

KLASIFIKASI DAN PENGENALAN POLA PENYAKIT CABAI DENGAN METODE CNN (*Convolution Neural Network*)

Bella Nurbuana Tri Cahya Ningrum¹, Erlina Nasrinatun Ni'mah², Miranda Putri Arifin³,
Made Ayu Dusea Widya Dara⁴

^{1,2,3,4}Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri

E-mail: *¹bellanurbuana@gmail.com, ²erlinann15@gmail.com, ³mirandaputriarifin@gmail.com,
⁴madedara@gmail.com

Abstrak – Penelitian ini mengusulkan sistem klasifikasi menggunakan *Convolutional Neural Network* (CNN) untuk identifikasi penyakit pada buah cabai. Dalam pengujian dengan 162 citra, algoritma mencapai akurasi 100%. Namun, saat diuji dengan 236 citra, termasuk data penyakit lalat yang belum dikenali sistem, akurasi menurun drastis menjadi 45%. Penyakit seperti bercak abu, virus kuning cabai, dan layu bakteri telah berhasil diidentifikasi dengan presisi tinggi. Meskipun demikian, tantangan muncul ketika data penyakit baru diuji, menyebabkan penurunan signifikan dalam performa sistem. Hasil ini menggarisbawahi pentingnya pengembangan model yang responsif terhadap penyakit baru serta kebutuhan akan dataset yang lebih luas dan representatif. Sistem ini menawarkan potensi besar dalam mendukung petani cabai dalam mendeteksi penyakit, namun membutuhkan peningkatan agar dapat mengenali variasi penyakit yang lebih luas dan tetap handal dalam situasi lingkungan.

Kata Kunci — Cabai, CNN, Klasifikasi, Performa

1. PENDAHULUAN

Deteksi penyakit tanaman merupakan hal terpenting dalam memelihara dan merawat tanaman. Penyakit yang tidak bisa terdeteksi dan dibiarkan begitu saja akan mengakibatkan kerusakan pada tanaman. Kerusakan tersebut akan berdampak bagi ekonomi dan petani. Beberapa penelitian telah dilakukan untuk mendapat pengetahuan tentang berbagai penyakit tanaman, sehingga perawatan pada tanaman menjadi lebih mudah dengan mengetahui ciri-ciri dari setiap penyakit, dan cara pengobatan tanaman, hingga cara pencegahan penyakit [1].

Perkembangan teknologi saat ini telah mendorong cara untuk mendeteksi penyakit menggunakan suatu sistem yang dapat dijalankan melalui *smartphone* atau komputer. Sistem deteksi ini sangat direkomendasikan karena hasil akurasi yang dihasilkan cukup akurat. Banyak penelitian yang menggunakan metode *Deep Learning* sebagai pondasi untuk membuat sistem. Banyak penelitian yang dilakukandengan memanfaatkan sistem klasifikasi untuk mendiagnosa suatu penyakit dengan menggunakan objek gambar sebagai masukannya dan sistem ini dibangun untuk menggantikan peran seorang pakar. Dengan menggunakan sistem klasifikasi ini dapat mempermudah dalam mendiagnosa penyakit pada tanaman cabai.

Adapun beberapa penelitian yang dilakukan oleh peneliti sebelumnya, yaitu penelitian yang dilakukan oleh Uly Nuhanatika yang berjudul “Penentuan Tingkat Kematangan Cabe Rawit (*Capsicum frutescens* L.) Berdasarkan *Gray Level Co-Occurrence Matrix*” bertujuan untuk klasifikasi tingkat kematangan cabai rawit menggunakan ekstraksi fitur warna dan tekstur. Ekstraksi fitur warna diambil dari nilai mean saturasi, sedangkan ekstraksi fitur tekstur menggunakan nilai fitur *Gray level Co-Occurrence Matrix* (GLCM). Menghasilkan pelatihan sistem yang mampu mengklasifikasi tingkat kematangan cabai rawit dengan akurasi sebesar 81,4% dan akurasi proses pengujian cabai rawit sebesar 74,2% [2]. Selanjutnya penelitian yang dilakukan oleh Joelyan Vicky, Frisca Ayu dan Bagas Julianto yang berjudul “Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN” dimana proses implementasi yang dilakukan dengan model dan testing data pada data daun alpukat yang terkena penyakit mempunyai hasil yang baik dengan menggunakan metode CNN mampu mendeteksi penyakit pada daun alpukat dengan akurasi sebesar 80% [3]. Selanjutnya penelitian yang dilakukan oleh Alya Zalvadila, Purnawansyah, Lukman Syafie, Herdianti Darwis yang berjudul “Klasifikasi Penyakit Tanaman Bawang Merah Menggunakan Metode SVM dan CNN” penelitian ini menggunakan CNN dan SVM dengan kernel RBF, linear, dan polynominal yang memberikan nilai accuracy, precision, recall, dan F1 score yang masing-masing bernilai 100%. SVM diawali dengan proses ekstraksi fitur menggunakan GLCM, apabila SVM menggunakan kernel sigmoid akan menghasilkan performa yang jauh lebih buruk [4].

Petani sering menghadapi tantangan dalam mengidentifikasi secara akurat penyakit yang menyerang tanaman cabai. Keterbatasan pengetahuan dan pengalaman petani dapat menyebabkan kesulitan dalam

membedakan gejala penyakit yang serupa. Oleh karena itu, kebutuhan akan sistem klasifikasi penyakit tanaman cabai menjadi penting. Dengan adanya sistem ini, petani dapat dengan cepat dan akurat mengidentifikasi jenis penyakit yang menyerang tanaman cabai mereka. Hal ini tidak hanya memungkinkan tindakan pencegahan yang lebih dini dan tepat, tetapi juga membantu dalam optimalisasi penggunaan sumber daya, mengurangi kerugian hasil panen, dan secara keseluruhan meningkatkan produktivitas pertanian dengan cara yang lebih efisien dan berkelanjutan.

Berdasarkan permasalahan yang telah disebutkan diatas maka peneliti membuat penelitian tentang klasifikasi dan pengolahan citra penyakit pada tanaman cabai menggunakan metode CNN (*Convolution Neural Network*) dengan proses deteksi warna yang diawali dengan konversi ruang warna citra RGB (*Red, Green, Blue*) menjadi HSV (*Hue, Saturation, Value*). Pada proses klasifikasi warna dilakukan berdasarkan pengelompokan nilai Hue, selanjutnya proses thresholding yang menghasilkan citra biner. Setelah itu dilakukan proses ekstraksi ciri morfologi dari citra biner berdasarkan parameter eccentricity dan metric. Hasil penelitian diharapkan mampu membedakan jenis penyakit pada tanaman cabai.

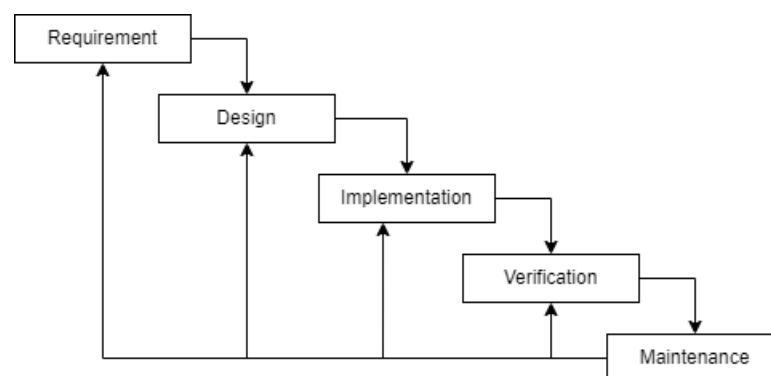
2. METODE PENELITIAN

2.1 Pengumpulan Data

Metode yang digunakan untuk pengumpulan data pada penelitian ini adalah studi pustaka, yaitu dalam pencarian teori dan pengumpulan data, peneliti menyimpulkan informasi dari kepustakaan yang berhubungan dengan penelitian tersebut. Sumber yang digunakan oleh penulis yaitu dari buku, jurnal penelitian yang sebelumnya yang berkaitan dengan topik penelitian tersebut, serta sumber lainnya yang di dapatkan melalui internet.

2.2 Metode Waterfall

Dalam perancangan sistem klasifikasi ini menggunakan metode *waterfall*. Metode ini terdiri dari 5 proses dalam perancangan sistem yaitu *requirement*, *design*, *implementation*, *verification*, dan *maintenance*, Seperti yang terlihat pada gambar 1.



Gambar 1. Metode Waterfall

2.3 Pengolahan Citra

a. Segmentasi Warna RGB

Segmentasi Warna RGB adalah proses pemisahan objek pada citra berwarna menggunakan model RGB (*Red, Green, Blue*). Pada setiap piksel dalam model warna RGB memiliki 3 Saluran warna: merah, hijau dan biru. Segmentasi warna RGB bertujuan untuk mengidentifikasi dan memisahkan objek dalam citra berdasarkan perbedaan warna dalam saluran warna RGB.

b. Ruang warna HSV

Ruang warna HSV merupakan model warna yang digunakan untuk mempresentasikan warna dalam citra. Dalam ruang warna HSV ini memisahkan warna menjadi 3 komponen utama, yaitu Hue(nuansa), Saturation (saturasi), dan Value (nilai).

a. Hue mempresentasikan jenis warna yang ada dalam citra dengan menunjukkan warna yang sebenarnya, yaitu merah, hijau, biru, kuning, cyan, magenta. Hue memiliki rentang angka dari 0 hingga 360 derajat.

b. Saturasi mengukur sejauh mana warna itu jenuh atau tidak. Nilai saturasi yang tinggi menunjukkan warna yang jenuh dan kuat, sementara nilai saturasi rendah menghasilkan warna yang pudar atau abu-abu.

- c. Value menggambarkan kecerahan dari warna yang nilai nya berkisar dai 0 sampai 100%, 0 adalah hitam dan 100% adalah putih

Ruang warna HSV diperoleh dari ruang warna RGB dengan persamaan 1, 2, 3, 4 dan 5:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \dots\dots\dots(1)$$

$$V = \max (r, g, b) \dots\dots\dots(2)$$

$$S = \left\{ \begin{array}{l} 0, \text{ jika } V = 0 \\ 1 - \frac{\min(r,g,b)}{v}, v > 0 \end{array} \right\} \dots\dots\dots(3)$$

$$H = \left\{ \begin{array}{l} 0, \text{ jika } S = 0 \\ \frac{60*(g-b)}{S*V}, \text{ jika } V = r \\ 60 * \left[2 + \frac{b-r}{S*V} \right], \text{ jika } V = g \\ 60 * \left[4 + \frac{r-g}{S*V} \right], \text{ jika } V = b \end{array} \right\} \dots\dots\dots(4)$$

$$H = H + 360 \text{ jika } H < 0 \dots\dots\dots(5)$$

2.4 Algoritma CNN

Convolution Neural Network merupakan model jaringan saraf tiruan yang digunakan untuk mengelola citra dan pengenalan pola. CNN digunakan untuk ekstrasi fitur-fitur visual penting dari citra, hal ini sangat berguna dalam berbagai sistem aplikasi pengenalan objek, klasifikasi citra, dan segmentasi citra [1]. Algoritma CNN dipergunakan untuk Deep Learning dengan mengambil input dan memberikan bobot dan bias yang dapat dipelajari pada fitur dalam dibandingkan dengan klasifikasi algoritma lainnya [5]. Filter metode primitive direkayasa secara manual, dengan pelatihan yang cukup, dengan Langkah – Langkah sebagai berikut :

- (1) Inisialisasi parameter model, langkah yang setara dengan menyuntikkan noise ke dalam model
- (2) for $i = 1, 2, \dots n$ (n adalah nomor epoch)
 - a. Performa dengan *forward propagation* :

$\forall i$, hitung nilai prediksi dari a melalui jaringan saraf tiruan : $\hat{y} i \theta$

Evaluasi fungsi dapat dilihat pada persamaan 2 :

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \delta(\hat{y}_i \theta, y_i) \dots\dots\dots(6)$$

Dimana m adalah ukuran dari data training dan θ adalah model parameter dan (*) fungsi perkalian.

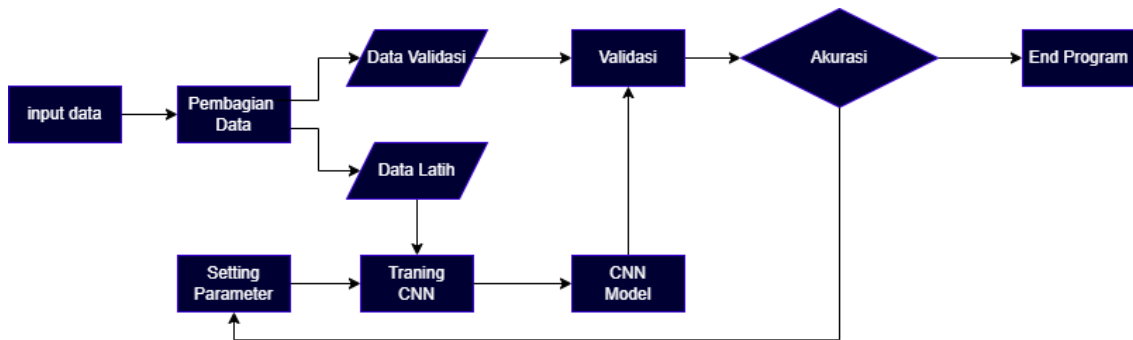
- b. Performa dengan *back propagation* :

Terapkan metode penurunan untuk memperbarui parameter dengan persamaan 7 :

$$\theta =: G () \dots\dots\dots(7)$$

2.5 Flowchart

Pada gambar 2 akan ditampilkan diagram alur proses pengolahan data, pembersihan data, pelatihan dan pengujian, serta evaluasi untuk mencapai akurasi terbaik dalam mengidentifikasi citra penyakit pada cabai mentah dan kering.. [6]



Gambar 2. Flowchart

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

Dalam penelitian ini, digunakan data citra digital cabai jenis Asmoro yang diambil melalui kamera Hp dengan resolusi sebesar 104 Megapixel. Citra digital tersebut dibagi menjadi dua kategori, yaitu kering dan mentah, dengan total 162 citra dalam sampel, terdiri dari 105 citra kering dan 57 citra mentah. Data ini kemudian terbagi menjadi dua bagian, yakni data training dan data testing, dengan perbandingan 80% untuk data training dan 20% untuk data testing. Pada proses klasifikasi citra digital, digunakan perangkat lunak Jupiter dan Visual Studio Code. Visual Studio Code memanfaatkan beberapa package, antara lain TensorFlow Keras, Cv2, dan Numpy, untuk mendukung proses analisis. Tahapan analisis dalam penelitian ini melibatkan langkah-langkah sebagai berikut:

1. *Import Library:*

Import library yang diperlukan seperti *train_test_split* untuk membagi data menjadi set pelatihan dan pengujian, *LabelEncoder* untuk mengubah label menjadi bentuk biner, dan beberapa komponen dari *TensorFlow* untuk membangun model *neural network*.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import InputLayer, Flatten, Dense, Conv2D, MaxPool2D, Dropout
from tensorflow.keras.optimizers import Adam
import numpy as np
import cv2
import glob
import matplotlib.pyplot as plt
```

Gambar 3. *Import Library*

2. *Load Dataset:*

Definisikan path dataset dan daftar label terkait. Selanjutnya, lakukan iterasi untuk membaca setiap gambar dalam dataset. Selama iterasi, ubah ukuran setiap gambar menjadi (32, 32), dan tambahkan gambar yang telah diubah ukurannya beserta labelnya ke dalam array data dan labels secara bersamaan.



Gambar 4. Dataset Cabai Kering



Gambar 5. Dataset Cabai Mentah

3. *Preprocessing Data:*
Konversi data menjadi *array NumPy* dan normalisasi nilai pikselnya ke rentang [0, 1]. Menggunakan *LabelEncoder* untuk mengubah label menjadi *format biner*.
4. *Pembagian Data:*
Memisahkan data menjadi set pelatihan (x_{train} , y_{train}) dan set pengujian (x_{test} , y_{test}) menggunakan *train_test_split*.
5. *Pembangunan Model:*
Membangun model *sequential* menggunakan *TensorFlow*. Model terdiri dari beberapa layer *Convolutional Neural Network (CNN)* untuk mengekstrak fitur dan beberapa layer *Fully Connected* untuk klasifikasi.
6. *Compile Model:*
Mengompilasi model dengan menggunakan fungsi kerugian (*binary_crossentropy*) dan pengoptimal (*Adam*). *Metric* yang diukur adalah akurasi.
7. *Pelatihan Model:*
Melakukan pelatihan model dengan data pelatihan (x_{train} , y_{train}) selama beberapa *epoch*. Hasil pelatihan disimpan dalam variabel *H*.
8. *Evaluasi Model:*
Menggunakan data pengujian (x_{test} , y_{test}) untuk mengevaluasi model. Mencetak laporan klasifikasi dan menggambar grafik akurasi pada set pelatihan dan pengujian.
9. *Prediksi dengan Data Baru:*
Menggunakan model yang telah dilatih untuk melakukan prediksi pada gambar baru (mentah (16).jpg). Menampilkan hasil prediksi dengan label yang sesuai pada gambar tersebut.
10. *Tampilkan Hasil:*
Menampilkan gambar hasil prediksi dengan menambahkan teks label di atasnya.



Gambar 6. Hasil Gambar Terpilih

3.2 Analisa Sistem

Hasil Analisa sistem menunjukkan kemampuan pembacaan banyak data baru yang diinputkan melalui preprocessing untuk klasifikasi penyakit pada buah cabai [7]. Pengujian ini dilakukan dengan testing dataset yang berisi 74 data citra yang belum dikenali serta data yang sudah dikenali oleh sistem.



Gambar 7. Pengujian Data Baru

Data baru yang berisi penyakit lalat telah dilakukan pengenalan ke dalam sistem, kemudian di hitung akurasi dengan dataset yang sudah dikenali oleh sistem sehingga mendapatkan akurasi yang berubah sedikit lebih turun dari akurasi sebelumnya, yaitu sebagai berikut :

```
# menghitung nilai akurasi model terhadap data test
predictions = model.predict(x_test, batch_size=32)
target = (predictions > 0.5).astype(np.int64) # or np.int32 or other appropriate types
print(classification_report(y_test, target, target_names=label_list))

0.6s
1/2 [=====] - 0s 20ms/step
precision    recall  f1-score   support

 mentah       1.00     0.70     0.82         28
  kering       0.50     1.00     0.67         17
   lalat       0.00     0.00     0.00          11

 accuracy          0.58
 macro avg          0.57
weighted avg          0.58
```

Gambar 8. Akurasi data baru ditambahkan

```
# menghitung nilai akurasi model terhadap data test
predictions = model.predict(x_test, batch_size=17)
target = (predictions > 0.5).astype(np.int64) # or np.int32 or other appropriate type
print(classification_report(y_test, target, target_names=label_list))
```

```
2/2 [-----] - 0s 0%/step
```

	precision	recall	f1-score	support
mentah	1.00	1.00	1.00	28
kering	1.00	1.00	1.00	13
accuracy			1.00	33
macro avg	1.00	1.00	1.00	33
weighted avg	1.00	1.00	1.00	33

Gambar 9. Akurasi data baru yang sudah dikenali

Berdasarkan perhitungan pada analisa sistem diatas, dapat diambil kesimpulan bahwa akurasi algoritma Convolutional Neural Network dalam klasifikasi penyakit buah cabai ini dengan menggunakan 162 citra yang sudah dikenali oleh sistem memiliki akurasi 1,0 atau 100%, namun jika ditambahkan data baru sebanyak 74 data, sehingga memiliki total jumlah data keseluruhan adalah 236 data, maka akurasi yang terjadi menurun menjadi 0,4 atau 45%.

4. SIMPULAN

Berdasarkan hasil implementasi yang telah dilakukan, kesimpulan yang didapatkan dari hasil penelitian yang telah dilakukan adalah sebagai berikut:

1. Sistem sudah berhasil mengimplementasikan metode CNN untuk mendeteksi penyakit pada cabai.
2. Dengan adanya sistem ini dapat membantu mengetahui sejauh mana performa CNN dalam mengklasifikasikan data buah cabai dalam bentuk citra
3. Model CNN yang telah dibuat berhasil mendapatkan hasil akurasi yang dapat dilihat perbandingannya untuk menjadi pertimbangan penelitian kedepannya.
4. Data yang digunakan masih belum terlalu mempengaruhi perubahan yang lebih spesifik dan kompleks

5. SARAN

Berdasarkan hasil pembuatan sistem klasifikasi dan pengenalan pola untuk mendeteksi penyakit pada cabai menggunakan metode CNN, dalam sistem masih terdapat kesalahan maupun kekurangan. Oleh karena itu ada beberapa saran yang dapat diberikan yaitu bisa mengimplementasikan model tersebut dalam sistem berbasis android ataupun web agar mudah di akses siapa saja.

Hasil uji yang sudah diterapkan dapat membantu kemudahan mengetahui seberapa akurat dan seberapa baik performa dari CNN untuk klasifikasi ini. Kedepannya peneliti dapat membuat perhitungan hasil akurasi yang lebih unggul lagi dari penelitian sebelumnya.

DAFTAR PUSTAKA

[1] A. T. R. Dzaky, “Deteksi Penyakit Tanaman Cabai Menggunakan Metode Convolutional Neural Network,” *e-Proceeding Eng.*, vol. 8, no. 2, pp. 3040–3055, 2021.

[2] Z. E. Fitri, U. Nuhanatika, A. Madjid, and A. M. N. Imron, “Penentuan Tingkat Kematangan Cabe Rawit (*Capsicum frutescens* L.) Berdasarkan Gray Level Co-Occurrence Matrix,” *J. Teknol. Inf. dan Terap.*, vol. 7, no. 1, pp. 1–5, 2020, doi: 10.25047/jtit.v7i1.121.

[3] J. V. P. Putra, F. Ayu, and B. Julianto, “Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN,” *Stain. (Seminar Nas. Teknol. Sains)*, vol. 2, no. 1, pp. 155–162, 2023.

[4] A. Zalvadila, L. Syafie, and H. Darwis, “Klasifikasi Penyakit Tanaman Bawang Merah Menggunakan Metode SVM dan CNN,” *J. Inform. J. Pengemb. IT*, vol. 8, no. 3, pp. 255–260, 2023.

- [5] P. Kalkura, P. R. B, S. K. N, Surya, and M. Ramyashree, “Pest control management system using organic pesticides,” *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 175–180, 2021, doi: 10.1016/j.gltp.2021.08.058.
- [6] D. Irfansyah, M. Mustikasari, and A. Suroso, “Arsitektur Convolutional Neural Network (CNN) Alexnet Untuk Klasifikasi Hama Pada Citra Daun Tanaman Kopi,” *J. Inform. J. Pengemb. IT*, vol. 6, no. 2, pp. 87–92, 2021.
- [7] J. Kecerdasan, T. Informasi, D. H. Firdaus, B. Imran, L. D. Bakti, and E. Suryadi, “Klasifikasi Penyakit Katarak Pada Mata Menggunakan Metode Convolutional Neural Network (Cnn) Berbasis Web Web-Based Classification of Cataract in the Eyes Using Convolutional Neural Network (Cnn) Method,” *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 1, no. 3, pp. 18–26, 2022.