

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian Pertama

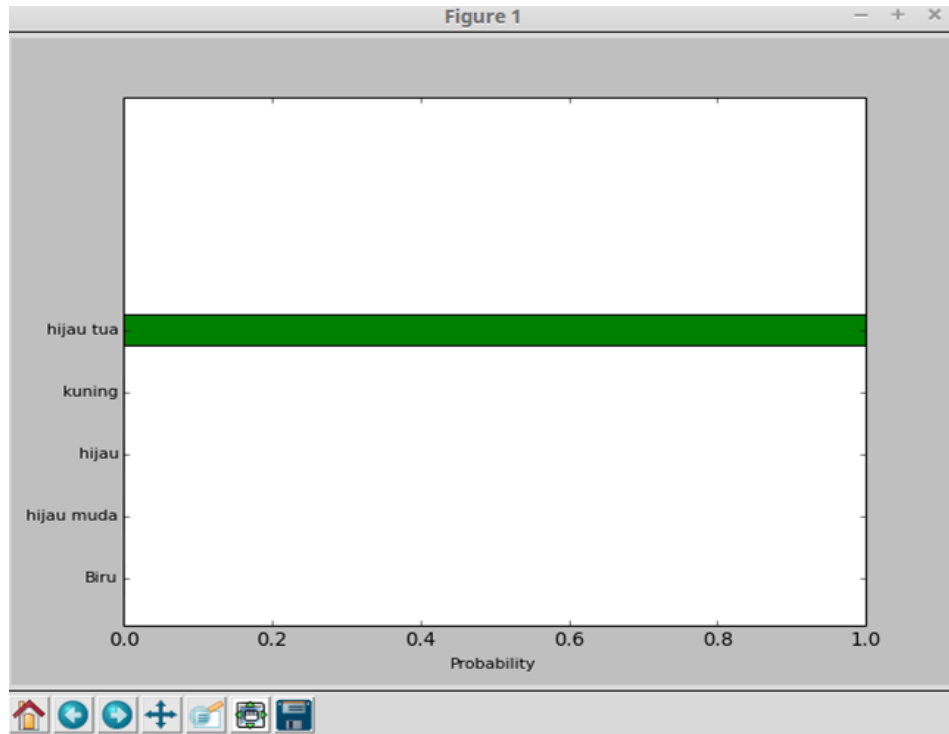
Berdasarkan perancangan yang dilakukan sebelumnya, maka pengujian yang dilakukan adalah menggunakan warna-warna berikut sebagai data model yang akan disimpan di dalam database:



Gambar 4.1 Warna Model

Selanjutnya pengujian perbandingan warna dilakukan menggunakan dua metode yaitu intersection dan correlation. Berikut hasil pengujian yang didapat:

1. Hasil Menggunakan Metode Intersection



Gambar 4.2 Hasil Histogram Hijau Tua Metode *Intersection*

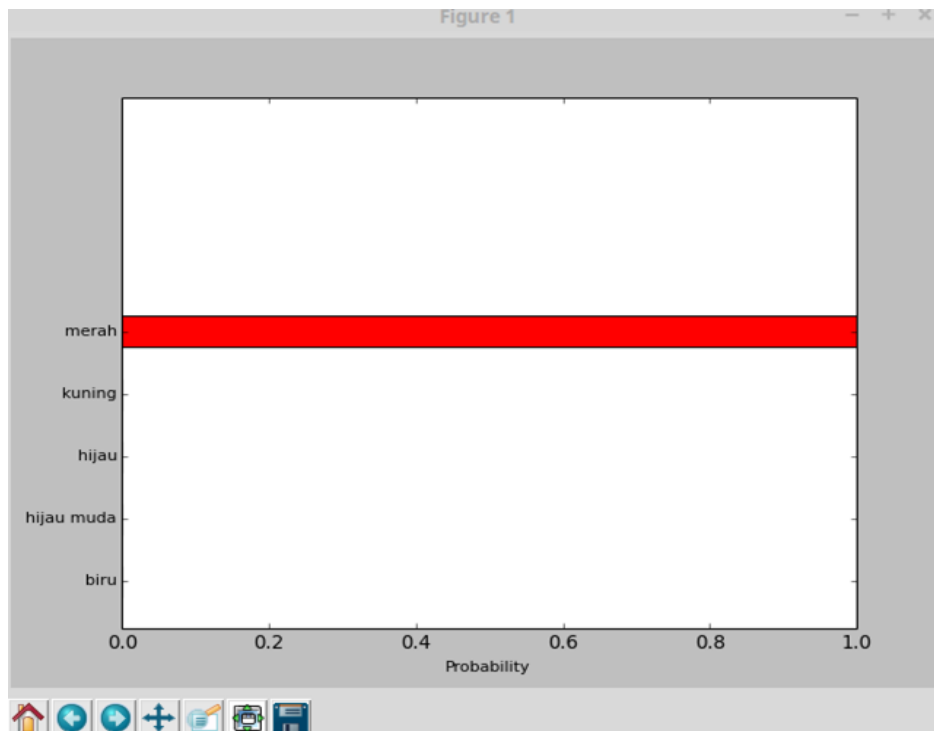
```
Terminal
agem@gagem ~/Documents/SKRIPSI/SCRIPT/Komparasi $ python tesComparasion.py
Comparison Array:
[0. 0. 0. 0. 5.34065479]
Distribution Array:
[0. 0. 0. 0. 1.]
```

Gambar 4.3 Nilai Array Hijau Tua Metode *Intersection*

Pengujian pertama yang dilakukan ini menggunakan gambar warna hijau tua sebagai data uji dan 5 data model yaitu biru, hijau muda, hijau, hijau tua, dan kuning. Seperti dilihat pada gambar, setelah melalui pemrosesan menggunakan program openCV, warna hijau tua dibandingkan dengan data model sehingga menampilkan hasil histogram di channel hijau tua seperti yang tertera pada gambar. Selain itu, nilai yang tertera dari hasil pengujian adalah 1 untuk hijau tua dan 0 untuk warna lainnya, hal ini menunjukkan keserasian antar data yang diuji dengan data model hijau tua seperti yang dijelaskan di dalam tabel distance value. Kemudian pengujian selanjutnya yang dilakukan adalah pengujian dengan menggunakan dua gambar yang mirip, berikut hasil percobannya:



Gambar 4.4 Uji Semangka



Gambar 4.5 Hasil Histogram Semangka Metode *Intersection*

```

Terminal
agem@agem ~ $ cd Documents
agem@agem ~/Documents $ cd SKRIPSI/SCRIPT/Komparasi
agem@agem ~/Documents/SKRIPSI/SCRIPT/Komparasi $ python tesComparasion.py
Comparison Array:
[0. 0. 0. 0. 0.00047003]
Distribution Array:
[0. 0. 0. 0. 1.]

```

Gambar 4.6 Nilai Array Semangka Metode *Intersection*

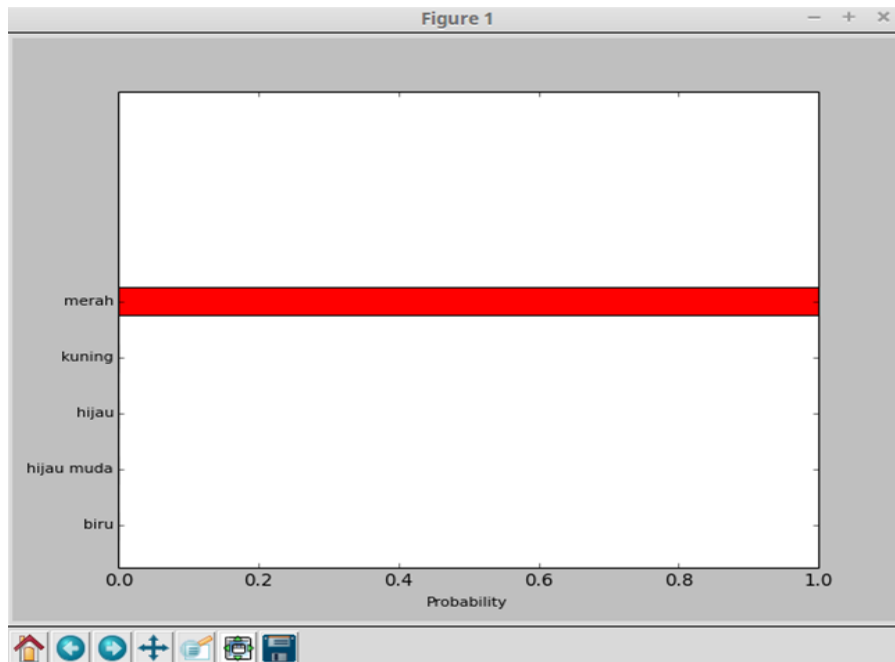
Pengujian ini menggunakan model data uji sebuah gambar semangka. Hasil yang didapat seperti terlihat pada gambar histogram di atas, bahwa gambar

semangka yang dibaca menggunakan metode intersection berhasil dibaca oleh data model warna merah. Hal ini terjadi karena metode intersection bekerja dengan cara mengidentifikasi warna terkuat di dalam gambar semangka tersebut. Bila dilihat dari array yang ditampilkan, nilai yang ditunjukkan untuk channel warna merah adalah 1 yang berarti sangat cocok, sedangkan untuk 4 warna lainnya nilainya 0 atau sama sekali tidak cocok.

2. Hasil Menggunakan Metode Correlation

Pengujian kali ini menggunakan gambar warna merah sebagai data uji untuk diklasifikasikan warnanya. Berbeda dari pengujian sebelumnya, kali ini metode yang digunakan untuk mengklasifikasikan warna adalah metode correlation (korelasi). Metode ini akan mencari hubungan antar variabel pada gambar yang diuji dan gambar model.

Hasil pengujian pertama yang diperoleh seperti terlihat pada gambar di bawah, bahwa histogram yang nilainya penuh ditunjukkan oleh channel merah. Hal tersebut menandakan program dapat membaca data uji dengan baik. Selain hasil dari histogram, dapat dilihat pula hasil array yang ditampilkan pada terminal menunjukkan nilai 1.0 dimana jika merujuk pada tabel distance value, nilai 1.0 pada metode correlation memiliki arti exact match atau benar benar cocok.



Gambar 4.7 Hasil Histogram Merah Metode *Correlation*

```

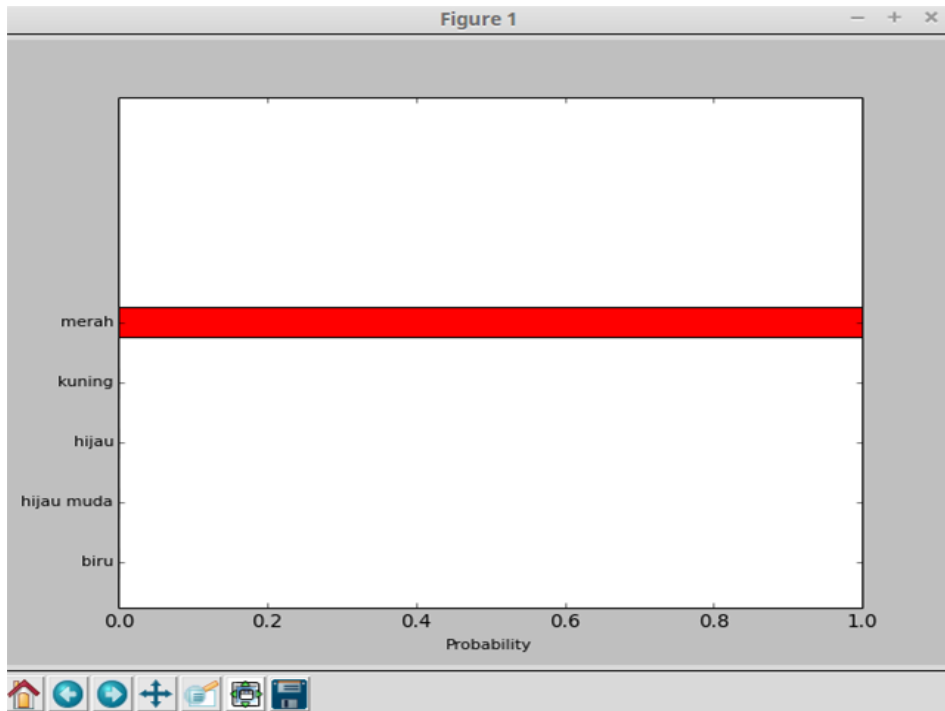
Terminal
agem@agem ~/Documents/SKRIPSI/SCRIPT/Komparasi $ python tesComparasion.py
Comparison Array:
[-4.76837386e-07 -4.76837386e-07 -4.76837386e-07 -4.76837386e-07
 1.00000000e+00]
Distribution Array:
[-4.76838295e-07 -4.76838295e-07 -4.76838295e-07 -4.76838295e-07
 1.00000191e+00]

```

Gambar 4.8 Nilai Array Merah Metode *Correlation*

Menggunakan warna merah sebagai data uji merupakan sesuatu yang mudah diuji karena warna merah pada data uji jelas atau khusus (nilai rgb nya berbeda dengan data model lainnya). Untuk itu, pengujian kedua akan dilakukan

pengujian menggunakan data uji yang lebih rumit. Berikut hasil pengujian kedua menggunakan metode correlation:



Gambar 4.9 Hasil Histogram Semangka Metode *Correlation*

```
Terminal
agem@agem ~/Documents/SKRIPSI/SCRIPT/Komparasi $ python tesComparasion.py
Comparison Array:
[-3.44685881e-05 -3.44685881e-05 -3.44685881e-05 -3.44685881e-05
 4.36142534e-04]
Distribution Array:
[-0.1155624 -0.1155624 -0.1155624 -0.1155624 1.46224962]

```

Gambar 4.10 Nilai Array Model Uji Semangka Metode *Correlation*

Seperti hasil pengujian pada metode intersection sebelumnya, pengujian menggunakan metode correlation ini menggunakan data uji gambar 21 di atas dan menghasilkan hasil klasifikasi warna yang sama. Hal tersebut ditunjukkan dengan tampilan histogram yang bernilai 1 pada channel merah dimana nilai 1 pada correlation juga berarti hasil data uji dapat dikatakan exact match atau sangat cocok. Meskipun data yang ditampilkan histogram untuk metode correlation dan intersection pengujian gambar semangka ini sama, namun komposisi array yang dihasilkan oleh data model berbeda. Berikut tabel perbandingan hasil array untuk kedua pengujian:

Tabel 4.1 Perbandingan Nilai *Intersection* dan *Correlation*

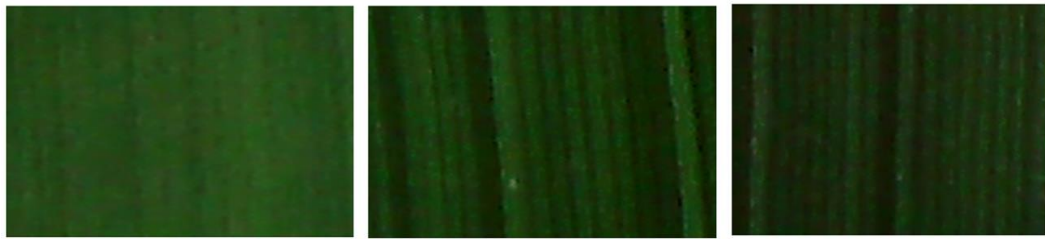
Warna	Intersection		Correlation	
	comparison	distribution	comparison	distribution
Biru	0.	0.	-3.44685881	-0.1155624
Hijau muda	0.	0.	-3.44685882	-0.1155625
Hijau	0.	0.	-3.44685883	-0.1155626
Kuning	0.	0.	-3.44685884	-0.1155627
Merah	0.00047003	1.	4.36142534	1.46224962

4.2 Pengujian kedua

Pengujian kedua ini tetap menggunakan dua metode namun dilakukan pada BWD sebagai model yang diuji. Berikut hasil pengujian:

4.2.1 Intersection

Pengujian kali ini menggunakan data model gambar berupa warna warna hijau daun padi. Berikut tampilan warna yang telah didapat:



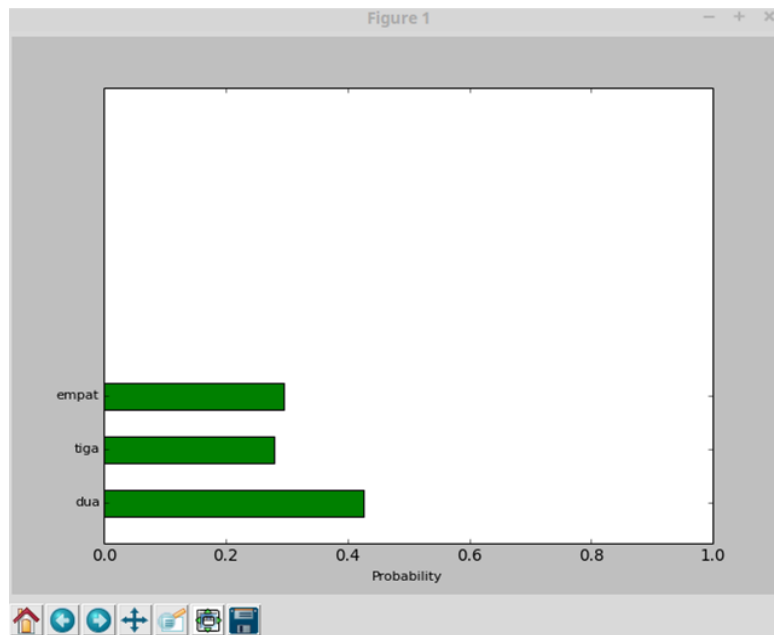
Gambar 4.11 Model Daun Padi

Tiga gambar tersebut (daun2, daun3, daun4) akan dimuat sebagai model di dalam OpenCV dimana nantinya akan dipanggil sebagai pembanding yang digunakan untuk mengklasifikasikan warna dari gambar uji. Pembacaan dan pemanggilan untuk tiap gambar tersebut akan dilakukan oleh program di dalam OpenCV. Sebelum dioperasikan, tiga gambar tersebut harus di-*crop* atau dipotong terlebih dahulu sehingga warna putih pada *background* gambar daun hilang.



Gambar 4.12 Uji Daun Padi

Selain tiga gambar model daun tersebut, pengujian kali ini akan menggunakan tiga gambar uji (Gambar 4.12) untuk nanti diklasifikasikan menurut warnanya. Ketiga gambar uji tersebut didapat dari hasil *capture* warna BWD yang kemudian akan diberi nama sesuai klasifikasi warna pada BWD. Nama gambar uji ini juga disamakan dengan nama gambar model yaitu BWD2, BWD3, dan BWD4. Pada pengujian pertama, penulis akan menggunakan metode *intersection* dengan data dari gambar BWD2, berikut hasil yang didapatkan:



Gambar 4.13 Hasil Histogram BWD 2 Metode *Intersection*

Seperti terlihat dari hasil histogram di atas (Gambar 4.13), perbandingan yang dilakukan berhasil. Untuk lebih jelasnya dapat dilihat nilai yang dihasilkan oleh comparison array dan distribution array seperti berikut:

```

Terminal
agem@agem ~ $ cd Desktop
agem@agem ~/Desktop $ python l.py
Comparison Array:
[2.40467517 1.57560261 1.67082711]
Distribution Array:
[0.42552301 0.2788132 0.29566379]

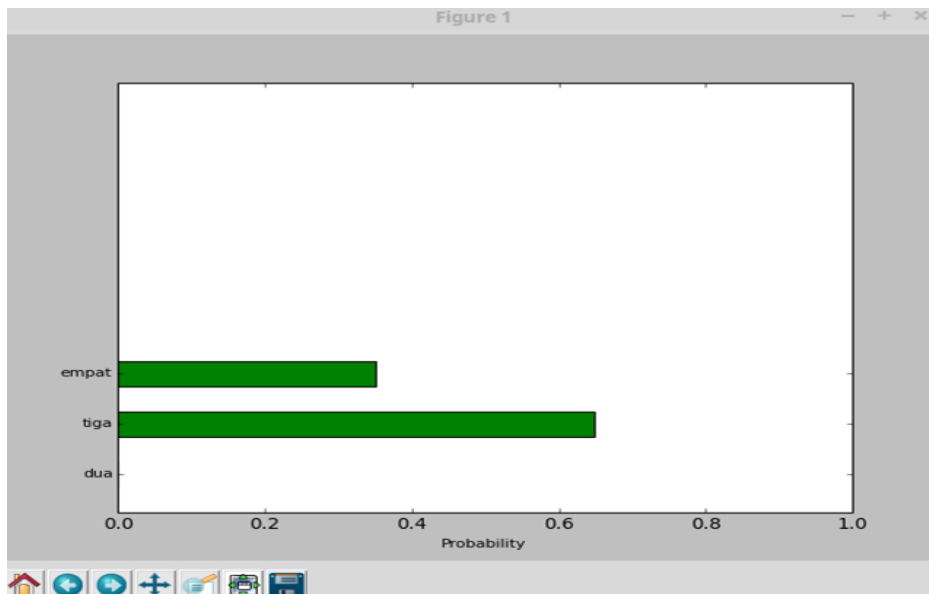
```

Gambar 4.14 Nilai Array Model Uji BWD 2 Metode *Intersection*

Nilai yang ditunjukkan oleh *bin* dua adalah nilai terbesar, dimana hal ini menunjukkan tingkat kecocokan yang paling tinggi ada pada gambar dua. Untuk

lebih membuktikan tingkat kecocokan yang dihasilkan, maka bisa dilihat dari nilai *distribution array* yang ditampilkan.

Setelah pengujian terhadap gambar dua, pengujian berlanjut ke gambar tiga. Berikut hasil pengujian yang didapat:



Gambar 4.15 Hasil Histogram BWD 3 Metode *Intersection*

```
Terminal
agem@agem ~/Desktop $ python l.py
Comparison Array:
[0. 0.72652864 0.39336484]
Distribution Array:
[0. 0.64874799 0.35125201]
```

Gambar 4.16 Nilai Array Model Uji BWD 3 Metode *Intersection*

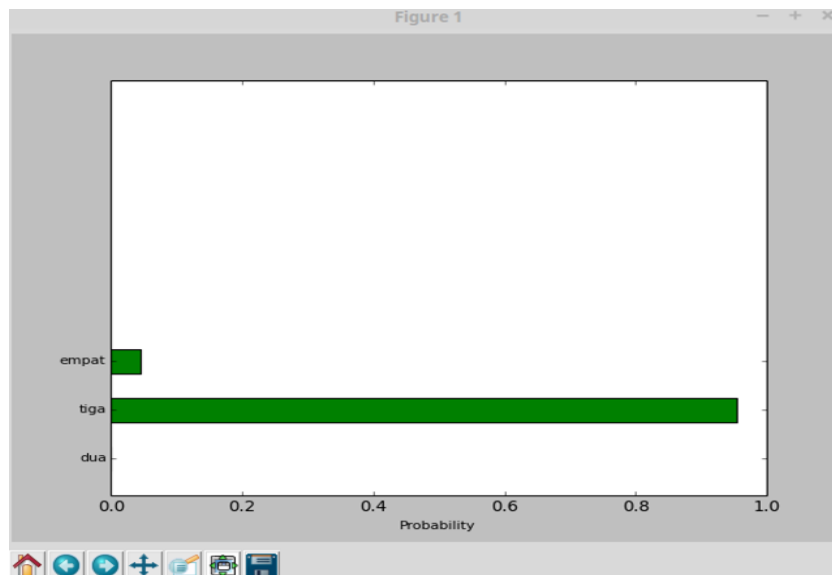
Seperti pengujian sebelumnya, pengujian kali ini berhasil dengan klasifikasi warna BWD 3 cocok terhadap data model daun3. Bila dilihat dari nilai *array*-nya, terdapat sedikit perbedaan antara *comparison array* dengan *distribution array*.

Comparison array merupakan nilai awal atau biasa disebut dengan *raw distance* dimana dengan nilai ini, kita belum bisa menentukan apakah data gambar yang dibandingkan cocok atau tidak.

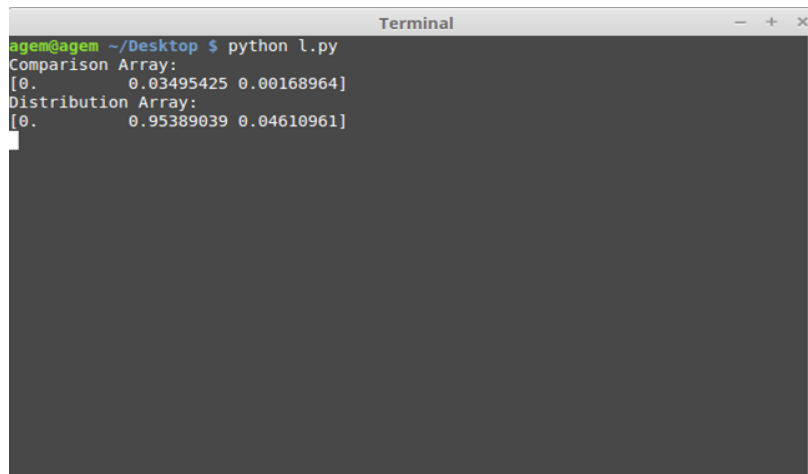
Untuk dapat mengatakan suatu data kemungkinan cocok, maka nilai dari *comparison array* harus diproses terlebih dahulu. Pemrosesan ini dilakukan untuk menormalisasikan *array* yang ada dengan cara membaginya menggunakan jumlah nilai *comparison array* itu sendiri. Berikut bentuk lebih jelasnya:

```
probability_array = comparison_array / np.sum(comparison_array)
```

Melihat dari hasil yang didapat setelah dilakukan perhitungan, gambar uji pengujian kedua ini (gambar BWD3) dapat diklasifikasikan sebagai data model nomer 3. Setelah dua pengujian tersebut, dilakukan pengujian terakhir dengan data uji gambar BWD4. Berikut hasil yang didapatkan:



Gambar 4.17 Hasil Histogram BWD 4 Metode *Intersection*



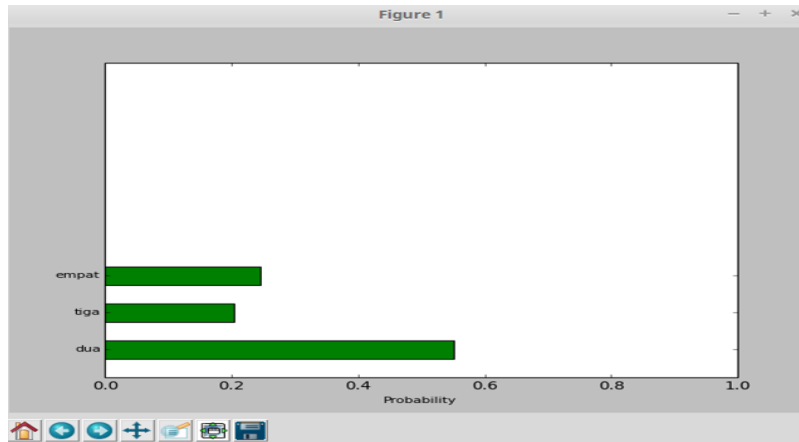
```
Terminal
agem@agem ~/Desktop $ python l.py
Comparison Array:
[0.      0.03495425 0.00168964]
Distribution Array:
[0.      0.95389039 0.04610961]
```

Gambar 4.18 Nilai Array Model Uji BWD 4 Metode *Intersection*

Seperti terlihat pada gambar histogram tersebut, proses klasifikasi gambar uji BWD4 ini tidak berjalan lancar seperti pengujian sebelumnya. Data gambar BWD4 yang diujikan terbaca oleh program sebagai gambar daun3. Kegagalan yang terjadi menandakan hubungan antar citra yang diuji dengan gambar model yang telah diproses menjadi data model tidak memenuhi syarat-syarat metode *intersection*.

4.2.2 Correlation

Pengujian dengan menggunakan metode *correlation* pada daun padi ini menggunakan data uji dan data model yang sama dengan pengujian sebelumnya. Berikut hasil pengujian klasifikasi BWD2 yang menggunakan metode *correlation*:



Gambar 4.19 Hasil Histogram BWD 2 Metode *Correlation*

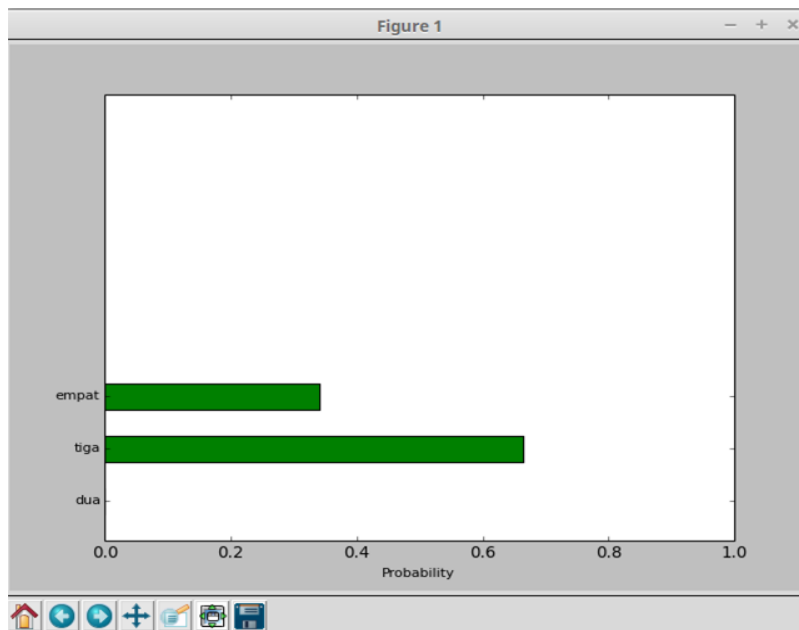
```

Terminal
agem@agem ~ $ cd Desktop
agem@agem ~/Desktop $ python l.py
Comparison Array:
[0.10423716 0.03844317 0.04626303]
Distribution Array:
[0.55168469 0.20346397 0.24485133]

```

Gambar 4.20 Nilai Array Model Uji BWD 2 Metode *Correlation*

Data uji pertama yang diujikan menggunakan metode *correlation* ini adalah BWD2, dengan hasil seperti terlihat pada histogram bahwa gambar yang diujikan berhasil diklasifikasikan sebagai data model daun2. Hal tersebut juga terlihat dari nilai *array* yang ada sebesar 0.55 merupakan nilai terbesar, dimana data model warna dua berarti menjadi warna yang paling cocok dengan data uji gambar daun2. Pengujian selanjutnya akan dicoba menggunakan BWD3, berikut hasilnya:



Gambar 4.21 Hasil Histogram BWD 3 Metode *Correlation*

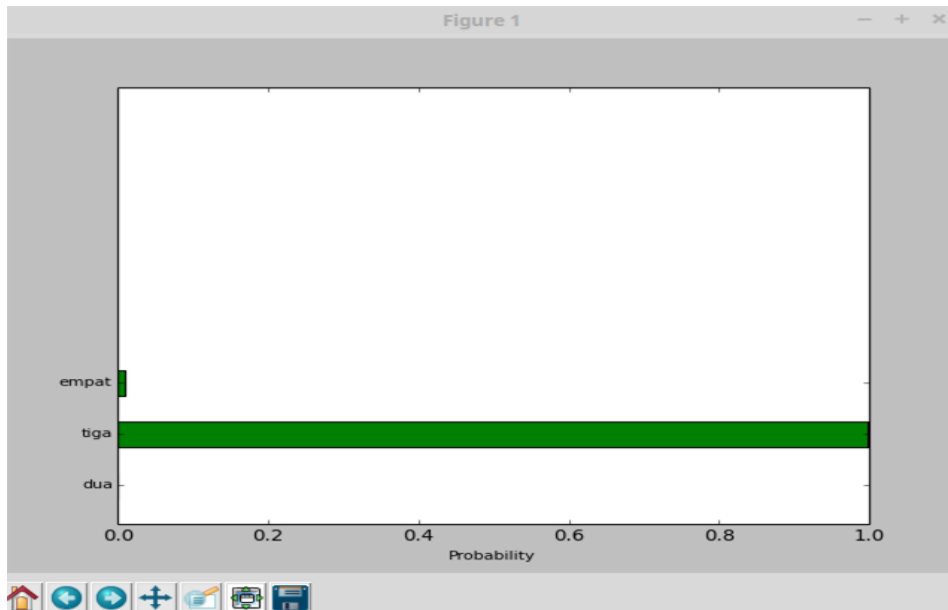
```

Terminal
agem@gagem ~/Desktop $ python l.py
Comparison Array:
[-0.00011536  0.01463246  0.00753347]
Distribution Array:
[-0.00523177  0.6635864  0.34164537]

```

Gambar 4.22 Nilai Array Model Uji BWD 3 Metode *Correlation*

Hasil pengujian di atas menunjukkan adanya tingkat kecocokan data uji BWD3 dengan data model daun3. Kecocokan dapat dilihat pula dari hasil *distribution array* yang ditampilkan, bahwa nilai yang dihasilkan sebesar 0.66 dan nilai tersebut lebih besar dari dua data yang lain.



Gambar 4.23 Hasil Histogram BWD 4 Metode *Correlation*

```

Terminal
agem@agem ~/Desktop $ python l.py
Comparison Array:
[-2.99641958e-05  4.06214664e-03  3.55573235e-05]
Distribution Array:
[-0.0073663  0.998625  0.0087413]

```

Gambar 4.24 Nilai Array Model Uji BWD 4 Metode *Correlation*

Pengujian selanjutnya menggunakan gambar uji BWD4, namun setelah diproses menggunakan metode *correlation*, tingkat kecocokan yang dihasilkan paling tinggi ada pada data uji daun3, bukan daun4. Untuk lebih jelasnya dapat dilihat dari nilai *array*-nya (Gambar 4.24). Pada pengujian menggunakan metode

correlation, tingkat kecocokan paling tinggi ditunjukkan dengan nilai 1.0, untuk tingkat kecocokan sedang atau setengah ditentukan oleh nilai 0.7, sedangkan tingkat kecocokan terendah atau sama sekali tidak cocok ditunjukkan oleh nilai -1.0. Sehingga dari percobaan kali ini dapat dilihat bahwa data uji BWD4 bukanlah benar-benar tidak cocok, hanya saja syarat-syarat sesuai metode *correlation* pada data uji BWD4 kurang terpenuhi untuk dapat diklasifikasikan sebagai data model daun4, sehingga terjadi perbedaan klasifikasi.