

KLASIFIKASI CITRA BUNGA MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)

Marsha Alifia Vesthi, Dwi Rolliawati, Noor Wahyudi, Yunita Ardilla

Sistem Informasi, Universitas Islam Negeri Sunan Ampel Surabaya

Jl. Dr. Ir. Soekarno No.682 Surabaya, Indonesia

marshaavesthi@gmail.com

ABSTRAK

Klasifikasi citra bunga menjadi tantangan dalam pengolahan citra digital karena adanya variasi warna, bentuk, dan tekstur yang beragam pada tiap spesies. Penelitian ini bertujuan untuk mengidentifikasi lima jenis bunga, yaitu *daisy*, *sunflower*, *tulip*, *dandelion*, dan *rose* menggunakan *Convolutional Neural Network* (CNN). Dataset citra bunga yang digunakan berjumlah 4.317 citra dari Kaggle. Tahapan preprocessing mencakup resize citra, normalisasi piksel, serta augmentasi data untuk meningkatkan kemampuan generalisasi model. Arsitektur CNN yang diterapkan terdiri atas lapisan convolutional, pooling, dan dense untuk mengekstraksi fitur dan melakukan klasifikasi. Hasil pengujian menunjukkan bahwa model mencapai akurasi validasi sebesar 74,07%. Kesalahan klasifikasi terutama terjadi karena kemiripan visual antarspesies atau pencahayaan kurang baik. Penelitian ini menunjukkan bahwa CNN cukup efektif digunakan untuk klasifikasi citra bunga dan berpotensi dikembangkan lebih lanjut dalam aplikasi pertanian, konservasi tanaman, maupun edukasi berbasis teknologi.

Kata kunci : Klasifikasi Citra Bunga, Convolutional Neural Network (CNN), Deep Learning

1. PENDAHULUAN

Klasifikasi citra adalah salah satu cabang dalam bidang pengolahan citra digital yang bertujuan untuk mengidentifikasi dan mengategorikan objek dalam gambar ke dalam kategori tertentu. Dalam beberapa tahun terakhir, perkembangan teknologi kecerdasan buatan, khususnya *deep learning*, telah mendorong kemajuan pesat dalam klasifikasi citra [1]. Salah satu metode yang populer adalah *Convolutional Neural Networks* (CNN), sebuah arsitektur jaringan saraf tiruan yang dirancang untuk mengolah data visual [2]. Dengan kemampuannya dalam mengekstraksi fitur kompleks dari gambar, CNN telah berhasil diterapkan dalam berbagai aplikasi seperti pengenalan wajah, deteksi objek, dan klasifikasi citra [3].

Bunga adalah salah satu jenis tanaman dengan keragaman yang sangat luas. Menurut data dari Royal Botanic Gardens, Kew, terdapat lebih dari 369.000 spesies bunga yang telah diidentifikasi di seluruh dunia, dengan angka tersebut diperkirakan terus bertambah seiring ditemukannya spesies baru setiap tahunnya. Bunga mencakup sekitar 80% dari total keanekaragaman tumbuhan di bumi dan tersebar di berbagai habitat, mulai dari daerah tropis hingga kutub. Selain perannya dalam menjaga keseimbangan ekosistem, bunga juga memiliki nilai ekonomi besar di bidang hortikultura, kosmetik, dan farmasi [4].

Klasifikasi citra bunga ini menjadi tantangan tersendiri karena bunga memiliki karakteristik visual yang beragam, termasuk warna, bentuk kelopak, pola, dan ukuran. Kemiripan visual di antara jenis bunga tertentu, seperti *daisy* dan *sunflower*, sering kali menyulitkan proses klasifikasi. Selain itu, kondisi variabel dalam proses pengambilan gambar, seperti pencahayaan, sudut pengambilan gambar, dan latar belakang yang berbeda, turut memperumit akurasi klasifikasi citra [5]. Beberapa penelitian sebelumnya

menunjukkan bahwa CNN telah cukup efektif dalam klasifikasi bunga. Namun, permasalahan kemiripan visual antarspesies dan keterbatasan generalisasi masih sering menyebabkan kesalahan klasifikasi.

Penelitian ini menggunakan dataset *Flower Recognition Dataset* yang terdiri dari gambar berbagai jenis bunga dengan warna dan karakteristik beragam. Dataset ini memiliki kategori yang jelas dan terstruktur sehingga cocok digunakan untuk melatih model CNN dalam klasifikasi citra bunga [6]. Melalui pendekatan ini, diharapkan dapat dicapai model yang mampu melakukan klasifikasi bunga dengan akurasi tinggi, yang relevan untuk aplikasi dunia nyata, seperti sistem identifikasi tanaman berbasis gambar [7].

Tujuan dari penelitian ini adalah untuk menguji kemampuan CNN dalam mengklasifikasikan berbagai jenis bunga berdasarkan dataset yang telah disediakan. Kinerja model akan dievaluasi menggunakan metrik seperti akurasi, presisi, *recall*, *f1-score*, dan *confusion matrix* untuk memberikan gambaran menyeluruh tentang performa model [8]. Dengan pendekatan ini, penelitian ini diharapkan dapat memberikan wawasan yang bermanfaat secara praktis dalam pengembangan aplikasi identifikasi tanaman berbasis gambar untuk kepentingan hortikultura, edukasi, maupun konservasi.

2. TINJAUAN PUSTAKA

Convolutional Neural Network (CNN) adalah salah satu metode dalam *deep learning* yang banyak digunakan untuk pemrosesan gambar. CNN memiliki kemampuan untuk mengenali pola-pola kompleks dalam data citra dan telah terbukti efektif dalam berbagai aplikasi seperti pengenalan wajah, deteksi objek, serta klasifikasi citra [9]. Untuk memahami lebih dalam mengenai penggunaan CNN dalam klasifikasi citra bunga, berikut akan dijelaskan konsep dasar dan komponen utama dalam arsitektur CNN.

2.1. Penelitian Terdahulu

Beberapa studi terdahulu telah menggunakan metode *Convolutional Neural Network* (CNN) untuk klasifikasi citra bunga, karena kemampuannya dalam mengekstraksi fitur visual secara otomatis tanpa memerlukan proses ekstraksi manual.

Munandar dan Rozi [9] melakukan penelitian mengenai arsitektur CNN untuk klasifikasi citra bunga dengan menggunakan VGG16 dan NasNetMobile, baik dengan *fine-tuning* maupun tanpa *fine-tuning*. Hasilnya, model NasNetMobile dengan *fine-tuning* mencapai akurasi tertinggi sebesar 99,15%, sedangkan tanpa *fine-tuning* hanya sebesar 97,45%. Penelitian ini menunjukkan bahwa *fine-tuning* dapat meningkatkan akurasi dan kemampuan generalisasi model.

Aini dkk. [10] menerapkan model CNN dengan algoritma optimasi Nadam untuk klasifikasi bunga menggunakan dataset Oxford17 dan Oxford102. Hasil penelitian menunjukkan akurasi 60% dan 84% pada dataset Oxford17 serta 42% dan 64% pada dataset Oxford102, dengan hasil lebih baik pada pendekatan *transfer learning*. Perbedaan hasil ini menunjukkan bahwa kompleksitas dataset dan variasi morfologi bunga sangat memengaruhi performa model.

Sementara itu, Rahman dkk. [7] mengembangkan sebuah model klasifikasi bunga berbasis MobileNetV2 dengan pendekatan *transfer learning* menggunakan bobot ImageNet. Dataset berisi 2.137 citra dari 11 jenis bunga lokal Indonesia. Hasil menunjukkan bahwa peningkatan akurasi dari 42% menjadi 73% dengan model yang menggunakan *transfer learning*.

Soomro [11] mengusulkan sistem klasifikasi bunga berbasis *Artificial Neural Network* (ANN) yang memanfaatkan *Discrete Wavelet Transform* (DWT) dan *Gray Level Co-occurrence Matrix* (GLCM) untuk mengekstraksi fitur tekstur. Sistem ini diuji pada empat jenis bunga dan menghasilkan akurasi 82%. Hasil menunjukkan bahwa kombinasi ANN dan fitur tekstural untuk klasifikasi bunga dapat meningkatkan kinerja klasifikasi dengan efisiensi tinggi.

Selain itu, Hanafiah dkk. [12] meneliti model CNN berbasis AlexNet dan VGG16 menggunakan pendekatan *transfer learning* dengan dataset bunga dari Kaggle. Model VGG16 memberikan performa terbaik dengan akurasi 95,02%, sedangkan AlexNet hanya mencapai 85,69%. Penelitian ini menegaskan bahwa semakin dalam arsitektur CNN, semakin baik pula kemampuan model dalam mengenali fitur visual yang kompleks pada klasifikasi citra bunga.

Berdasarkan berbagai penelitian tersebut, dapat disimpulkan bahwa CNN merupakan metode yang efektif dalam klasifikasi citra bunga. Namun, tingkat akurasi bergantung pada arsitektur yang digunakan, teknik augmentasi data, serta strategi *fine-tuning* yang diterapkan. Oleh karena itu, penelitian ini berfokus pada penerapan arsitektur CNN dasar dengan optimasi parameter pelatihan dan augmentasi data untuk dapat menghasilkan model yang akurat dan efisien tanpa menggunakan pendekatan *transfer learning*.

2.2. Citra Bunga

Citra adalah representasi visual dari objek atau pemandangan yang dapat dihasilkan melalui berbagai cara, termasuk fotografi atau pemindaian. Dalam konteks pengolahan citra digital, citra merujuk pada gambar yang diproses dalam bentuk data numerik yang dianalisis, dimodifikasi, atau diklasifikasikan menggunakan teknik komputasi. Citra terdiri dari elemen dasar yang disebut piksel (*picture element*), yang mewakili warna atau intensitas cahaya tertentu.

Citra bunga adalah gambar yang menampilkan bunga dalam berbagai bentuk, ukuran, warna, dan jenis. Setiap bunga memiliki karakteristik visual yang khas, seperti bentuk kelopak, warna, tekstur, dan susunan bunga [13]. Dalam pengolahan citra digital, pengenalan dan klasifikasi citra bunga merupakan tantangan yang kompleks, karena bunga dapat muncul dengan berbagai latar belakang, pencahayaan, dan sudut pandang yang bervariasi [5]. Hal ini menuntut penggunaan teknik yang efektif dalam mengekstraksi dan mengenali pola-pola penting dalam citra bunga dengan memanfaatkan teknologi *deep learning*.

2.3. Deep Learning

Deep Learning adalah subbidang dari machine learning yang berfokus pada penggunaan jaringan saraf tiruan (*artificial neural networks*) yang memiliki banyak lapisan (*deep networks*) untuk memproses data dalam jumlah besar dan kompleks. Berbeda dengan machine learning tradisional yang mengharuskan ekstraksi fitur manual, *deep learning* memungkinkan model untuk secara otomatis belajar dari data melalui lapisan-lapisan yang semakin dalam dan kompleks. Teknologi ini sangat efektif dalam menangani data yang tidak terstruktur, seperti gambar, suara, dan teks.

Deep learning memungkinkan komputer untuk meniru cara manusia belajar, yaitu mengidentifikasi pola-pola tersembunyi yang ada di dalam data. Salah satu algoritma utama dalam *deep learning* adalah *Convolutional Neural Network* (CNN), yang telah terbukti sangat sukses dalam tugas pengolahan citra, termasuk dalam klasifikasi citra bunga [14].

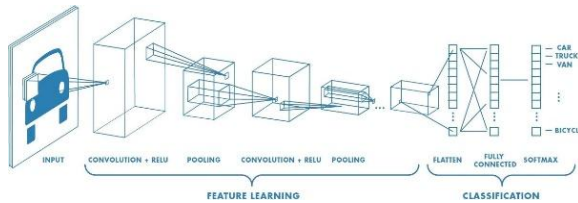
2.4. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah jenis jaringan syaraf tiruan yang dirancang untuk bekerja dengan data berbentuk gambar. CNN memanfaatkan *convolution* untuk mengekstraksi fitur dari citra secara otomatis. CNN terdiri dari beberapa lapisan (*layer*) yang bekerja untuk mengidentifikasi fitur visual yang ada dalam citra, seperti garis, tekstur, atau bentuk. Dalam CNN, setiap lapisan memiliki tugas untuk mengekstraksi dan mengolah informasi.

CNN memiliki kemampuan untuk mempelajari fitur dari data gambar dengan sedikit pengolahan manual, yang membedakannya dengan metode pengolahan citra tradisional. CNN dapat menangani masalah kompleksitas gambar dan variasi seperti rotasi, pergeseran, dan perubahan ukuran objek [15].

2.4. Arsitektur Jaringan CNN

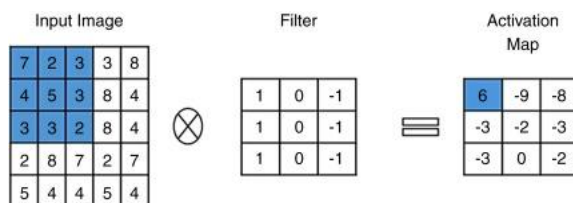
Arsitektur CNN mencakup sejumlah layer utama, seperti *convolutional layer*, *pooling layer*, dan *fully-connected layer*, serta teknik regulasi seperti *dropout*. Berikut penjelasan tiap layer yang ditunjukkan pada Gambar 1 untuk mempermudah pemahaman.



Gambar 1. Arsitektur CNN

2.5. Convolutional Layer

Convolutional Layer (Lapisan Konvolusi) adalah komponen pertama yang sangat penting dalam CNN. Fungsinya adalah untuk mengekstraksi fitur dari citra input. Pada lapisan ini, filter atau kernel digunakan untuk melakukan operasi konvolusi pada citra input. Proses konvolusi ini ditunjukkan pada Gambar 2.

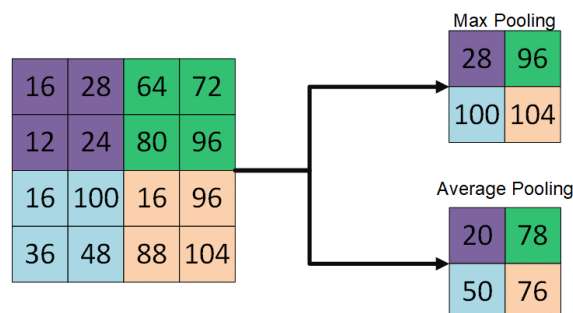


Gambar 2. Proses Konvolusi

Dari Gambar 2 diatas, dapat dilihat bahwa proses konvolusi melibatkan pergeseran filter dan melakukan operasi perkalian dan penjumlahan, sehingga CNN mampu mengekstraksi fitur citra secara otomatis.

2.6. Pooling Layer

Pooling layer bertujuan mengurangi dimensi *feature map* untuk menekan kompleksitas komputasi, tanpa menghilangkan informasi penting. Dua jenis pooling yang umum digunakan adalah *Max Pooling* (mengambil nilai maksimum) dan *Average Pooling* (mengambil nilai rata-rata). Ilustrasi perbedaan antara kedua metode ditunjukkan pada Gambar 3.

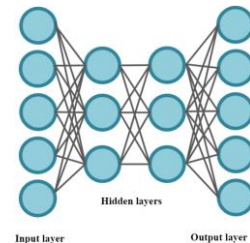


Gambar 3. Max Pooling dan Average Pooling

Dari Gambar 3, dapat dilihat bahwa pooling membantu membuat model lebih robust terhadap perubahan kecil pada citra, seperti rotasi atau translasi.

2.7. Fully-Connected Layer

Fully-connected layer menghubungkan semua neuron dari layer sebelumnya ke neuron dalam layer ini untuk menghasilkan output akhir dari proses klasifikasi yang dapat dilihat pada Gambar 4.

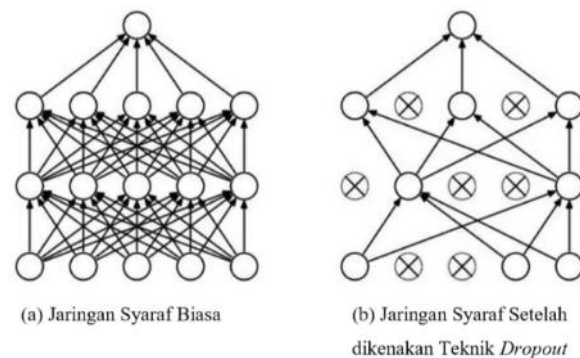


Gambar 4. Fully-Connected Layer

Dari Gambar 4, dapat dilihat bahwa semua neuron terhubung penuh, sehingga memungkinkan integrasi informasi dari fitur yang diekstraksi CNN.

2.8. Dropout

Dropout adalah teknik regulasi untuk mencegah *overfitting* dengan mematikan sejumlah neuron secara acak selama pelatihan yang ada pada Gambar 5.



Gambar 5. Proses Dropout

Dari Gambar 5, dapat dilihat bahwa beberapa neuron tidak aktif dalam iterasi tertentu, sehingga model lebih mampu melakukan generalisasi.

3. METODE PENELITIAN

Penelitian ini bertujuan untuk membangun dan mengevaluasi model klasifikasi gambar menggunakan *Convolutional Neural Network* (CNN). Tahapan yang dilakukan yaitu pengumpulan dataset, *preprocessing* data, perancangan arsitektur CNN, pelatihan model, pengujian model, analisis hasil, dan kesimpulan.

3.1. Pengumpulan Dataset

Dataset yang digunakan diperoleh dari platform Kaggle dengan judul "Flowers Recognition Dataset" [<https://www.kaggle.com/datasets/alxmamaev/flowers-recognition>]. Dataset ini terdiri dari lima kelas bunga, yaitu *daisy*, *dandelion*, *rose*, *sunflower*, dan

tulip. Setiap kelas telah dikelompokkan dalam folder sesuai label masing-masing. Variasi kualitas dan latar belakang gambar dalam dataset ini menjadikannya relevan untuk melatih model CNN agar lebih robust terhadap perbedaan kondisi citra.

3.2. Preprocessing Data

a. Resize (Mengubah Ukuran) Citra

Dataset bunga yang digunakan memiliki ukuran resolusi bervariasi untuk setiap gambar. Sebelum masuk ke tahap *feature extraction* pada arsitektur CNN, dilakukan proses *resize* untuk mengubah resolusi piksel menjadi seragam, yaitu 150x150 piksel. Proses ini bertujuan agar citra dapat diolah secara efisien oleh CNN serta membantu meningkatkan akurasi model dalam mengenali pola dan fitur visual.

b. Pembagian Dataset

Dataset dibagi menjadi dua bagian, yaitu data latih (80%) dan data uji (20%). Pembagian dilakukan secara acak sehingga setiap kelas bunga tetap terwakili secara proporsional. Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk mengevaluasi performa model secara objektif.

3.3. Arsitektur CNN

Arsitektur CNN yang dirancang terdiri dari *convolutional layer* untuk ekstraksi fitur dari gambar, *pooling layer* untuk reduksi dimensi, dan *dense layer* untuk klasifikasi. Optimasi *hyperparameter* dilakukan dengan mencoba beberapa kombinasi, seperti ukuran gambar, jenis *optimizer* (RMSProp dan Adam), serta jumlah neuron pada *dense layer*. Tujuannya adalah memperoleh konfigurasi model dengan performa terbaik dalam klasifikasi citra bunga.

3.4. Pelatihan Model

Model dilatih menggunakan data latih yang telah melalui tahap *preprocessing*. Fungsi *loss* yang digunakan adalah categorical crossentropy, sedangkan metrik utama evaluasi selama pelatihan adalah akurasi. Parameter pelatihan, seperti ukuran *batch*, jumlah *epoch*, dan *steps per epoch*, diatur untuk memastikan proses pelatihan berjalan optimal dan konvergen.

3.5. Pengujian Model

Pengujian dilakukan menggunakan data uji untuk mengukur performa model. Pengujian ini bertujuan untuk menentukan kombinasi *hyperparameter* terbaik yang menghasilkan performa paling optimal. Evaluasi ini dilakukan menggunakan beberapa metrik berikut:

a. Accuracy, untuk mengukur persentase prediksi yang benar terhadap seluruh data uji.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

b. Precision, untuk mengukur seberapa tepat model dalam memprediksi kelas positif, yaitu berapa banyak prediksi positif yang benar.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

c. Recall, untuk mengukur tingkat keberhasilan model dalam menemukan semua data positif, yaitu seberapa banyak data positif yang berhasil diprediksi dengan benar oleh model.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

d. F1-Score, sebagai rata-rata harmonis antara precision dan recall pada dataset.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

e. Confusion Matrix, untuk memberikan gambaran distribusi hasil prediksi model terhadap label sebenarnya pada setiap kelas bunga.

3.6. Analisis Hasil

Analisis dilakukan berdasarkan visualisasi grafik akurasi dan loss pada tahap pelatihan untuk memantau stabilitas dan konvergensi pada model. Kesalahan klasifikasi pada citra tertentu dianalisis lebih lanjut untuk mengetahui kelemahan model, khususnya pada bunga dengan karakteristik visual mirip.

3.7. Kesimpulan

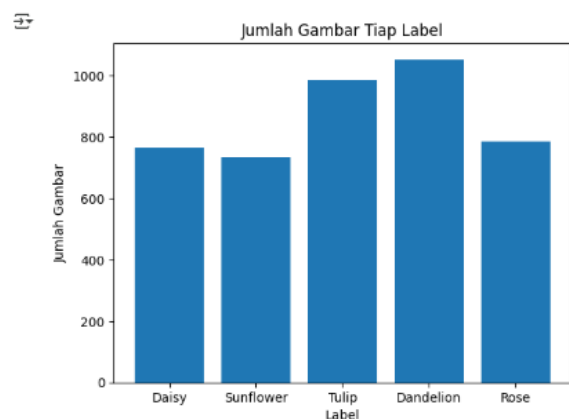
Kesimpulan diambil berdasarkan hasil evaluasi model, meliputi efektivitas CNN dalam klasifikasi citra serta rekomendasi pengembangan lebih lanjut.

4. HASIL DAN PEMBAHASAN

Pada bagian ini, akan dibahas mengenai hasil yang diperoleh dari model *Convolutional Neural Network* (CNN). Dengan melalui beberapa proses, mulai dari persiapan data, *preprocessing*, hingga pelatihan model, hasil yang diperoleh akan dianalisis untuk memberikan pemahaman lebih lanjut mengenai kekuatan dan kelemahan model yang telah dibangun.

4.1. Pengenalan Data dan Proses Import

Dataset ini terdiri dari 4.317 gambar yang mencakup lima kelas bunga, yaitu *daisy*, *dandelion*, *rose*, *sunflower*, dan *tulip*. Setiap kelas memiliki jumlah gambar yang bervariasi sehingga perlu dilakukan visualisasi distribusi awal menggunakan diagram batang untuk memastikan keseimbangan data.



Gambar 6. Diagram Batang Dataset

4.2. Preprocessing Data

Sebelum data digunakan dalam pelatihan model, dilakukan tahap *preprocessing* untuk menyiapkan citra agar lebih mudah diproses oleh CNN. Seluruh gambar diubah ukurannya menjadi 150×150 piksel sehingga dimensi menjadi konsisten. Nilai piksel yang awalnya berada di rentang $[0,255]$ diubah menjadi $[0,1]$.

Selanjutnya, dilakukanlah *categorical encoding* pada label kelas bunga menggunakan metode *One-Hot Encoding*, sehingga setiap kelas direpresentasikan dalam bentuk vektor biner berdimensi lima.

```
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical

# Konversi label menjadi angka menggunakan LabelEncoder
le = LabelEncoder()
Y = le.fit_transform(L)

# One-hot encoding untuk label
Y = to_categorical(Y, num_classes=5)

from sklearn.model_selection import train_test_split

# Mengatur seed untuk memastikan hasil yang konsisten
np.random.seed(2023)
random.seed(2023)

# Membagi dataset menjadi data latih (training) dan data uji (testing)
X_train, X_test, Y_train, Y_test = train_test_split(
    np.array(X), Y, test_size=0.2, random_state=42)

# Menampilkan ukuran data latih dan data uji untuk fitur (X) dan label (Y)
print(f"X_train: {X_train.shape}")
print(f"X_test: {X_test.shape}")
print(f"Y_train: {Y_train.shape}")
print(f"Y_test: {Y_test.shape}")
```

Gambar 7. Proses Normalisasi Gambar

Pada Gambar 7 ditunjukkan proses konversi label menggunakan labelEncoder dan *One-Hot Encoding*.

Setelah itu, dataset dibagi menjadi dua bagian, yaitu 80% untuk data latih (*training set*) dan 20% untuk data uji (*testing set*) dengan menggunakan *train_test_split*. Proses pembagian dilakukan secara acak (*random split*) untuk memastikan setiap kelas bunga terdistribusi secara proporsional pada kedua kelompok data. Proses ini juga menampilkan ukuran (*shape*) dari masing-masing data latih dan uji, baik untuk fitur (X) maupun label (Y), yang digunakan sebagai dasar dalam pelatihan model CNN.

Tabel 1 menunjukkan bahwa dataset citra bunga telah dibagi menjadi 3.453 sampel data latih dan 864

sampel data uji. Setiap citra memiliki dimensi 150×150 piksel dengan 3 kanal warna (RGB), sedangkan label hasil *One-Hot Encoding* memiliki dimensi lima yang merepresentasikan jumlah kelas bunga.

Tabel 1. Hasil Normalisasi Gambar

Data	Jumlah Sampel	Dimensi Input	Jumlah Kelas
X_train	3.453	$150 \times 150 \times 3$	–
X_test	864	$150 \times 150 \times 3$	–
Y_train	3.453	–	5
Y_test	864	–	5

4.3. Arsitektur Model CNN

Model *Convolutional Neural Network* (CNN) yang digunakan dalam penelitian ini dirancang untuk dapat mengekstraksi fitur visual dari citra bunga dan mengklasifikasikannya ke dalam lima kelas bunga. Arsitektur model CNN yang dibangun terdiri atas beberapa lapisan utama, yaitu:

- Convolutional Layer* dengan jumlah filter yang meningkat (32, 64, dan 96) menggunakan fungsi aktivasi ReLU dengan kernel 5×5 pixel untuk mengekstraksi pola dasar hingga kompleks.
- Max Pooling Layer* setelah setiap convolutional layer untuk mereduksi dimensi citra tanpa menghilangkan informasi penting.
- Flatten Layer* untuk mengubah hasil konvolusi menjadi vektor satu dimensi agar dapat diproses pada lapisan berikutnya.
- Dense Layer* yang berisi 512 neuron dengan fungsi sebagai pengklasifikasi utama, dilengkapi juga dengan fungsi aktivasi yaitu ReLU.
- Dropout Layer* sebesar 0,2 untuk mengurangi risiko *overfitting* dengan cara menonaktifkan sebagian neuron secara acak saat pelatihan.
- Output Layer* dengan 5 neuron dan fungsi aktivasi softmax untuk mengklasifikasikan citra ke dalam lima kelas bunga sesuai label.

Secara keseluruhan, jumlah parameter yang dilatih (*trainable parameters*) pada model ini adalah 12.662.309, yang menunjukkan kompleksitas jaringan cukup tinggi untuk menangani variasi citra bunga.

Tabel 2. Hasil Draf Arsitektur CNN

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 146, 146, 32)	2,432
max_pooling2d (MaxPooling2D)	(None, 73, 73, 32)	0
conv2d_1 (Conv2D)	(None, 71, 71, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 35, 35, 64)	0
conv2d_2 (Conv2D)	(None, 33, 33, 96)	55,392
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 96)	0
flatten (Flatten)	(None, 24576)	0
dense (Dense)	(None, 512)	12,583,424
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 5)	2,565
Total params: 12,662,309 (48.30 MB)		
Trainable params: 12,662,309 (48.30 MB)		
Non-trainable params: 0 (0.00 B)		

Berdasarkan Tabel 2, total parameter yang dilatih (*trainable parameters*) yaitu berjumlah 12.662.309, sementara *non-trainable parameter* bernilai nol.

Jumlah ini menunjukkan bahwa kompleksitas jaringan cukup tinggi sehingga model berpotensi mempelajari variasi visual bunga secara detail. Namun, tingginya jumlah parameter juga menuntut sumber daya komputasi yang besar dan meningkatkan risiko *overfitting* jika data pelatihan tidak mencukupi

4.4. Proses Pelatihan Model

Pelatihan model CNN dilakukan menggunakan data latih yang telah melalui tahap *preprocessing* dan augmentasi. Kompilasi model ini dilakukan dengan menggunakan *optimizer Adam* (learning rate = 0,001), fungsi *loss categorical_crossentropy* yang sesuai untuk klasifikasi multi-kelas, serta metrik evaluasi *accuracy*.

```
# Optimisasi model menggunakan algoritma adam
model.compile(optimizer=Adam(learning_rate=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Menyiapkan generator untuk augmentasi gambar
datagen = ImageDataGenerator(
    rotation_range=10,
    zoom_range=0.1,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True
)

datagen.fit(X_train)

# Training model CNN
train_dataset = datagen.flow(X_train, Y_train, batch_size=128)

history = model.fit(train_dataset,
                    epochs=50,
                    validation_data=(X_test, Y_test),
                    steps_per_epoch=X_train.shape[0] // 128)
```

Gambar 8. Pelatihan Model

Pada Gambar 8 menunjukkan bahwa pelatihan model CNN mencakup definisi algoritma optimasi, fungsi loss, dan metrik evaluasi. Kode tersebut juga menampilkan penggunaan *ImageDataGenerator* untuk melakukan augmentasi data guna memperluas variasi dataset agar lebih robust. Selanjutnya, model dilatih dengan data hasil augmentasi menggunakan batch size 128 selama 50 epoch, serta melibatkan data validasi untuk memantau kinerja model di luar data latih.

```
Epoch 48/50: 18s 438ms/step - accuracy: 0.6707 - loss: 0.6947 - val_accuracy: 0.7234 - val_loss: 0.7233
Epoch 49/50: 26/26 - accuracy: 0.7544 - loss: 0.6456 - val_accuracy: 0.7303 - val_loss: 0.7432
Epoch 42/50: 28s 820ms/step - accuracy: 0.7969 - loss: 0.6875 - val_accuracy: 0.7895 - val_loss: 0.7856
Epoch 43/50: 231s 8s/step - accuracy: 0.7633 - loss: 0.6856 - val_accuracy: 0.7326 - val_loss: 0.7280
Epoch 44/50: 26/26 - accuracy: 0.7422 - loss: 0.6125 - val_accuracy: 0.7269 - val_loss: 0.7527
Epoch 45/50: 199s 7s/step - accuracy: 0.7748 - loss: 0.5915 - val_accuracy: 0.7269 - val_loss: 0.7120
Epoch 46/50: 18s 442ms/step - accuracy: 0.7891 - loss: 0.6171 - val_accuracy: 0.7280 - val_loss: 0.7273
Epoch 47/50: 26/26 - accuracy: 0.7847 - loss: 0.5799 - val_accuracy: 0.7407 - val_loss: 0.6948
Epoch 48/50: 19s 477ms/step - accuracy: 0.7578 - loss: 0.7044 - val_accuracy: 0.7407 - val_loss: 0.6971
Epoch 49/50: 199s 7s/step - accuracy: 0.7899 - loss: 0.5604 - val_accuracy: 0.7361 - val_loss: 0.6911
Epoch 50/50: 18s 490ms/step - accuracy: 0.7812 - loss: 0.6186 - val_accuracy: 0.7407 - val_loss: 0.7038
```

Gambar 9. Hasil Pelatihan Model

Selama pelatihan, setiap epoch menghasilkan laporan perkembangan akurasi dan loss baik pada data latih maupun validasi. Gambar 9 menampilkan hasil keluaran proses pelatihan yang memuat nilai akurasi dan loss pada setiap epoch. Dari hasil tersebut terlihat bahwa akurasi data latih meningkat secara bertahap,

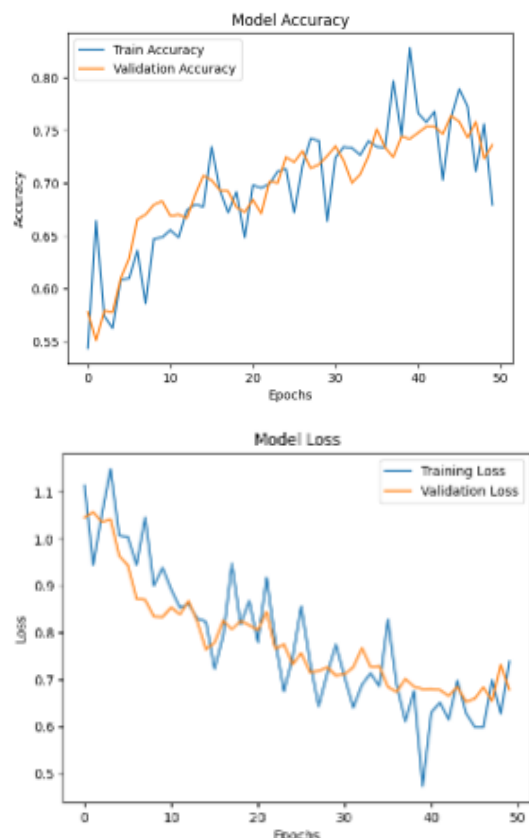
dimulai dari sekitar 63% pada awal pelatihan hingga mencapai 78,9% pada epoch terakhir. Sementara itu, akurasi validasi cenderung stabil di kisaran 73–74%.

Nilai loss pada data latih menunjukkan tren penurunan signifikan, dari awal sekitar 0,95 menjadi 0,56 pada epoch terakhir, menandakan bahwa model mampu meminimalkan kesalahan prediksi pada data latih. Namun, nilai loss validasi masih berfluktuasi pada kisaran 0,69–0,78, yang menunjukkan adanya *overfitting* ringan, yaitu suatu kondisi di mana model belajar sangat baik pada data latih tetapi kurang optimal dalam generalisasi terhadap data baru.

Hasil ini menunjukkan bahwa model CNN yang dibangun berhasil mempelajari pola visual dari citra bunga dengan cukup baik, meskipun masih terdapat ruang untuk perbaikan pada aspek generalisasi model.

4.5. Proses Pengujian Model

Setelah proses pelatihan selesai, model diuji menggunakan dataset uji yang terdiri dari 864 citra bunga untuk mengevaluasi kinerjanya pada data baru yang tidak digunakan dalam pelatihan model. Untuk mengilustrasikan proses pembelajaran dan performa pada data validasi, hasil pelatihan divisualisasikan dalam bentuk grafik akurasi dan grafik loss, seperti yang ditunjukkan pada Gambar 10.



Gambar 10. Grafik Akurasi dan Loss

Berdasarkan grafik, terlihat bahwa akurasi pelatihan meningkat secara konsisten sepanjang epoch, sedangkan akurasi validasi cenderung stabil pada kisaran 70–74%. Hal ini menunjukkan bahwa

model mampu mempelajari pola dari data latih, tetapi generalisasi terhadap data validasi relatif terbatas. Pada sisi lain, loss pelatihan terus menurun secara signifikan, sedangkan loss validasi berfluktuasi dan tidak menunjukkan penurunan yang konsisten setelah beberapa epoch. Pola ini menandakan adanya indikasi overfitting, di mana model lebih unggul pada data latih dibandingkan pada data validasi.

Evaluasi pada data uji memberikan gambaran objektif mengenai kemampuan generalisasi model. Hal ini mengindikasikan bahwa meskipun model cukup efektif dalam mengenali pola visual dari bunga, masih terdapat ruang untuk peningkatan performa, khususnya dalam mengurangi kesalahan klasifikasi pada kelas bunga yang memiliki kemiripan visual.

4.6. Analisis Hasil

Evaluasi dilakukan dengan menghitung metrik *accuracy*, *precision*, *recall*, dan *f1-score* untuk setiap kelas, serta rata-rata tertimbang (*weighted average*). Hasil evaluasi menunjukkan bahwa model CNN yang dibangun mampu mencapai akurasi sebesar 74,07%, dengan nilai rata-rata pada *precision* sebesar 74,95%, *recall* 74,07%, dan *f1-score* 73,91%.

Tabel 3. Hasil Evaluasi

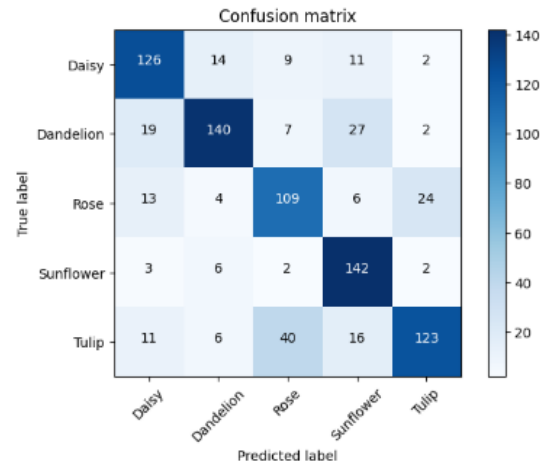
	precision	recall	f1-score	support
Daisy	0.73	0.78	0.75	162
Dandelion	0.82	0.72	0.77	195
Rose	0.65	0.70	0.67	156
Sunflower	0.70	0.92	0.80	155
Tulip	0.80	0.63	0.70	196
accuracy			0.74	864
macro avg	0.74	0.75	0.74	864
weighted avg	0.75	0.74	0.74	864

Dari Tabel 3 diatas, terlihat adanya variasi pada performa. Model memiliki performa terbaik pada kelas *sunflower*, dengan nilai *recall* mencapai 92%, yang menunjukkan kemampuan tinggi dalam mengenali citra bunga ini. Sebaliknya, performa terendah terdapat pada kelas *rose*, dengan *precision* hanya 65%, yang berarti model sering salah mengklasifikasikan bunga *rose* ke kelas lain yang memiliki kemiripan visual. Sementara itu, kelas *daisy*, *tulip*, dan *dandelion* menunjukkan hasil yang cukup seimbang dengan *precision* dan *recall* berada pada kisaran 70–75%.

Temuan ini mengindikasikan bahwa CNN cukup efektif dalam melakukan klasifikasi citra, namun masih menghadapi tantangan dalam membedakan kelas yang memiliki karakteristik visual mirip. Hasil ini sejalan dengan evaluasi grafik akurasi dan loss pada tahap sebelumnya, yang menunjukkan bahwa adanya keterbatasan generalisasi model. Oleh karena itu, peningkatan performa masih dimungkinkan, misalnya dengan menambah jumlah data latih, menggunakan arsitektur CNN yang lebih dalam, atau menerapkan teknik regularisasi yang lebih optimal.

4.7. Confusion Matrix

Confusion matrix digunakan untuk menganalisis distribusi hasil prediksi model terhadap label sebenarnya. Hasil *confusion matrix* ini menunjukkan bahwa sebagian besar kelas dikenali dengan baik oleh model, terutama pada kelas *sunflower* dan *daisy* yang mayoritas diprediksi benar. Namun, kesalahan klasifikasi masih terjadi, misalnya beberapa citra *rose* salah diklasifikasikan sebagai *tulip*, serta beberapa citra *daisy* salah dikenali sebagai *dandelion*.



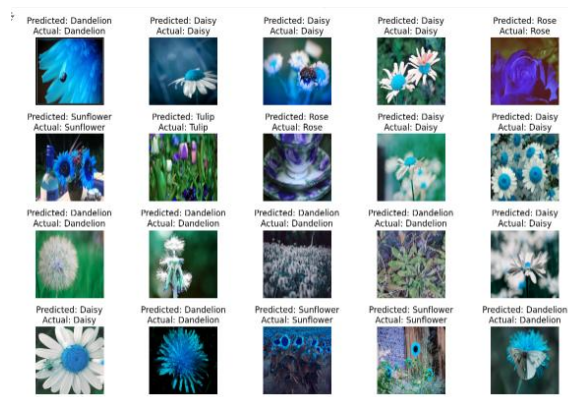
Gambar 11. Confusion Matrix

Analisis hasil yang ditunjukkan pada Gambar 11 mengindikasikan bahwa kesalahan klasifikasi banyak dipengaruhi oleh kemiripan visual antarspesies bunga, baik dari segi bentuk kelopak maupun warna, sehingga model memerlukan peningkatan dalam mengekstraksi fitur yang lebih spesifik. Hal ini konsisten dengan hasil evaluasi pada Tabel 3, di mana nilai *precision* dan *recall* untuk ketiga kelas tersebut berada pada kisaran 70–75%. Secara keseluruhan, *confusion matrix* ini dapat memberikan gambaran visual yang lebih detail mengenai distribusi kesalahan klasifikasi.

4.8. Klasifikasi Benar dan Salah

Selain evaluasi kuantitatif, analisis kualitatif juga dilakukan dengan melihat contoh citra yang berhasil dan gagal diklasifikasikan oleh model. Pada prediksi yang benar, model mampu mengenali bunga dengan baik ketika citra memiliki kualitas tinggi, latar belakang sederhana, serta objek bunga dominan dalam gambar. Misalnya, bunga *sunflower* dengan bentuk dan warna yang khas hampir selalu diprediksi tepat.

Sebaliknya, pada prediksi yang salah, ditemukan beberapa pola penyebab kesalahan. Pertama, adanya kemiripan visual antarspesies, seperti *rose* yang sering terklasifikasi sebagai *tulip* karena bentuk kelopaknya mirip. Kedua, citra dengan latar belakang kompleks atau pencahayaan buruk membuat model kesulitan mengekstraksi fitur penting. Ketiga, bunga yang tidak ditampilkan secara utuh, misalnya hanya sebagian kelopak terlihat atau bunga masih dalam kuncup juga meningkatkan risiko salah klasifikasi.



Gambar 12. Prediksi Citra Bunga

Analisis prediksi citra bunga pada Gambar 12 menunjukkan bahwa meskipun model cukup andal dalam mengenali citra bunga dengan kualitas baik, performanya masih perlu ditingkatkan lagi untuk menangani citra dengan kondisi yang lebih kompleks.

5. KESIMPULAN DAN SARAN

Penelitian ini membangun model klasifikasi citra bunga menggunakan *Convolutional Neural Network* (CNN) dengan lima kelas bunga, yaitu *daisy*, *rose*, *sunflower*, *dandelion*, dan *tulip*, menggunakan dataset Flowers Recognition dari Kaggle yang berjumlah 4.317 citra. Hasil evaluasi menunjukkan bahwa model telah mencapai akurasi sebesar 74,07%. Berdasarkan *confusion matrix*, kelas *sunflower* memiliki performa terbaik dengan *recall* mencapai 92%, sedangkan kesalahan klasifikasi paling banyak terjadi pada kelas *rose* dan *tulip* karena memiliki kemiripan visual. Secara keseluruhan, CNN terbukti cukup efektif dalam melakukan klasifikasi citra bunga, meskipun masih terdapat adanya keterbatasan dalam membedakan kelas dengan karakteristik serupa. Untuk penelitian selanjutnya, disarankan penerapan *transfer learning* model *pretrained* (VGG, ResNet, atau EfficientNet), penambahan variasi augmentasi data, dan penggunaan teknik regulasi seperti *EarlyStopping* untuk dapat mengurangi terjadinya *overfitting* pada model.

DAFTAR PUSTAKA

- [1] Z. Wang, K. Wang, X. Wang, and S. Pan, "A Convolutional Neural Network Ensemble for Flower Image Classification," in *Proceedings of the 2020 9th International Conference on Computing and Pattern Recognition*, Xiamen China: ACM, Oct. 2020, pp. 225–230. doi: 10.1145/3436369.3437427.
- [2] M. F. Rabbi *et al.*, "An Ensemble-based Deep Learning Model for Multi-class Flower Recognition," in *2023 International Conference on Next-Generation Computing, IoT and Machine Learning (NCIM)*, Gazipur, Bangladesh: IEEE, June 2023, pp. 1–6. doi: 10.1109/NCIM59001.2023.10212540.
- [3] S. Anand, M. K. Swaroopa, M. Nainwal, and T. M., "An intelligent flower classification framework: optimal hybrid flower pattern extractor with adaptive dynamic ensemble transfer learning-based convolutional neural network," *Imaging Sci. J.*, vol. 72, no. 1, pp. 52–75, Jan. 2024, doi: 10.1080/13682199.2023.2183317.
- [4] A. Pratiwi and A. Fauzi, "IMPLEMENTATION OF DEEP LEARNING ON FLOWER CLASSIFICATION USING CNN METHOD," *J. Tek. Inform. Jutif*, vol. 5, no. 2, pp. 487–495, Apr. 2024, doi: 10.52436/1.jutif.2024.5.2.1674.
- [5] I. Gilang Perwati, N. Suarna, and T. Suprpti, "ANALISIS KLASIFIKASI GAMBAR BUNGA LILY MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) DALAM PENGOLAHAN CITRA," *JATI J. Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 2908–2915, May 2024, doi: 10.36040/jati.v8i3.9193.
- [6] H. A. Pratiwi, M. Cahyanti, and M. Lamsani, "IMPLEMENTASI DEEP LEARNING FLOWER SCANNER MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK," *Sebatik*, vol. 25, no. 1, pp. 124–130, June 2021, doi: 10.46984/sebatik.v25i1.1297.
- [7] A. Rahman, M. Salim, and I. Riadi, "KLASIFIKASI CITRA SPESIES BUNGA DI INDONESIA BERBASIS CONVOLUTIONAL NEURAL NETWORK MENGGUNAKAN TEKNIK TRANSFER LEARNING," *J. Softw. Eng. Comput. Intell.*, vol. 2, no. 02, pp. 92–100, Jan. 2025, doi: 10.36982/jseci.v2i02.4942.
- [8] G. W. Intyanto, "Klasifikasi Citra Bunga dengan Menggunakan Deep Learning: CNN (Convolution Neural Network)," *J. Arus Elektro Indones.*, vol. 7, no. 3, p. 80, Dec. 2021, doi: 10.19184/jaei.v7i3.28141.
- [9] A. N. R. Munandar and A. F. Rozi, "Analisis Arsitektur Convolutional Neural Network Untuk Klasifikasi Citra Bunga," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 6, no. 3, pp. 522–531, July 2024, doi: 10.47233/jteksis.v6i3.1413.
- [10] Q. Aini, Z. Zulfiandri, R. Firmansyah, and Y. M. Arif, "Applying convolutional neural network and Nadam optimization in flower classification," *Bull. Electr. Eng. Inform.*, vol. 13, no. 4, pp. 2865–2877, Aug. 2024, doi: 10.11591/eei.v13i4.6203.
- [11] Z. Soomro, "IMAGE RECOGNITION ON FLOWER CLASSIFICATION USING NEURAL NETWORK," 2022.
- [12] M. Hanafiah, M. A. Adnan, S. Abdul-Rahman, S. Mutalib, A. M. A. Malik, and M. R. Shamsuddin, "Flower Recognition using Deep Convolutional Neural Networks," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 1019, no. 1, p. 012021, Apr. 2022, doi: 10.1088/1755-1315/1019/1/012021.
- [13] H. Noprisson, "Fine-Tuning Model Transfer Learning VGG16 Untuk Klasifikasi Citra Penyakit Tanaman Padi," *JSAI J. Sci. Appl.*

- Inform.*, vol. 5, no. 3, pp. 244–249, Nov. 2022, doi: 10.36085/jsai.v5i3.3609.
- [14] Zulkipli, L. A. S. I. Akbar, and B. Kanata, “KLASIFIKASI GENUS TANAMAN ANGGREK MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN MENGGUNAKAN ARSITEKTUR VGG 16,” *J. Inform. Teknol. Dan Sains Jinteks*, vol. 6, no. 4, pp. 820–829, Nov. 2024, doi: 10.51401/jinteks.v6i4.4982.
- [15] F. Fitriani, “KLASIFIKASI JENIS BUNGA DENGAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN),” *Tek. Teknol. Inf. Dan Multimed.*, vol. 2, no. 2, pp. 64–68, Dec. 2021, doi: 10.46764/teknimedia.v2i2.39.