











Author : Ragil Hadi Prasetyo

## 1. Dataset

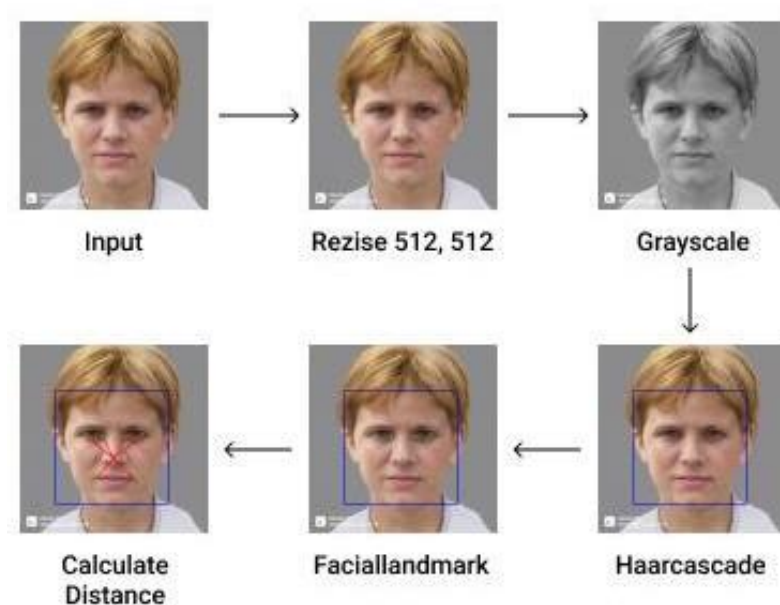
Dari dataset yang diberikan terdapat 4 *person* yang berbeda yang disesuaikan dengan penamaan filenya. Kemudian membagi dataset tersebut dengan menggunakan 1 gambar setiap *person* untuk dijadikan sebagai training data dan sisanya menjadi testing data. Berikut merupakan pembagian dataset yang dapat dilihat pada **Tabel 1**.

**Tabel 1 Pembagian Dataset**

Id	Training	Testing		
A				
B				
C				
D				

## 2. Preprocessing

Proses preprocessing dimulai dengan melakukan resize gambar menjadi 512 untuk lebar dan 512 untuk tinggi. Resize dilakukan agar semua gambar memiliki ukuran yang sama. Selanjutnya dilakukan proses grayscale gambar yaitu mengubah gambar yang sebelumnya memiliki 3 channel menjadi 1 channel saja. Proses selanjutnya yaitu deteksi wajah menggunakan haarcascade. Proses selanjutnya yaitu mendeteksi beberapa titik pada wajah menggunakan facial landmark. Beberapa titik pada wajah digunakan dan dihitung jaraknya sehingga mendapatkan fitur. Berikut merupakan proses preprocessing yang dapat dilihat pada **Gambar 1**.



**Gambar 1 Proses Preprocessing Gambar**

## 3. Ekstraksi Ciri

Ekstraksi ciri yang dilakukan pada gambar yaitu menghitung jarak beberapa titik pada wajah dengan titik di bagian hidung. Ekstraksi ciri ini dilakukan karena jarak beberapa titik pada wajah dengan hidung pada setiap manusia memiliki perbedaan. Oleh karena itu ekstraksi ciri ini dilakukan dan akan digunakan sebagai fitur untuk rekognisi wajah manusia. Berikut merupakan visualisasi dari perhitungan jarak titik untuk mendapatkan fitur yang dapat dilihat pada **Gambar 2**.



**Gambar 2 Visualisasi Perhitungan Jarak Pada Beberapa Titik**

Dalam menghitung jarak antara 2 titik menggunakan Euclidian Distance. Euclidian distance merupakan salah satu cara menghitung jarak antara 2 titik koordinat. Berikut merupakan formula dari Euclidian Distance pada **Gambar 3**.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

$\mathbf{p}, \mathbf{q}$  = two points in Euclidean n-space

$q_i, p_i$  = Euclidean vectors, starting from the origin of the space (initial point)

$n$  = n-space

**Gambar 3 Formula Euclidian Distance**

Proses penghitungan jarak menggunakan Euclidian distance dilakukan pada 6 titik pada wajah dan menghasilkan 6 fitur. Berikut merupakan hasil perhitungan jarak menggunakan Euclidian distance yang dapat dilihat pada **Gambar 4**.

	feature 1	feature 2	feature 3	feature 4	feature 5	feature 6	id
0	103.855669	68.425142	69.354164	106.212052	111.400180	54.000000	0
1	103.004854	66.037868	66.573268	102.415819	119.883277	58.549125	1
2	97.005155	68.095521	68.264193	99.201814	101.518471	46.000000	2
3	103.947102	77.472576	80.498447	110.344914	109.201648	48.466483	3

**Gambar 4 Hasil Perhitungan Jarak**

#### 4. Machine Learning

Algoritma machine learning yang digunakan yaitu K-Nearest Neighbours (KNN). KNN digunakan karena KNN mencari data tetangga terdekat dengan data input. KNN juga tidak memerlukan data training yang banyak untuk mendapatkan akurasi yang baik. KNN sebelumnya dilakukan training dengan menggunakan dataset yang telah dibuat dan disimpan modelnya dengan menggunakan library pickle. Parameter yang digunakan pada model ini yaitu jumlah  $K = 1$  dikarenakan dataset yang digunakan sangat sedikit sehingga pemilihan angka 1 adalah yang terbaik. Berikut merupakan waktu yang diperlukan pada saat training data menggunakan KNN yang dapat dilihat pada **Tabel 2**.

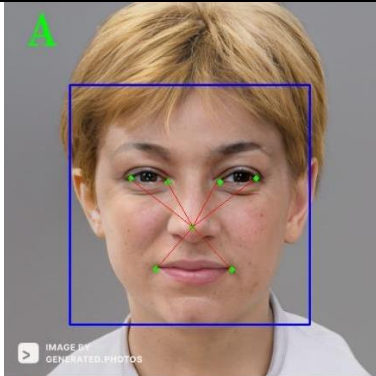
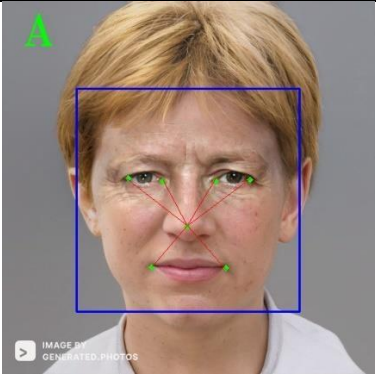
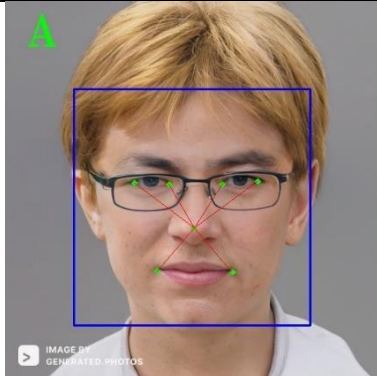
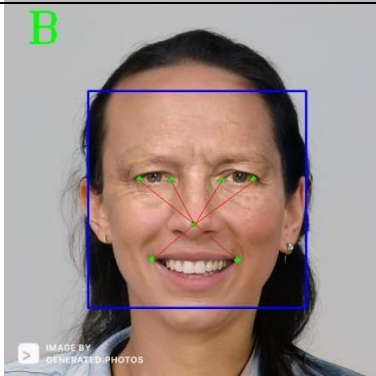
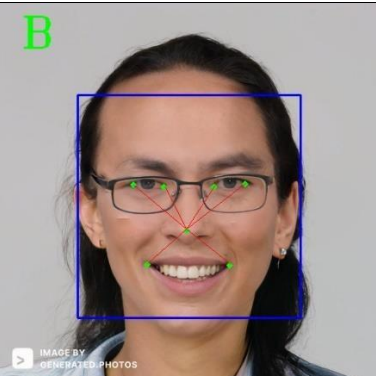
**Tabel 2 Waktu Training KNN**

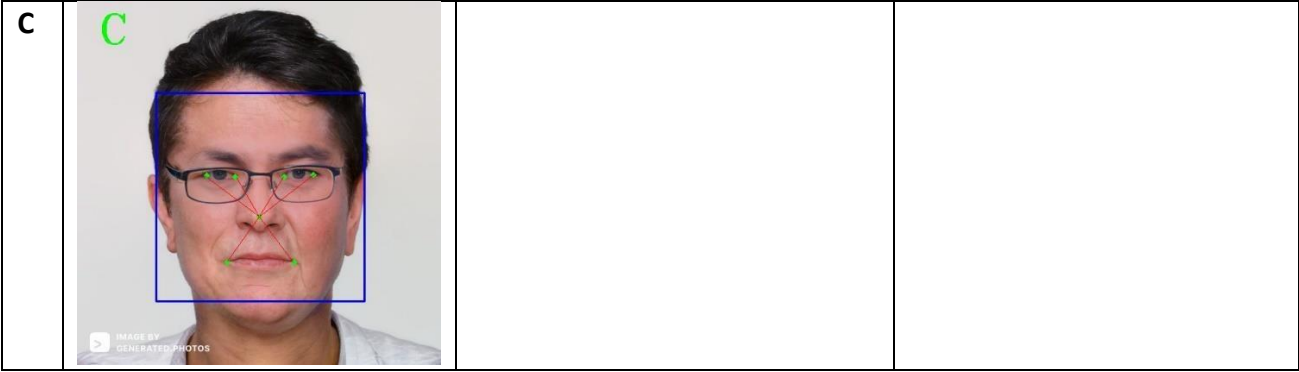
Waktu Training Data	0.03600335121154785 S
---------------------	-----------------------

#### 5. Output

Setelah dilakukan training model maka model tersebut akan digunakan pada test data. Hasil model ditampilkan pada gambar pada bagian kiri atas. Berikut hasil model pada testing data yang dapat dilihat pada **Tabel 3**

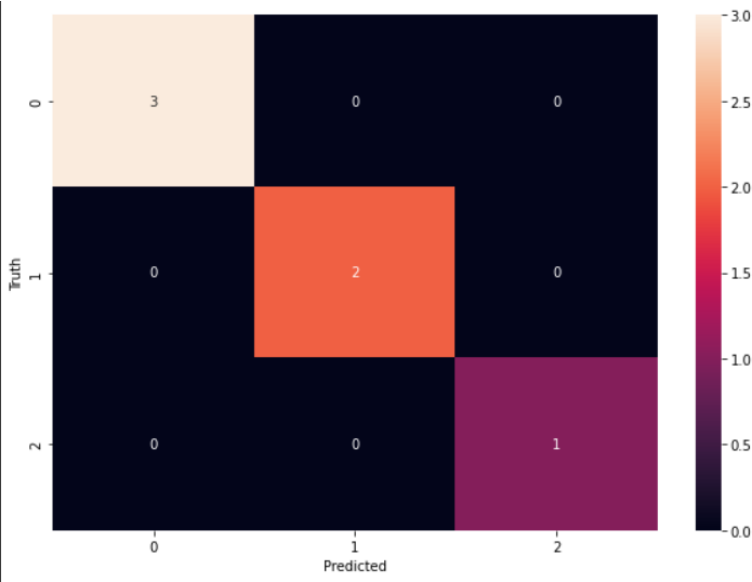
**Tabel 3 Hasil Model Pada Testing Data**

Id	Testing		
A			
			



**6. Evaluasi**

Setelah model dicoba pada testing data kemudian dilakukan evaluasi untuk menentukan performa dari model. Berikut merupakan beberapa evation metrics untuk mengukur performa model.



**Gambar 5 Confussion Matrix**

	precision	recall	f1-score	support
0	1.00	1.00	1.00	3
1	1.00	1.00	1.00	2
2	1.00	1.00	1.00	1
accuracy			1.00	6
macro avg	1.00	1.00	1.00	6
weighted avg	1.00	1.00	1.00	6

**Gambar 6 Classification Report**

**Tabel 4 Rata-Rata Waktu Komputasi Klasifikasi KNN**

<b>No</b>	<b>Filename</b>	<b>Waktu Komputasi</b>
<b>1</b>	AB	0.0010008811950683594 S
<b>2</b>	AC	0.0009984970092773438 S
<b>3</b>	AD	0.0009617805480957031 S
<b>4</b>	BC	0.002015829086303711 S
<b>5</b>	BD	0.0010027885437011719 S
<b>6</b>	CD	0.0010006427764892578 S
<b>Rata-Rata Waktu Komputasi</b>		<b>0.0011634031931559246 S</b>