

## BAB IV

### IMPLEMENTASI SISTEM

#### 4.1 *Dataset* Daun Terong

Setelah mendapatkan *dataset* citra daun terong dengan jumlah 1000 gambar daun sehat dan sakit, kemudian di kelompokkan menjadi 2 kelas dengan membuat 2 folder yaitu Daun Terong Sehat dan Daun Terong Sakit. Berikut tabel jumlah data dari setiap kelas.

Tabel 4.1 Jumlah *dataset* masing-masing kelas Daun Terong

Kelas	Jumlah
Jumlah Sakit	500
Jumlah Sehat	500

Berikut contoh *dataset* citra daun terong sehat dan sakit berdasarkan kelasnya masing-masing dapat dilihat pada Gambar 4.1 dan 4.2



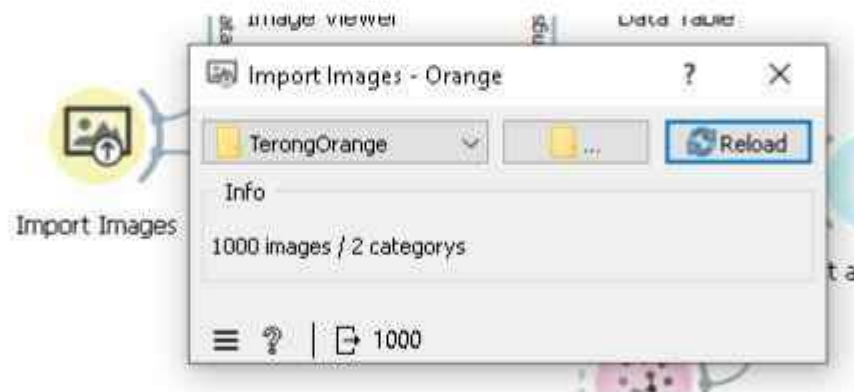
Gambar 4. 1 Citra Daun Terong Sehat



Gambar 4. 2 Citra Daun Terong Sakit

#### 4.2 Proses *Import Images*

Setelah Mendapatkan *dataset* dan mengelompokannya ke dalam kelas masing-masing. Tahap selanjutnya mengimport *dataset* daun terong yang sehat dan sakit ke dalam aplikasi *Orange* dengan menggunakan *widget import images*. Tampilan *widget import images* pada aplikasi *Orange* dapat dilihat pada Gambar 4.3



Gambar 4. 3 Tampilan *Widget Import Images*

Pada gambar diatas dapat dilihat bahwa *dataset* daun terong sehat dan sakit yang berjumlah 1000 gambar dengan 2 kelas berhasil di *import*. Untuk menampilkan dan memastikan bahwa *dataset* tersebut telah berhasil

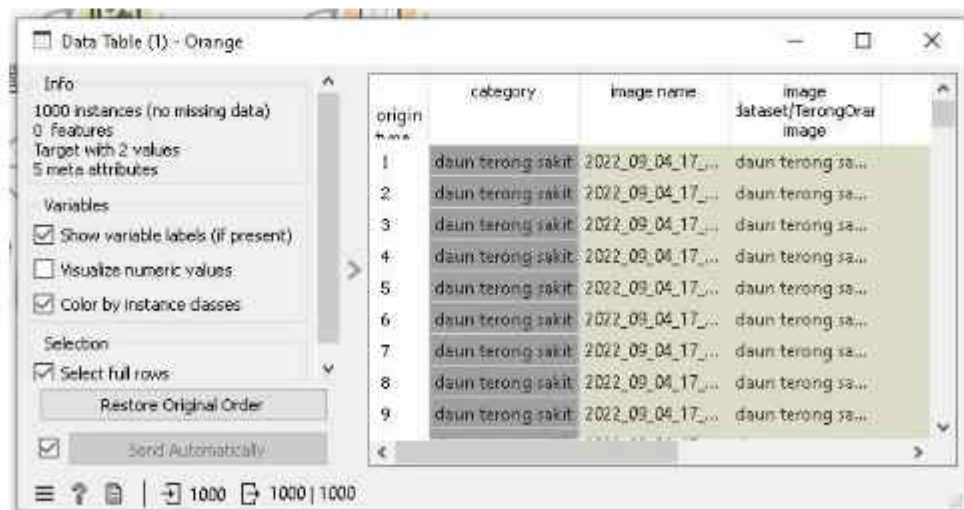
dimasukan ke dalam aplikasi *Orange* maka dibutuhkan *widget image viewer*.

Tampilan *widget image viewer* dapat dilihat pada Gambar 4.4



Gambar 4. 4 Tampilan *Widget Image Viewer*

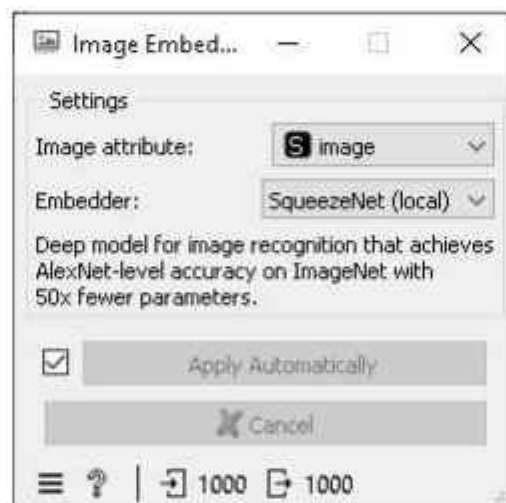
Setelah menampilkan *dataset* gambar, dapat pula ditampilkan data atribut seperti ukuran, lebar, dan tinggi gambar dengan menggunakan *widget data table*. Tampilan dari *widget data table* dapat dilihat pada Gambar 4.5



Gambar 4. 5 Kinerja Data Tabel Pada *Image Viewer*

### 4.3 Proses Image Embedding

Setelah menginput gambar menggunakan *widget import images*, tahap selanjutnya adalah melakukan *embedder* dengan *SqueezeNet* menggunakan *Image Embedding*. Tampilan *widget image embedding* dapat dilihat pada Gambar 4.6



Gambar 4. 6 Tampilan *Widget Image Embedding*

Setelah menggunakan *SqueezeNet* sebagai *embedder*, tahap selanjutnya menggunakan *widget data tabel* untuk menampilkan hasil kinerja *embedder*. Tampilan hasil dari kinerja *embedder* dapat dilihat pada Gambar 4.7

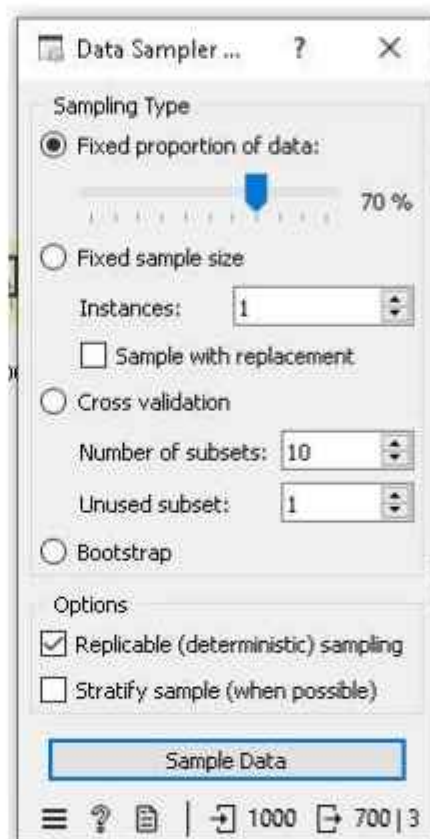
hidden origin	category	image name	image	size
1	daun terong saiki	1022_05_04_17_...	data/terong sa...	3470613
2	daun terong saiki	1022_05_04_17_...	data/terong sa...	3425095
3	daun terong saiki	1022_05_04_17_...	data/terong sa...	3425074
4	daun terong saiki	1022_05_04_17_...	data/terong sa...	3041510
5	daun terong saiki	1022_05_04_17_...	data/terong sa...	2442260
6	daun terong saiki	1022_05_04_17_...	data/terong sa...	3551405
7	daun terong saiki	1022_05_04_17_...	data/terong sa...	3013973
8	daun terong saiki	1022_05_04_17_...	data/terong sa...	2602194
9	daun terong saiki	1022_05_04_17_...	data/terong sa...	2711592
10	daun terong saiki	1022_05_04_17_...	data/terong sa...	2541004
11	daun terong saiki	1022_05_04_17_...	data/terong sa...	3092000

Gambar 4. 7 Hasil Kinerja *Embedder*

Pada gambar diatas, proses *image embedding* berhasil membaca *dataset* citra daun terong sehat dan daun terong sakit, dan berhasil mengeluarkan serta meningkatkan data tabel dengan tambahan kolom sebanyak 1000 kolom.

#### 4.4 Data Sampler

*Widget data sampler* mengimplementasikan beberapa metode pengambilan sampel data. *Outputnya* sampel dan data pelengkap (dengan contoh dari *set input* yang tidak termasuk dalam data *set* sampel). *Output* diproses setelah data *set input* disediakan dan sampel ditekan.



Gambar 4. 8 Tampilan *Widget Data Sampler*

#### 4.5 Pembangunan Klasifikasi *K-Nearest Neighbors*

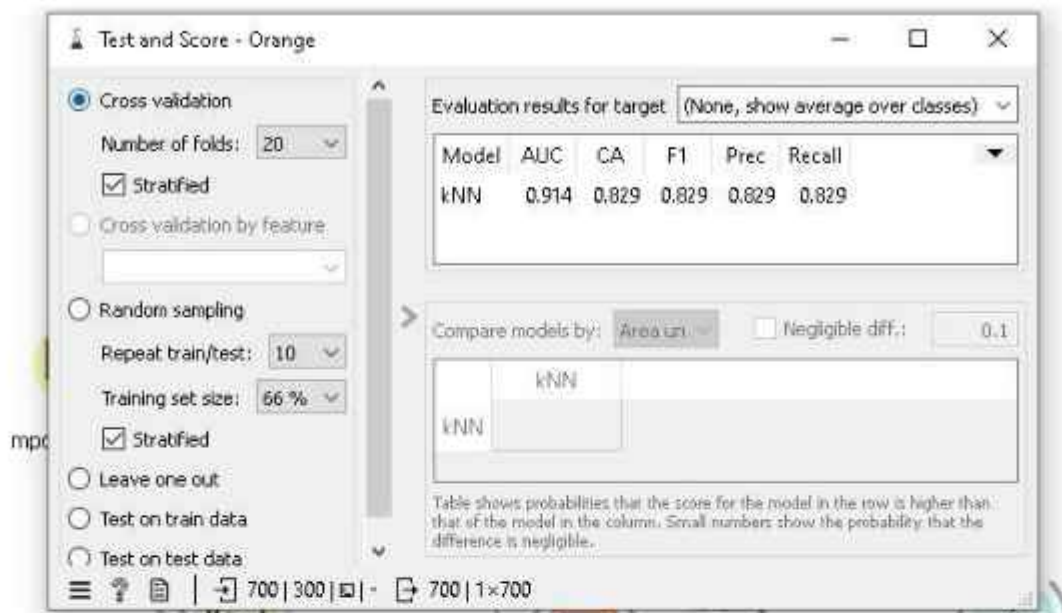
Pada tahap pembangunan klasifikasi, percobaan klasifikasi ini menggunakan metode *K-Nearest Neighbors* dengan menggunakan *number of neighbors = 3*, *metric = Euclidean* dan *weight = Uniform*. Tampilan dari *widget K-Nearest Neighbors* dapat dilihat pada Gambar 4.8



Gambar 4. 9 Tampilan *Widget K-Nearest Neighbor*

#### 4.6 *Training dan Testing*

Pada proses *training* dan *testing* menggunakan *widget test and score* untuk membangun model klasifikasi yang telah dibuat sebelumnya dengan mencoba 2, 3, 5 10, 20-*fold cross validation* yang ada pada *widget test and score*. Tampilan *widget test and score* dapat dilihat pada Gambar 4.9



Gambar 4. 10 Tampilan *Widget Test and Score*