

Klasifikasi Tanaman Anggrek Menggunakan Metode CNN Berbasis Web Django

Mochammad Fedro Firdaus¹, Yodhi Pratama Iswoyo², Yansyah Nurullah Ahmadi³

^{1,2,3}Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Nusantara PGRI Kediri

E-mail: ¹[*1rizkybangi@gmail.com](mailto:rizkybangi@gmail.com), ²yodhipratama900@gmail.com, ³yansyah.nurullah@gmail.com

Abstrak – Anggrek merupakan salah satu tanaman hias yang banyak dibudidayakan. Tiap genus anggrek mempunyai cara budidaya yang berbeda, sehingga para pembudidaya anggrek yang baru memulai perlu mengetahui genus dari anggrek yang akan dibudidayakannya terlebih dahulu. Namun tidak sedikit pemula yang mencoba membudidayakan anggrek tanpa ada pengetahuan dan pengalaman yang cukup, sehingga anggrek yang dibudidayakan tidak tumbuh dan berbunga dengan optimal. Citra yang dapat diklasifikasikan oleh sistem hanya citra 5 genus tanaman anggrek yang umum dibudidayakan, yaitu *Cattleya*, *Dendrobium*, *Oncidium*, *Phalaenopsis* dan *Vanda*. Klasifikasi citra genus tanaman anggrek menggunakan teknologi Deep Learning - Convolutional Neural Network (CNN) dapat membantu pembudidaya anggrek pemula dalam mengidentifikasi genus tanaman anggrek. Dalam penelitian ini akan dibuat sebuah model untuk klasifikasinya. Model tersebut akan diimplementasikan menggunakan framework django untuk mengupload gambar. Dataset yang digunakan pada penelitian ini didapatkan dari website github.com. Semua proses klasifikasi ini dilakukan melalui skema training dan testing, dimana tahap training menghasilkan sebuah model CNN beserta bobot (weight) yang telah diperbarui (updated), lalu tahap testing menggunakan model tersebut untuk diujikan terhadap data citra yang baru. Hasilnya bisa mengklasifikasi tanaman anggrek menggunakan model training yg sudah dilatih berupa `anggrek_model.H5` yang sudah diimplementasikan di framework django dan mendapatkan hasil akurasi sebesar 99%.

Kata Kunci — anggrek, klasifikasi citra, convolutional neural network.

1. PENDAHULUAN

Anggrek (*Orchidaceae*) merupakan salah satu tanaman hias yang sangat populer dan memiliki pesona yang indah [1]. Nilai penting dari tanaman anggrek terletak pada keindahan bunganya, dan keindahan bunga anggrek akan didapat jika dalam proses pembudidayaannya dilakukan secara tepat semenjak anggrek tersebut masih usia muda. Selain itu, masih sedikit sekali pakar anggrek yang dapat ditemui sehari-hari, dan waktu yang terbatas antara pakar dengan pembudidaya anggrek, serta literatur yang tidak mudah dijangkau. dalam penelitian ini dibangun sebuah sistem yang dapat mengklasifikasi citra genus taman anggrek secara otomatis. Genus tanaman anggrek yang dipilih yaitu yang paling umum dibudidayakan, terlebih sebagai tanaman hias, yaitu genus *Cattleya*, *Dendrobium*, *Oncidium*, *Phalaenopsis* dan *Vanda* [2]. Dengan terus berkembangnya teknologi pengolahan citra digital (digital image processing) dapat membantu menyelesaikan berbagai permasalahan sehari-hari, salah satunya klasifikasi citra. Dengan memanfaatkan metode *Convolutional Neural Network* (CNN) sebagai salah satu teknologi Deep Learning, masalah klasifikasi citra tanaman anggrek akan lebih mudah untuk dilakukan. CNN dipilih karena metode tersebut paling optimal dalam kasus klasifikasi citra, dimana salah satu kelebihanannya ialah ekstraksi fitur citra yang dilakukan secara otomatis, sehingga dapat menghemat waktu dan tenaga.

2. METODE PENELITIAN

Metode yang diterapkan untuk dalam penelitian ini untuk melakukan pengklasifikasikan adalah CNN (*Convolutional Neural Network*). Jaringan saraf konvolusional (CNN) ini menerima citra gambar sebagai input, dan melalui serangkaian lapisan konvolusi, citra tersebut diolah dengan menggunakan filter yang ditentukan. Setiap lapisan menghasilkan pola dari berbagai bagian citra, yang mempermudah proses klasifikasi.

2.1 Prosedur Penelitian

1. Studi Literatur

Pada tahap ini, dilakukan pencarian referensi yang relevan untuk mendukung penelitian yang akan dilaksanakan. Refrensi ini diperoleh melalui penelusuran jurnal atau artikel yang berkaitan

dengan citra. Sumber referensi tersebut bisa digunakan untuk menyusun pengklasifikasikan tanaman anggrek.

2. Pengumpulan Data

Pada tahap pengumpulan data ini, Dataset diambil dari sumber yang bersumber dari website Github.com, sesuai dengan rancangan studi literatur yang difokuskan pada klasifikasi tanaman anggrek menggunakan metode *Convolutional Neural Network*.

3. Analisa Sistem

Dalam analisa system, membahas mengenai metode apa yang akan digunakan penelitian. Dalam konteks ini, penelitian ini memanfaatkan metode *Convolutional Neural Network (CNN)* untuk melakukan klasifikasi tanaman anggrek [3].

4. Perancangan Sistem

Perancangan sistem yang diperoleh dari studi literatur, yang kemudian hasil tersebut diolah sesuai dengan penelitian yang akan dilakukan.

5. Desain Sistem

Desain sistem adalah rencana atau konsepsi yang akan diimplementasikan dalam bahasa pemrograman, dan perlu disesuaikan dengan karakteristik sistem yang akan dibangun.

6. Implementasi

Pada tahap implementasi ini membahas tentang perancangan sistem dan desain sistem yang telah disusun. Selanjutnya, implementasi akan dilakukan dalam bahasa pemrograman Python yang telah dipilih.

7. Uji Coba

Pada tahap pengujian, dilakukan verifikasi untuk memastikan bahwa implementasi sesuai dengan keinginan dan tujuan yang telah ditetapkan. Tujuan dari tahap ini adalah untuk mengidentifikasi dan meminimalisir kesalahan yang mungkin terjadi selama proses pengujian.

8. Debugging

Pada proses ini, dilakukan perbaikan terhadap error yang muncul selama tahap uji coba pada program.

9. Laporan

Penyusunan laporan dilakukan setelah semua proses telah selesai. Laporan ini disusun dengan memanfaatkan data yang telah dikumpulkan, menjelaskan rancangan dan pembuatan sistem, serta hasil pengujian yang telah dilakukan terhadap program.

2.2 Framework Django

Django adalah salah satu web framework berbasis Bahasa pemrograman Python. Django adalah web framework Python yang dipakai untuk membuat aplikasi web yang dinamis. Django dikembangkan oleh Django Software Foundation dan terus mendapatkan perbaikan sehingga membuat web framework yang satu menjadi pilihan utama bagi banyak pengembang aplikasi web [4].

2.3 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) merupakan salah satu metode machine learning dari pengembangan *Multi Layer Perceptron (MLP)* yang dirancang untuk mengolah data dua dimensi. CNN juga merupakan salah satu jenis metode dari Deep Neural Network yang didalamnya memiliki tingkat jaringan dan banyak digunakan di dalam citra. Metode CNN terdiri dari dua metode yaitu klasifikasi dengan menggunakan feed forward dan tahap pembelajaran dengan menggunakan backpropagation [5].

2.4 Analisa Dataset

Data yang digunakan pada penelitian ini adalah data gambar yang didapat dari Github.com. Data yang digunakan berupa gambar tanaman anggrek hias yaitu Cattleya, Dendrobium, Oncidium, Phalaenopsis dan Vanda dengan format JPG. Jumlah data yang digunakan sebanyak 2956 data. Data dibagi menjadi 2 yaitu data training dan testing. Data training terdapat 2270 data dan data testing terdapat 686 data dengan rincian 500 data setiap jenis tanaman.

```
total training cattleya images: 460
total training dendrobium images: 450
total training oncidium images: 460
total training phalaenopsis images: 450
total training vanda images: 450
total validation cattleya images: 115
total validation dendrobium images: 131
total validation oncidium images: 160
total validation phalaenopsis images: 160
total validation vanda images: 140
```

Gambar 1. Jumlah Data Training Dan Testing

2.5 Preprocessing Data

Proses pelatihan dimulai dengan menyesuaikan ukuran citra training dan validasi menjadi 32x32 piksel, diikuti dengan pembagian data menjadi fitur dan label. Label diubah menjadi representasi biner menggunakan one-hot encoding, dan fitur dinormalisasi dengan membagi setiap piksel dengan 255.0. Model *Convolutional Neural Network* (CNN) menggunakan Sequential API Keras dengan lapisan konvolusi dan dropout untuk mencegah overfitting, serta fungsi aktivasi *ReLU*. Fungsi `summary()` memberikan informasi parameter dan dimensi setiap lapisan. Data yang dipersiapkan siap untuk pelatihan menggunakan algoritma optimasi Adam, menggabungkan keuntungan algoritma gradient stokastik (SGD) dan metode adaptif.

2.6 Proses CNN

Setelah mendapatkan data tanaman anggrek, terdapat serangkaian proses pemrosesan data. Yang pertama mengekstrak dataset yang berupa ziped, kemudian dataset tersebut ditampilkan atau mereview untuk isi-isi dan jumlah dataset tersebut. Selanjutnya mengimport *library* dari *tensorflow*, yaitu library layer dan model. tahap selanjutnya melakukan setting arsitektur CNN (Konvolusi), di arsitektur cnn terdapat layer input konvolusi. Setiap layer konvolusi mengekstrak fitur menggunakan filter/kernel berukuran berbeda (3x3 dalam hal ini). Fungsi aktivasi ReLU diterapkan setelah setiap operasi konvolusi. Max pooling kemudian dilakukan dengan jendela 2x2 untuk mengurangi dimensi spasial peta fitur. Selanjutnya ke setting arsitektur CNN (Fully Connected) yang Tujuannya adalah membuat model neural network untuk mengenali dan mengklasifikasikan gambar bunga anggrek berdasarkan pola dan fitur visual yang terdapat dalam gambar tersebut dengan activationnya sigmoid dan relu. Tahap selanjutnya dilakukan Compile CNN dengan parameters fungsi loss categorical cross entropy dengan learning rate 0.001. Setelah dilakukan Compile CNN selanjutnya dilakukan pengolahan data gambar untuk pelatihan dan validasi model menggunakan *ImageDataGenerator* dari *TensorFlow*. Ini mencakup rescaling (merubah nilai piksel) dan pengaturan parameter untuk batch size, ukuran target gambar, serta mode kelas (binary). Proses ini mempersiapkan data untuk pelatihan model dalam bentuk batch dengan augmentasi gambar. Di tahap dilakukan pelatihan model menggunakan generator data. Proses ini melibatkan melalui beberapa *epoch* (iterasi melalui seluruh dataset) dan pada setiap langkahnya, model diperbarui berdasarkan batch gambar yang diberikan oleh generator data pelatihan. Data validasi juga digunakan untuk mengevaluasi kinerja model selama pelatihan lalu ditampilkan grafik selama training modelnya. Setelah tahapan tersebut selesai dilakukan konversi ke Keras_Model.h5 dan labels, untuk digunakan di website django.

2.7 Iterasi Epoch

Data dilatih dengan fungsi tensorflow dan keras pada data training dan data validasi, dengan jumlah epochs = 15. Dengan menggunakan 2270 data training dan 686 data validasi, dan proses iterasi berjalan sebanyak 15 iterasi, maka iterasi akurasi pada data validasi mengalami kenaikan. Saat proses iterasi berakhir dan mendapatkan akurasi sebesar 100% untuk data training dan 85,20% untuk data validasi.

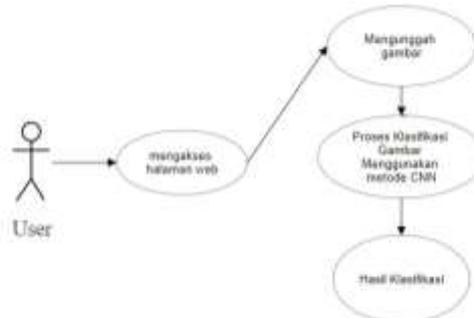
```
Epoch 1/15  
<ipython-input-55-91d346d1b720>:1: UserWarning: 'Model.fit_generator' is deprecated and will be removed in a  
  history = model.fit_generator(  
100/100 - 3s - loss: 1.3186 - acc: 0.8151 - val_loss: 1.2481 - val_acc: 0.8288 - 3s/epoch - 52ms/step  
Epoch 2/15  
100/100 - 4s - loss: 0.8948 - acc: 0.9735 - val_loss: 1.1858 - val_acc: 0.8468 - 4s/epoch - 37ms/step  
Epoch 3/15  
100/100 - 3s - loss: 0.8483 - acc: 0.9864 - val_loss: 1.8689 - val_acc: 0.8308 - 3s/epoch - 31ms/step  
Epoch 4/15  
100/100 - 4s - loss: 0.1317 - acc: 0.9804 - val_loss: 1.4452 - val_acc: 0.8368 - 4s/epoch - 45ms/step  
Epoch 5/15  
100/100 - 3s - loss: 0.0256 - acc: 0.9935 - val_loss: 2.1836 - val_acc: 0.8588 - 3s/epoch - 31ms/step  
Epoch 6/15  
100/100 - 4s - loss: 0.0147 - acc: 0.9980 - val_loss: 3.7279 - val_acc: 0.8308 - 4s/epoch - 56ms/step  
Epoch 7/15  
100/100 - 4s - loss: 0.0843 - acc: 0.9809 - val_loss: 3.2134 - val_acc: 0.8240 - 4s/epoch - 43ms/step  
Epoch 8/15  
100/100 - 4s - loss: 0.8384 - acc: 0.9960 - val_loss: 2.0627 - val_acc: 0.8600 - 4s/epoch - 36ms/step  
Epoch 9/15  
100/100 - 4s - loss: 1.4635e-05 - acc: 1.0000 - val_loss: 2.0709 - val_acc: 0.8500 - 4s/epoch - 36ms/step  
Epoch 10/15  
100/100 - 4s - loss: 1.3381e-06 - acc: 1.0000 - val_loss: 2.8005 - val_acc: 0.8520 - 4s/epoch - 41ms/step  
Epoch 11/15  
100/100 - 3s - loss: 5.5242e-07 - acc: 1.0000 - val_loss: 2.8079 - val_acc: 0.8500 - 3s/epoch - 31ms/step  
Epoch 12/15  
100/100 - 4s - loss: 3.7882e-07 - acc: 1.0000 - val_loss: 2.7786 - val_acc: 0.8460 - 4s/epoch - 38ms/step  
Epoch 13/15  
100/100 - 4s - loss: 3.7990e-07 - acc: 1.0000 - val_loss: 2.8167 - val_acc: 0.8560 - 4s/epoch - 40ms/step  
Epoch 14/15  
100/100 - 4s - loss: 3.1192e-07 - acc: 1.0000 - val_loss: 2.8978 - val_acc: 0.8520 - 4s/epoch - 36ms/step  
Epoch 15/15  
100/100 - 4s - loss: 2.6633e-07 - acc: 1.0000 - val_loss: 2.8944 - val_acc: 0.8520 - 4s/epoch - 39ms/step
```

Gambar 2. Proses Iterasi

```
Epoch 9/15  
100/100 - 4s - loss: 1.4635e-05 - acc: 1.0000 - val_loss: 2.0709 - val_acc: 0.8500 - 4s/epoch - 36ms/step  
Epoch 10/15  
100/100 - 4s - loss: 1.3381e-06 - acc: 1.0000 - val_loss: 2.8009 - val_acc: 0.8520 - 4s/epoch - 41ms/step  
Epoch 11/15  
100/100 - 3s - loss: 5.5242e-07 - acc: 1.0000 - val_loss: 2.8079 - val_acc: 0.8500 - 3s/epoch - 31ms/step  
Epoch 12/15  
100/100 - 4s - loss: 3.7882e-07 - acc: 1.0000 - val_loss: 2.7786 - val_acc: 0.8460 - 4s/epoch - 38ms/step  
Epoch 13/15  
100/100 - 4s - loss: 3.7990e-07 - acc: 1.0000 - val_loss: 2.8167 - val_acc: 0.8560 - 4s/epoch - 40ms/step  
Epoch 14/15  
100/100 - 4s - loss: 3.1192e-07 - acc: 1.0000 - val_loss: 2.8978 - val_acc: 0.8520 - 4s/epoch - 36ms/step  
Epoch 15/15  
100/100 - 4s - loss: 2.6633e-07 - acc: 1.0000 - val_loss: 2.8944 - val_acc: 0.8520 - 4s/epoch - 39ms/step
```

Gambar 3. Hasil Iterasi ke-15

2.8 Perancangan Sistem

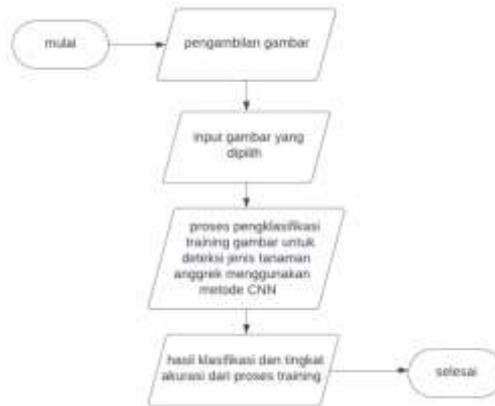


Gambar 4. Use Case Diagram

Pada Use Case Diagram diatas menjelaskan alur system klasifikasi tanaman anggrek menggunakan web django [6].

Alur dari sistem ini adalah :

1. Langkah pertama dari sistem ini adalah user mengakses web *local django*.
2. Menginput foto tanaman anggrek yang ingin di klasifikasi
3. Proses klasifikasi gambar menggunakan CNN tersebut memproses untuk mengenali kelas anggrek tersebut.
4. Jika system yang dibuat berhasil digunakan untuk mengklasifikasi tanaman anggrek maka proses tersebut selesai.

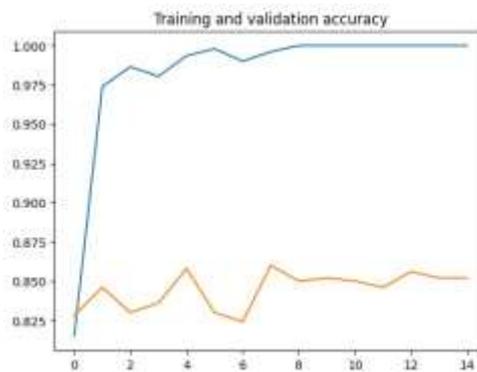


Gambar 5. Diagram Alur Proses

Pada Gambar 5 menjelaskan, *user* memasukkan gambar anggrek ke local web django. Setelah itu sistem memproses foto tanaman anggrek menggunakan metode *CNN (Convolutional Neural Network)*, yang terakhir hasil proses klasifikasi tersebut [7].

3. HASIL DAN PEMBAHASAN

Pada Bab ini merupakan pemaparan dari hasil implementasi dari perancangan program pada bab sebelumnya. Sistem yang dibuat sesuai dengan analisa dan perancangan yang nantinya mampu melakukan klasifikasi tanaman anggrek berdasarkan foto yang di input. Pada saat melakukan proses training data validasi mengalami penurunan akurasi sedangkan data training mengalami kenaikan, hal ini bisa dikatakan sebagai *overfitting*. *Overfitting* sendiri bisa dikatakan sebagai kemampuan model untuk membentuk ke data baru. Dalam penelitian ini, model yang dibuat mengalami *overfitting*, dengan kata lain, model yang telah dibuat mampu memahami dengan baik pola-pola dalam data pelatihan, tetapi ketika diterapkan pada data validasi, kemampuannya untuk beradaptasi secara luas mengalami penurunan kinerja. Ada kemungkinan bahwa model yang telah dibuat sudah tepat, namun kompleksitas data pelatihan yang lebih tinggi daripada data validasi dapat menyebabkan model lebih condong untuk mengingat detail-detail kecil dalam data pelatihan yang sebenarnya tidak relevan untuk data baru. Hasil akurasi training dan validasi



Gambar 6. Hasil Akurasi *Training* dan *Validation*

Pada gambar 6 merupakan hasil akurasi training dan validasi yang mana garis yang berwarna biru merupakan akurasi training dan garis warna orange merupakan akurasi validasi. Dari data tersebut akurasi bagian training terus meningkat sedangkan akurasi bagian validasi pada akhir iterasi menurun, sehingga bisa dikatakan sebagai *overfitting*.



Gambar 7. Hasil *Training Dan Validation Loss*

Pada Gambar 7 menunjukkan nilai training dan validation loss selama proses pelatihan dan evaluasi pada setiap epoch atau iterasi. Dari data tersebut loss data training mengalami penurunan dan loss data validasi mengalami peningkatan, dan menunjukkan bahwa hal tersebut mengalami overfitting.

3.1 Pembuatan Label Dan Model CNN

```
+ Kode + Teks
# Build the CNN model
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(img_width, img_height, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(5, activation='softmax')) # Assuming 5 classes

# compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# train the model
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // batch_size,
    epochs=15,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // batch_size)

# save the model
model.save('angrek_model.h5')

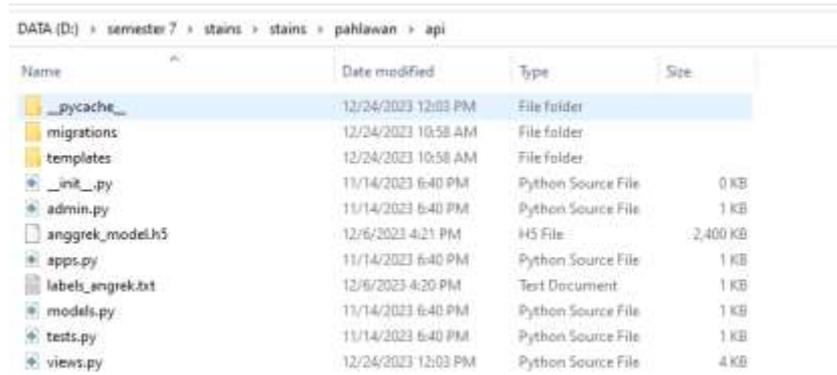
# save the labels
with open('labels_angrek.txt', 'w') as file:
    for label in train_generator.class_indices:
        file.write(label + '\n')
```

Gambar 8. *Source Code* Pembuatan Label dan Model

Pada gambar 8 merupakan pembuatan label dan model setelah data dilatih menggunakan fungsi tensorflow dan keras. Hasil data yang dilatih kan dijadikan file dengan nama model yaitu `angrek_model.h5` dan file label dengan nama `label_angrek.txt`. file tersebut digunakan untuk mengklasifikasikan gambar-gambar berdasarkan pola atau karakteristik yang telah dipelajari oleh model tersebut.

```
Found 2276 images belonging to 5 classes.
Found 686 images belonging to 5 classes.
Epoch 1/15
79/79 [*****] - 144s 2s/step - loss: 0.6379 - accuracy: 0.7735 - val_loss: 1.8286 - val_accuracy: 0.8921
Epoch 2/15
79/79 [*****] - 137s 2s/step - loss: 0.1025 - accuracy: 0.9723 - val_loss: 1.2683 - val_accuracy: 0.8512
Epoch 3/15
79/79 [*****] - 134s 2s/step - loss: 0.0670 - accuracy: 0.9821 - val_loss: 1.0753 - val_accuracy: 0.8542
Epoch 4/15
79/79 [*****] - 138s 2s/step - loss: 0.0470 - accuracy: 0.9857 - val_loss: 1.3817 - val_accuracy: 0.8497
Epoch 5/15
79/79 [*****] - 148s 2s/step - loss: 0.0508 - accuracy: 0.9848 - val_loss: 1.4681 - val_accuracy: 0.8423
Epoch 6/15
79/79 [*****] - 143s 2s/step - loss: 0.0156 - accuracy: 0.9946 - val_loss: 1.7451 - val_accuracy: 0.8348
```

Gambar 9. Proses Dalam Pembuatan Model Dan Label

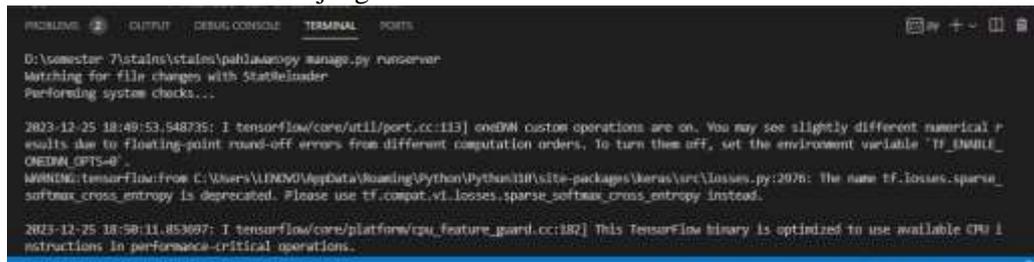


Gambar 10. Hasil dalam Pembuatan Model dan Label

Dalam gambar 10 merupakan hasil dari pembuatan model dan label berupa file model yang bernama `anggrek_model.h5` dan file label dengan nama `label_anggrek.txt`. file tersebut dijadikan satu dalam folder `api` dengan tujuan memudahkan dalam pemanggilan model dan label tersebut.

3.2 Tahapan Uji Coba klasifikasi Dengan *Framework* Web Local Django

1. Menjalankan *Framework* Local Django



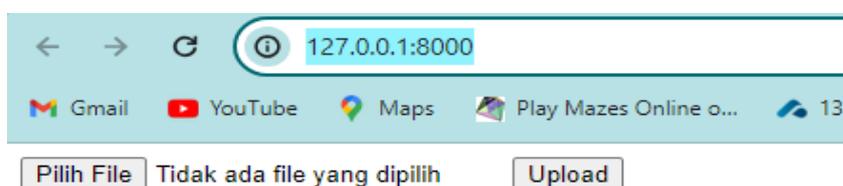
Gambar 11. Menjalankan *Framework* Local Django

Pada gambar 11 merupakan cara menjalankan *framework* Django yaitu dengan cara mengetik `python manage.py runserver`. Maka program akan menjalankan dan menampilkan *development server* yang akan dituju. dan cara masuk kedalam *development server* tersebut yaitu dengan cara mengeklik alamat yang ditunjukkan oleh *development server* tersebut.



Gambar 12. Alamat *development server* *Framework* Local Django

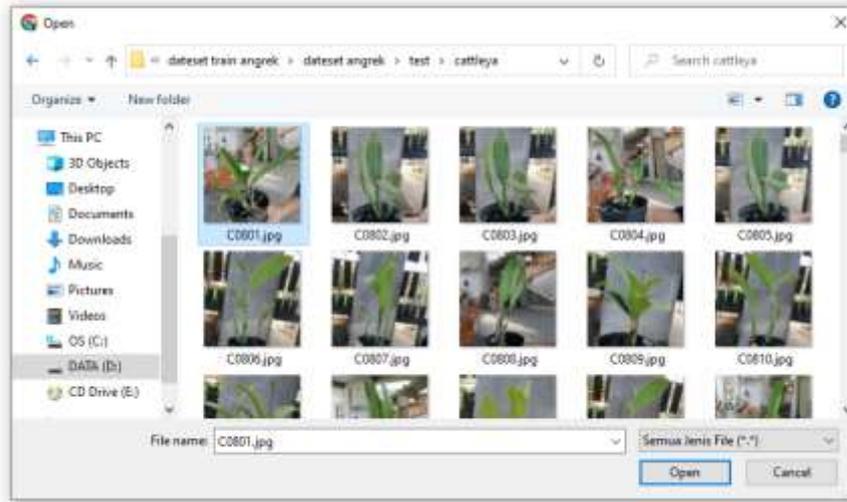
2. Halaman Upload Gambar



Gambar 13 Halaman Upload Gambar

Pada Gambar 13 merupakan tampilan awal framework Django untuk menginputkan sebuah gambar yang akan diklasifikasi yaitu dengan cara menekan tombol pilih file, maka akan diarahkan ke dalam tempat penyimpanan gambar yang akan diklasifikasi.

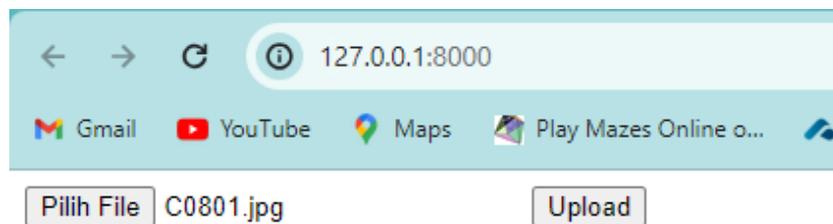
3. Pemilihan Gambar Yang Akan Di Upload



Gambar 14. Pemilihan Gambar Yang Akan Di Upload

Pada gambar 14 merupakan pemilihan gambar yang akan di upload, setelah menekan gambar yang dipilih kemudian tekan tombol open untuk memasukkan gambar tersebut.

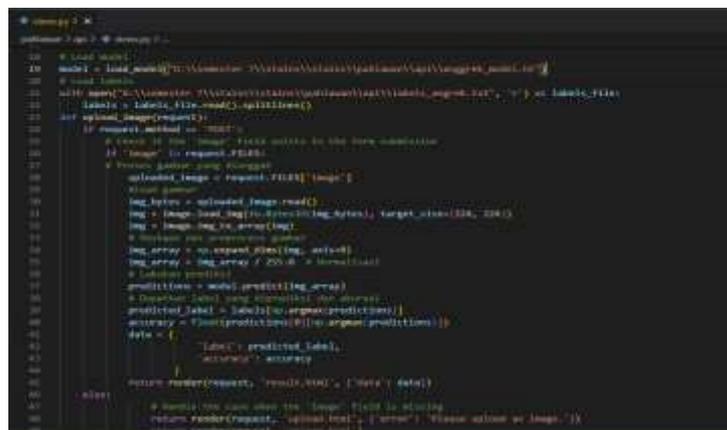
4. Tampilan file gambar yang telah di upload



Gambar 15. Tampilan file gambar yang telah di upload

Pada gambar 15 merupakan tampilan halaman yang telah memilih gambar yang ingin di klasifikasi kemudian tekan tombol upload untuk mengklasifikasi gambar tersebut, dengan mengambil contoh memilih foto dari folder tanaman anggrek cattleya.

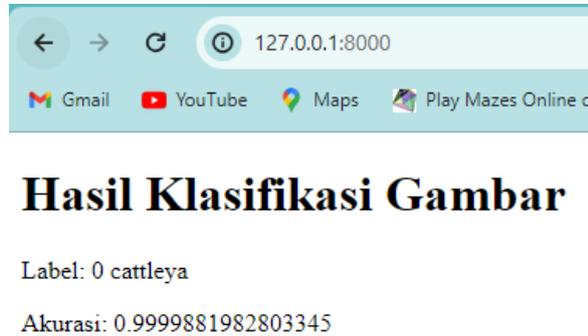
5. Proses Klasifikasi Gambar



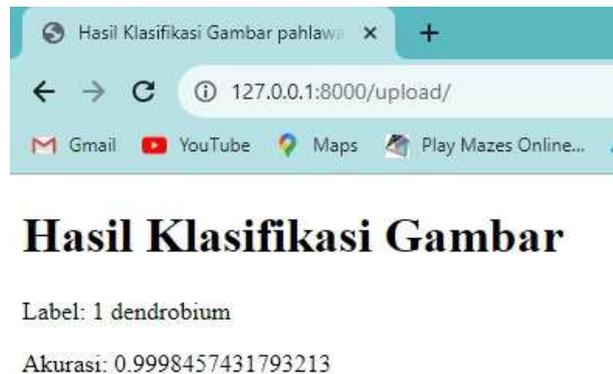
Gambar 16. Proses Klasifikasi Gambar

Pada gambar 16 merupakan proses dalam mengklasifikasi gambar yaitu dengan melakukan pengambilan model dan label terlebih dahulu. Dan model tersebut digunakan untuk pembandingan pada gambar yang di upload kemudian melakukan prediksi dengan cara membandingkan gambar yang di upload dengan model gambar yang telah di buat, maka akan mendapatkan hasil klasifikasi gambar tersebut.

6. Hasil Klasifikasi Gambar Yang Dituju



Gambar 17. Hasil Klasifikasi Gambar Tanaman Anggrek Cattleya



Gambar 18. Hasil Klasifikasi Gambar Tanaman Anggrek Dendrobium

Gambar diatas merupakan hasil dari memasukkan gambar tanaman anggrek yang diupload yaitu muncul hasil klasifikasi gambar yang isinya label dan akurasi pada gambar tersebut.

4. SIMPULAN

Dari penelitian yang dilakukan dapat ditarik kesimpulan yaitu berdasarkan proses implementasi yang dilakukan dengan model dan testing data yang dijalankan pada program berbasis web. Terhadap klasifikasi tanaman anggrek yang menggunakan metode CNN mampu mengklasifikasi jenis tanaman anggrek dengan tingkat akurasi sebesar 99% untuk klasifikasi jenis tanaman anggrek.

5. SARAN

Diharapkan pada penelitian selanjutnya dapat menambah jumlah dataset untuk mengklasifikasikan jenis tanaman anggrek yang lain atau dengan membandingkan metode lain dengan hasil yang baik sehingga dapat menghasilkan nilai akurasi yang lebih baik dan hasil yang signifikan.

DAFTAR PUSTAKA

- [1] K. Husodo, C. Lubis and Z. Rusd, "KLASIFIKASI TANAMAN ANGGREK MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DENGAN ARSITEKTUR VGG-16," *JURNAL SISTEM INFORMASI DAN TEKNIK*

- KOMPUTER*, vol. 8, no. 2, pp. 253-258, 2023.
- [2] M. R. R. Allaam and A. T. Wibowo, "Klasifikasi Genus Tanaman Anggrek Menggunakan Metode Convolutional Neural Network (CNN)," *e-Proceeding of Engineering*, vol. 8, no. 2, pp. 3150-3179, 2021.
- [3] D. D. T. Amelia, J. Sulaksono and D. W. Widodo, "Sistem Pendeteksi Kekerasan Berbasis Cnn (Convolutional Neural Network)," *PROSIDING SEMINAR NASIONAL TEKNOLOGI DAN SAINS*, vol. 2, no. 1, pp. 457-462, 2023.
- [4] D. Saputra and R. F. Aji, "ANALISIS PERBANDINGAN PERFORMA WEB SERVICE REST MENGGUNAKAN FRAMEWORK LARAVEL, DJANGO DAN RUBY ON RAILS UNTUK AKSES DATA DENGAN APLIKASI MOBILE," *Bangkit Indonesia*, vol. 2, no. VII, pp. 17-22, 2018.
- [5] J. Vicky, F. Ayu and B. Julianto, "Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN," *PROSIDING SEMINAR NASIONAL TEKNOLOGI DAN SAINS*, vol. 2, no. 1, pp. 155-162, 2023.
- [6] Purwanto and Sumardi, "Perancangan Klasifikasi Tanaman Herbal Menggunakan Transfer Learning pada Algoritma Convolutional Neural Network (CNN)," *JURNAL ILMIAH INFOKAM*, vol. XVIII, no. 2, pp. 105-118, 2022.
- [7] J. Vicky, F. Ayu and B. Julianto, "Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN," *PROSIDING SEMINAR NASIONAL TEKNOLOGI DAN SAINS*, vol. 2, no. 1, pp. 155-162, 2023.