *image recognition, Convolutional Neural Network, Deep Learning*

## **Abstrak**

Tempat wisata adalah tempat di mana orang-orang berlibur bersama keluarga atau pun sendirian. Tempat ini juga bisa dijadikan lahan bisnis yang bagus jika disertai dengan pengelolaan yang tepat. Tujuan dari penelitian ini adalah untuk membuat sistem pengenalan objek wisata yang dapat melakukan pencarian dengan gambar. Metode yang digunakan adalah Deep Learning dengan algoritma Convolutional Neural Network (CNN). Penerapan metode CNN dilakukan dengan bahasa pemrograman Python dengan menggunakan dataset yang dikelompokkan menjadi dua bagian, data training dan data test. Dalam proses training, data yang digunakan berjumlah 720 gambar yang terdiri dari 12 kelas. Sedangkan untuk tes, data yang digunakan adalah 10 gambar untuk setiap kelas, sehingga totalnya ada 120 gambar. Data tersebut diambil dari satu sudut pandang sehingga gambar yang dapat diidentifikasi juga hanya dari sudut pandang tersebut. Setelah tes dilakukan, maka dilakukan perhitungan akurasi dan penarikan kesimpulan. Dari hasil pengujian, didapatkan nilai precision 66,7%, recall 79,5%, F1 score 72,5% dan akurasi sebesar 66,7%.

**Kata kunci** : *image recognition, Convolutional Neural Network, Deep Learning*

## 1. PENDAHULUAN

Tuban merupakan salah satu kabupaten yang berada di Provinsi Jawa Timur. Kota ini memiliki berbagai tempat wisata yang bisa dijadikan tujuan untuk berlibur. Tuban juga memiliki julukan sebagai Kota Wali dan Kota Seribu Goa. Julukan tersebut berasal dari banyaknya tempat wisata berupa Goa dan Makam Wali. Selain itu, Tuban juga memiliki objek wisata lain seperti pantai, taman bermain dan lain sebagainya.

Objek wisata adalah tempat yang dikunjungi oleh orang-orang untuk melepas lelah dan mendapatkan hiburan. Tempat ini merupakan salah satu aspek penting tidak terkecuali di Kabupaten Tuban. Di zaman modern ini orang-orang cenderung berbagi gambar melalui sosial media saat sedang berwisata. Dengan adanya media pencarian gambar, pencarian detail tempat wisata dapat dilakukan dengan leluasa.

Deep learning adalah salah satu cabang machine learning (ML). Convolutional Neural Network (CNN) adalah salah satu algoritma dari Deep Learning. CNN merupakan pengembangan dari Multi Layer Perceptron yang dirancang untuk mengolah data dalam bentuk grid. Salah satu data yang diolah adalah citra dua dimensi, misalnya gambar atau suara.[1]

Dalam penggunaannya, CNN akan membantu mengenali gambar yang akan dicari dengan sebuah filter pemecah gambar. Lalu hasil pengenalan tersebut dicocokkan dengan model yang didapat dari data training Deep Learning. Setelahnya dua keluaran akan didapatkan, yaitu berupa objek berhasil atau gagal ditemukan. Jika objek berhasil ditemukan, maka akan muncul informasi mengenainya.

Pada penelitian yang dilakukan oleh Aرسال et al (2020) [2] , metode CNN

digunakan untuk teknologi Face Recognition. Penelitian ini berhasil menggunakan Face Recognition lima orang dataset wajah pegawai bank. Masing-masing orang terdiri dari 70 data wajah. Sehingga total data wajah yang digunakan 350 data wajah. Hasil dari pengujian ketiga dataset tersebut berhasil mengidentifikasi wajah yang ditangkap oleh kamera dengan persentase keakuratan 95%.

Dengan adanya penerapan CNN pada aplikasi, diharapkan pengenalan objek wisata bisa dilakukan dengan mudah dan lebih akurat. Orang-orang akan mendapat informasi berbagai kawasan hiburan hanya dengan sebuah gambar.

## 2. METODE PENELITIAN

### 2.1 Bahan Dan Perangkat Penelitian

Dalam melakukan penelitian ini, terdapat perangkat keras dan perangkat lunak yang digunakan sebagai bahan dan perangkat penelitian. Berikut adalah perangkat keras yang digunakan dalam pembuatan sistem pengenalan tempat wisata:

1. Laptop Asus X441U RAM 4GB
2. Processor Intel Core i3-6006U, 2.0GHz
3. OPPO A5 2020

Selain itu, terdapat beberapa perangkat lunak yang digunakan dalam penelitian. Perangkat lunak tersebut adalah:

1. Visual Studio Code merupakan *editor* kode yang dapat digunakan untuk berbagai macam Bahasa pemrograman. *Editor* ini digunakan untuk membuat kode python dan html untuk sistem.

2. Opera GX Browser merupakan sebuah aplikasi untuk membuka halaman web. Aplikasi ini digunakan untuk melakukan data training CNN dengan google colab. Selain itu sistem identifikasi tempat wisata juga dijalankan di *browser* ini.
3. Microsoft Excel adalah aplikasi untuk mengolah data dalam bentuk tabel. Dalam penelitian ini, aplikasi ini digunakan untuk menyimpan data tempat wisata yang kemudian disambungkan ke file python. Microsoft Excel juga digunakan dalam pengujian gambar. Hasil pengujian disimpan dan dihitung di sini.

## 2.2 Objek Penelitian

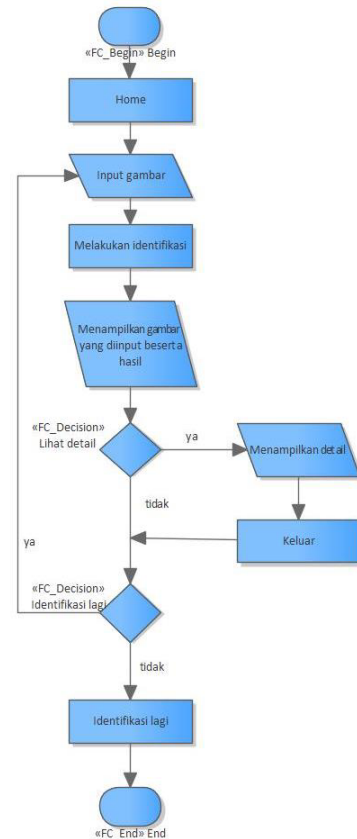
Penelitian dilakukan terhadap sekumpulan *dataset* berupa gambar tempat wisata. Data tersebut dibagi menjadi dua yaitu data *train* dan data *test*. Data *train* berisi 720 gambar yang terbagi menjadi 12 kelas. Sedangkan data *test* berisi 120 gambar sehingga tiap kelas memiliki 10 gambar. Hal yang diamati adalah keakuratan pencocokan gambar tempat wisata yang diperoleh dengan metode CNN.

Gambar tempat wisata yang diambil tidaklah menyeluruh. Dari satu tempat wisata hanya diambil satu sudut pandang untuk dikenali saat melakukan data *training*.

## 2.3 Tahapan Penelitian

### 2.3.1 Perancangan Sistem

Dalam perancangan sistem, dilakukan penyusunan alur dalam penggunaan sistem identifikasi tersebut. Penjabaran dari alur dijelaskan pada *flowchart* pada gambar 1.



**Gambar 1** Flowchart Sistem

Dalam sistem ini membutuhkan sebuah model untuk melakukan data training dan identifikasi. Rancangan arsitektur yang digunakan di ambil dari penelitian terdahulu oleh Dimas Setiyadi [3]. Rancangan tersebut dijabarkan pada tabel 1.

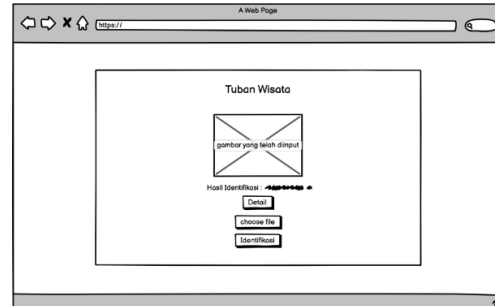
**Tabel 1** Arsitektur CNN

Proses	Parameter
Convolution	Filter = 64, kernel = (3,3), Padding = 1, Shape (48,48,1)
Batch Normalization	
Activation	relu
Max Pooling 2D	Pool_size = (2,2)
Dropout	0.25
Convolution	Filter = 64, kernel = (3,3), Padding = 1, Shape (48,48,1)

Batch Normalization	
Activation	relu
Max Pooling 2D	Pool_size = (2,2)
Dropout	0.25
Convolution	Filter = 64, kernel = (3,3), Padding = 1, Shape (48,48,1)
Batch Normalization	
Activation	relu
Max Pooling 2D	Pool_size = (2,2)
Dropout	0.25
Convolution	Filter = 64, kernel = (3,3), Padding = 1, Shape (48,48,1)
Batch Normalization	
Activation	relu
Max Pooling 2D	Pool_size = (2,2)
Dropout	0.25
Flatten	
Dennse	256
Batch Normalization	
Activation	relu
Dropout	0.25
Dense	Activation = softmax

### 2.3.1 Desain Sistem

Desain sistem dibuat menggunakan perangkat lunak Balsamiq Wireframes. Desain berisi tampilan mentah yang memperlihatkan konsep yang nantinya diterapkan ke sistem. Secara garis besar, desain yang digunakan digambarkan pada gambar 2.



Gambar 2 Desain Sistem

### 2.3.2 Skenario Pengujian

Metode yang digunakan untuk pengujian adalah dengan menguji keakuratan dari kecocokan gambar. Pengujian menggunakan 120 gambar yang terdiri dari masing-masing 10 gambar untuk 12 tempat wisata yang berbeda. Setelah mendapat hasil, maka akan diketahui perbandingan keakuratan tiap epoch

$$Acc = \frac{benar}{jumlah\ data} \times 100\%$$

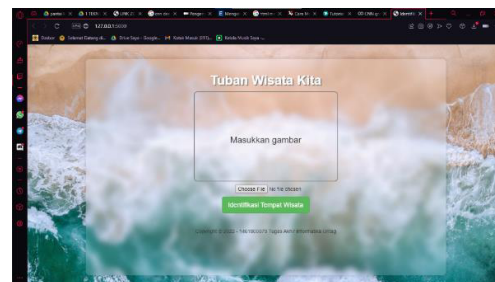
Dengan :

Acc = Akurasi

## 3. HASIL DAN PEMBAHASAN

### 3.1 Hasil

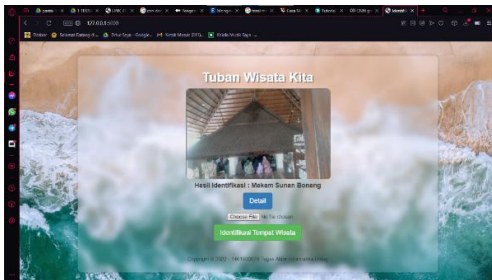
Program yang dihasilkan adalah sebuah sistem klasifikasi tempat wisata Kabupaten Tuban. Sistem ini dibuat dengan Bahasa Python menggunakan flask untuk dihubungkan ke html.



Gambar 3 Home

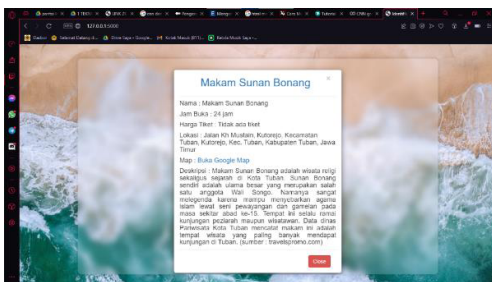
Halaman ini merupakan halaman awal yang berisi tombol untuk input data

serta tombol untuk memulai identifikasi. Kotak kosong yang ditampilkan adalah



Gambar 4 Hasil

Tidak ada banyak perubahan pada halaman ini. Gambar yang sebelumnya diinput akan diproses dan ditampilkan. Hasil identifikasi dan tombol detail akan muncul di bawahnya.



Gambar 5 Detail

Tombol detail jika ditekan akan memunculkan sebuah pop up berisi deskripsi tempat wisata. User dapat menutupnya dengan menekan tombol x atau tombol close di bawah. Setelah ditutup, user dapat melakukan identifikasi gambar lagi.

## 3.2 Pembahasan

### 3.2.1 Pengumpulan Data

Proses ini dilakukan dengan mengambil beberapa gambar dari internet dan mengunjungi tempat wisata di Kabupaten Tuban secara langsung. Data diambil dengan mengambil gambar dan menuliskan detail untuk deskripsi. Gambar yang diambil hanya dari satu sudut pandang saja.

tempat untuk menampilkan gambar yang diinput.

Data train terdiri dari total 720 gambar. Jumlah kelas adalah 12 yang terdiri dari Alun-alun Tuban, Makam Asmoroqondi, Goa Akbar, Klenteng Kwan Sing Bio, Makam Sunan Bonang, Masjid Agung Tuban, Museum Kambang Putih, Pantai Boom, Pantai Kelapa, Pantai Remen, Pemandian Bektiharjo, dan Pondok Pesantren Perut Bumi. Masing-masing kelas terdapat 60 data train.

### 3.2.2 Pembuatan Model Training

Proses ini dilakukan dengan menggunakan Bahasa Python di google colab. Dataset yang dikumpulkan dimasukkan ke google drive untuk diakses di google colab. Berikut adalah tahapan dalam pembuatan *model training*:

1. Upload dataset ke google drive.
2. Lakukan *mount drive* untuk menghubungkan ke direktori google drive.
3. Lakukan augmentasi terhadap gambar menggunakan fungsi `ImageDataGenerator`. Data kemudian di *rescale* menjadi  $1/255$ , dengan *rotation range* 30, *horizontal flip* untuk membalik gambar secara *horizontal*, *shear range* sebesar 0,3 serta *fill mode nearest* yaitu ketika ada area gambar yang tersisa saat pemrosesan, maka metode yang digunakan adalah mengisi area dengan pixel terdekat dan merentangkannya. Berikut merupakan contoh gambar yang telah melalui proses tersebut.
4. Tahap selanjutnya adalah melakukan perancangan layer. Model arsitektur yang digunakan diambil dari jurnal 'Deteksi Ekspresi Wajah Menggunakan Metode CNN'. Berikut adalah penjabarannya :

- a. Pada layer pertama terdapat layer konvolusi dengan jumlah layer konvolusi dengan jumlah filter 64. Kernel yang digunakan berukuran 3x3 dengan padding 1 dan fungsi aktivasi relu. Sementara *input shape* berukuran 150x150 (mengalami penyesuaian).
- b. Layer berikutnya adalah *batch normalization*. Layer ini digunakan untuk menormalkan aktivasi sebuah *volume input* sebelum meneruskannya ke lapisan berikutnya dalam jaringan.
- c. Berikutnya terdapat layer *Max Pooling 2D* yaitu untuk mereduksi ukuran gambar sebanyak mungkin. *Pooling* ini dalam bentuk matrix 2x2.
- d. Selanjutnya adalah *dropout*. Fungsi dari *dropout* adalah untuk mencegah masalah *overfitting*. cara kerjanya menonaktifkan beberapa *neuron* secara *random* yang tidak diperlukan. Nilai *dropout* yang digunakan adalah 0,25.
- e. Lapisan layer di atas diulang lagi sebanyak tiga kali.
- f. Kemudian terdapat layer *flatten*. Pada proses *flatten*, data *pooling* yang dimiliki berupa *array 2* dimensi kemudian dikonversi menjadi data satu dimensi.
- g. Lalu berikutnya adalah layer *batch normalization*.
- h. Layer selanjutnya adalah *dense*. *Dense* adalah layer dari *neuron*, yang setiap *neuron* menerima *input* dari semua *neuron* dari layer sebelumnya. Jumlah *neuron* yang digunakan adalah 256 dengan fungsi aktivasi relu.
- i. Setelah itu terdapat layer *dropout 0,25*.

- j. Dan layer terakhir adalah *dense* dengan fungsi aktivasi *softmax*. Fungsi ini digunakan karena jumlah kelas lebih dari dua (jumlah kelas = 12)

```

#MODEL CNN
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(64,(3,3), activation = 'relu',
        input_shape = (150,150,3),
        padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Dropout(0.25),

    tf.keras.layers.Conv2D(64,(3,3), activation = 'relu',
        padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Dropout(0.25),

    tf.keras.layers.Conv2D(64,(3,3), activation = 'relu',
        padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Dropout(0.25),

    tf.keras.layers.Conv2D(64,(3,3), activation = 'relu',
        padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Dropout(0.25),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(256, activation = 'relu'),
    tf.keras.layers.BatchNormalization(),

    tf.keras.layers.Dropout(0.25),
    tf.keras.layers.Dense(12, activation = 'softmax')
])

```

**Gambar 6** Susunan Layer

5. Setelah menyusun layer, maka tahap berikutnya adalah training. Pada proses ini menggunakan *step per epoch* sebanyak 20. Dari percobaan sebelumnya dengan *epoch 50, 100, dan 150*, didapatkan akurasi terbaik adalah *epoch 150*. Maka dari itu *epoch* yang digunakan untuk *training* adalah *epoch 150*. Parameter loss yang digunakan adalah *categorical\_crossentropy*. Parameter ini berguna untuk menentukan *loss function*. Berikutnya *optimizer* menggunakan ADAM. Parameter ini digunakan untuk menentukan algoritma *stochastic gradient descent*. Dan yang terakhir parameter *metrics* digunakan untuk menentukan performa *metric*.



### 3.2.3 Implementasi CNN ke Program

Dalam implementasi CNN, Bahasa yang digunakan adalah Python dan HTML. Detail dari tempat wisata disimpan di Microsoft Excel untuk dihubungkan ke Python.

Pertama-tama, Python dihubungkan ke HTML menggunakan library Flask. Di HTML akan meminta sebuah *input* berupa gambar yang ketika disubmit akan memicu *method POST*. Saat *method* ini terpicu, maka gambar yang telah diinput dan akan di simpan ke direktori yang disiapkan sebelumnya.



Gambar yang didapat lalu diproses dengan menggunakan model yang telah disimpan dari data *training* sebelumnya. Dari proses ini dapat menghasilkan nilai yang nantinya digunakan untuk menentukan nama dari tempat wisata tersebut. Selain itu, nilai yang diperoleh juga digunakan untuk memanggil deskripsi yang telah disimpan di Microsoft Excel menggunakan library Pandas.

Data yang dihasilkan dari proses sebelumnya kemudian dipanggil kembali untuk ditampilkan lewat HTML. Hal yang ditampilkan berupa gambar yang telah diinput, beserta hasil identifikasi dan deskripsinya.

### 3.2.4 Pengujian

Proses pengujian dilakukan terhadap 12 kelas. Setiap kelas terdapat 10 gambar yang diujikan, sehingga total gambar yang diujikan adalah 120 gambar. Berikut adalah perhitungan Confusion Matrixnya

Tabel 2 Pengujian Alun-alun Tuban

No	Gambar	Aktual	Prediksi	Ket
1		Alun-alun Tuban	Alun-alun Tuban	Benar
2		Alun-alun Tuban	Pantai Kelapa	Salah

3		Alun-alun Tuban	Alun-alun Tuban	Benar
4		Alun-alun Tuban	Pantai Remen	Salah
5		Alun-alun Tuban	Pantai Remen	Salah
6		Alun-alun Tuban	Alun-alun Tuban	Benar
7		Alun-alun Tuban	Alun-alun Tuban	Benar
8		Alun-alun Tuban	Alun-alun Tuban	Benar
9		Alun-alun Tuban	Alun-alun Tuban	Benar
10		Alun-alun Tuban	Alun-alun Tuban	Benar
<b>Total Benar</b>		7		

Pengujian dilakukan terhadap 12 tempat wisata. Untuk memudahkan, maka 11 tempat wisata yang belum dijabarkan disusun ke dalam *confusion matrix*. Hal ini agar perhitungan akurasi, precision, recall dan F1 score dapat dilakukan dengan efisien.

Tabel 3 Confusion Matrix

		Prediction											
		a	b	c	d	e	f	g	h	i	j	k	l
Actual	a	7								1	2		
	b		9							1			
	c			1	1						8		
	d				8			1			1		
	e					1	0						
	f		2				8						
	g				1			7			2		
	h			1					2	1	6		
	i		2						1	7			

j									1		
k			1						0	4	5
l						2			2		6

Keterangan :

1. a = Alun-alun Tuban
2. b = Makam Asmoroqondi
3. c = Goa Akbar
4. d = Klenteng Kwan Sing Bio
5. e = Makam Sunan Bonang
6. f = Masjid Agung Tuban
7. g = Museum Kambang Putih
8. h = Pantai Boom
9. i = Pantai Kelapa
10. j = Pantai Remen
11. k = Pemandian Bektiharjo
12. l = Pondok Pesantren Perut Bumi

### 3.2.5 Perhitungan Akurasi

Proses perhitungan akurasi dilakukan setelah pengujian. Total gambar yang diujikan sebanyak 120 buah. Gambar yang diujikan menyesuaikan dengan lokasi yang digunakan saat data training, karena jika menggunakan di luar itu maka gambar tidak dapat di kenali.

Dengan menggunakan nilai pada confusion matrix, maka dapat dihitung *precision*, *recall*, *F1 score* dan *accuracy* dari hasil pengujian.

**Tabel 4 Precision**

	TP	FP	Precision	All Precision
a	7	3	0,7	66,70%
b	9	1	0,9	
c	1	9	0,1	
d	8	2	0,8	
e	10	0	1	
f	8	2	0,8	
g	7	3	0,7	
h	2	8	0,2	
i	7	3	0,7	
j	10	0	1	
k	5	5	0,5	

l	6	4	0,6
---	---	---	-----

$$P = \frac{TP}{TP + FP}$$

$$All\ Precision = \frac{\sum_{i=1}^{12} P_i}{Jumlah\ Kelas}$$

Dengan:

- P = Precision
- TP = True Positive
- FP = False Positive

Dalam perhitungan di tabel 4, rumus yang digunakan adalah persamaan di atas. Sehingga dari nilai *True Positive* dan *False Positive*, diperoleh *precision* sebesar 66,7%.

**Tabel 5 Recall**

	TP	FN	Recall	All Recall
a	7	0	1	79,50%
b	9	4	0,7	
c	1	1	0,5	
d	8	3	0,7	
e	10	0	1	
f	8	0	1	
g	7	4	0,6	
h	2	0	1	
i	7	3	0,7	
j	10	25	0,3	
k	5	0	1	
l	6	0	1	

$$R = \frac{TP}{TP + FN}$$

$$All\ Recall = \frac{\sum_{i=1}^{12} R_i}{Amount\ of\ Class}$$

Dengan:

- R = Recall
- TP = True Positive
- FN = False Negative

Dalam perhitungan di tabel 5, rumus yang digunakan adalah persamaan di atas. Sehingga dari nilai *True Positive* dan *False Negative*, diperoleh *recall* sebesar



66,7%. Selanjutnya dilakukan perhitungan *F1 Score* dan *Accuracy*.

$$F1\ Score = \frac{2 \times All\ Precision \times All\ Recall}{All\ Precision + All\ Recall}$$

$$F1\ Score = \frac{2 \times 66,7\% \times 79,5\%}{66,7\% + 79,5\%}$$

$$F1\ Score = 72,5\%$$

$$Accuracy = \frac{\sum_{i=1}^n TP_i}{Amount\ of\ Data}$$

$$Accuracy = \frac{(7 + 9 + 1 + 8 + 10 + 8 + 7 + 2 + 7 + 10 + 5 + 6)}{120}$$

$$Accuracy = \frac{80}{120} \times 100\% = 66,7\%$$

Dengan demikian, nilai *precision* yang diperoleh adalah 66,7%. Lalu untuk nilai *recall* 79,5%. Dari nilai *precision* dan *recall* dapat diperoleh *F1 score* yaitu 72,5%. Lalu yang terakhir nilai akurasi yang didapat adalah 66,7%

#### 4. SIMPULAN

Berdasarkan pengujian dan perhitungan yang telah dilakukan, diperoleh kesimpulan sebagai berikut :

1. Dari hasil perhitungan, didapatkan nilai *precision* 66,7%, *recall* 79,5%, *F1 score* 72,5% dan akurasi sebesar 66,7%.
2. Sudut pandang pengambilan gambar sangat berpengaruh dalam melakukan proses identifikasi. Dalam pengujian, terdapat beberapa kelas yang memiliki nilai akurasi yang rendah. Gambar-gambar dalam kelas tersebut memiliki sudut pengambilan gambar yang kurang variatif. Hal ini menyebabkan sistem sulit

mengenali gambar yang diambil dari sudut pandang yang berbeda.

3. Warna gambar mempengaruhi hasil identifikasi. Dapat dilihat di beberapa gambar yang cukup mirip namun yang satu dapat dikenali dan yang lainnya tidak. Hal yang membedakan di antara gambar tersebut adalah warnanya. Jadi warna dapat memberi pengaruh terhadap proses identifikasi.
4. Variasi gambar memiliki pengaruh dalam pengenalan gambar. Apabila dalam suatu kelas terdapat gambar dengan ciri khas yang berbeda, misal gambar pantai dan pohon kelapa, maka akan mengacaukan proses identifikasi.

#### 5. DAFTAR PUSTAKA

- [1] Ilahiyah, Sarirotul dan Agung Nilogiri. 2018. Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. Jurnal Sistem & Teknologi Informasi Indonesia, 3(2), pp. 49-56.
- [2] Arsal, M. et al. 2020. Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN. Jurnal Nasional Teknologi & Sistem Informasi, 6(1), pp. 55-63.
- [3] Setiyadi, Dimas. (2020) Deteksi Ekspresi Wajah Menggunakan Metode CNN. Universitas 17 Agustus 1945 Surabaya.