



Analisa Penyakit Pada Tanaman Cabai Merah (*Capsicum Annuum L*) dengan Membandingkan Tingkat Akurasi Menggunakan Metode *Convolutional Neural Network* (CNN) dan *K-Nearest Neighbor* (KNN)

Ade Arya Anggara^{1*}, Abdurrahman Ridho², Cut Mutia³

^{1,2,3} Teknologi Informasi, Fakultas Teknik, Universitas Teuku Umar

Jl. Alue Peunyareng, Ujong Tanog Darat, Meurebo, Kabupaten Aceh Barar, Aceh, Indonesia

Email: ¹adearyaanggara@gmail.com, ²abdurrahman.ridho@utu.ac.id, ³cutmutia@utu.ac.id

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima: 19 April 2025
Revisi: 02 Mei 2025
Diterbitkan: 30 mei 2025

Kata Kunci:
Cabai Merah
Convolutional Neural Network
K-Nearest Neighbor
VGG16
Confusion Matrix

ABSTRAK

Cabai merah (*Capsicum annuum L*) adalah bumbu pokok yang ada di hampir setiap masakan Indonesia dan memiliki nilai jual tinggi dibandingkan tanaman lain. Namun, banyak faktor serangan yang terjadi baik biotik maupun abiotik, sehingga dapat menyebabkan gagalnya panen. Identifikasi yang cepat dapat mengurangi risiko gagalnya panen ini. Salah satu solusinya adalah menggunakan teknologi *computer vision* untuk mengklasifikasikan citra penyakit tanaman cabai merah. Dalam penelitian ini, digunakan metode *Convolutional Neural Network* (CNN) dan *K-Nearest Neighbor* (KNN) untuk membuat model klasifikasi yang dapat memprediksi penyakit pada tanaman cabai merah. Arsitektur VGG16 digunakan dalam model CNN dan KNN untuk dapat membantu dalam ekstraksi fitur dan pengenalan pola pada citra. Hasilnya, model CNN memberikan akurasi sebesar 86%, sementara model KNN menghasilkan akurasi sebesar 81%. Evaluasi model dilakukan dengan menggunakan metode *confusion matrix*. Metode *confusion matrix* ini akan menghitung nilai *accuracy*, *precision*, *recall* dan *F1-score* dari masing-masing model. Berdasarkan perhitungan yang telah dilakukan, didapatkan nilai akurasi tertinggi berada di model CNN.

Copyright © 2025 Jurnal Teknologi Informasi UTU
All rights reserved

1. Pendahuluan

Banyak penduduk di Indonesia yang suka akan makanan pedas. Mereka sering menggunakan cabai merah sebagai salah satu bumbu pokok ataupun sayuran yang selalu hadir di hampir setiap hidangan sehari-hari [1]. Cabai merah (*Capsicum annuum L*) merupakan salah satu komoditas tanaman pertanian tertinggi jika dibandingkan dengan komoditas lainnya [2]. Namun produksi cabai merah selalu terbatas oleh faktor serangan abiotik dan biotik [3]. Serangan biotik merupakan serangan yang disebabkan oleh virus dan bakteri. Sedangkan serangan abiotik merupakan serangan yang disebabkan oleh faktor iklim atau cuaca, air, tanah, dan cahaya. Identifikasi cepat penyakit pada tanaman cabai merah menjadi kunci yang memungkinkan petani untuk mengatasi dan memantau tanaman dengan penggunaan pestisida [4].

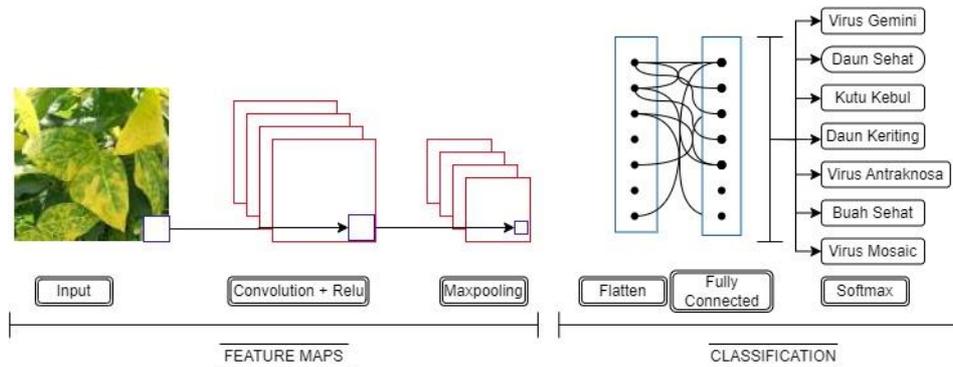
Dengan majunya teknologi, hal tersebut juga dapat diatasi dengan memanfaatkan *Computer Vision* untuk membantu dalam mengklasifikasikan penyakit yang ada pada tanaman cabai dengan menerapkan metode *Convolutional Neural Network* (CNN) dan *K-Nearest Neighbor* (KNN) [5]. CNN merupakan algoritma jaringan saraf tiruan untuk memproses data yang masuk melalui beberapa lapisan [6]. Dimana lapisan tersebut ialah lapisan konvolusi, lapisan *pooling*, dan lapisan *fully-connected*. KNN sendiri merupakan sebuah metode yang ada di *Machine Learning* dimana metode ini lebih fleksibel dan tidak

perlu membangun model yang terlalu kompleks, serta metode ini juga dapat menangani data yang memiliki atribut beragam dan dapat dengan baik mengatasi permasalahan *non-linear* dalam klasifikasi.

2. Metodologi Penelitian

2.1. Metode *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah sebuah struktur jaringan yang terdiri dari beberapa lapisan yang bertujuan untuk dapat mengidentifikasi fitur secara otomatis dalam data masukan, terutama citra [7]. Cara kerja CNN adalah dengan meniru jaringan otak saraf manusia [8]. Gambar 1. berikut merupakan ilustrasi arsitektur CNN.



Gambar 1. Ilustrasi Proses *Convolutional Neural Network*

Didalam CNN sendiri, terjadi perhitungan disetiap tahapannya menggunakan rumus-rumus yang berlaku, seperti berikut:

a. *Convolutional layer*

$$(I * K)(i, j) = \sum_{m, n} I(i + m, j + n) \cdot K(m, n) \quad (1)$$

Dimana:

I adalah input (gambar).

K adalah filter (kernel).

i, j adalah koordinat dari *output feature map*.

m, n adalah koordinat dari filter.

b. *Activation Function* (ReLU)

$$f(x) = \max(0, x) \quad (2)$$

Dimana:

x adalah *output* dari lapisan konvolusi.

c. *Pooling layer*

$$P(i, j) = \max_{0 \leq m < p} \max_{0 \leq n < p} I(i + m, j + n) \quad (3)$$

Dimana:

P adalah hasil dari *pooling*.

p adalah ukuran dari *pooling window*.

d. *Fully connected layer*

$$y = W \cdot x + b \quad (4)$$

Dimana:

y adalah *output*.

W adalah matriks bobot.

x adalah *input*.

b adalah bias.

e. *Softmax layer*

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad (5)$$

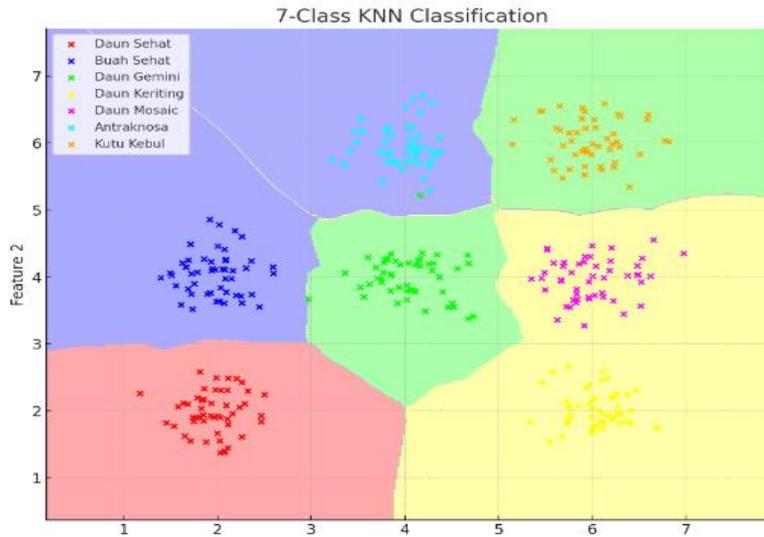
Dimana:

$\sigma(z)_i$ adalah probabilitas dari kelas i .

K adalah jumlah kelas.

2.2. Metode *K-Nearest Neighbor*

Algoritma *K-Nearest Neighbor* (KNN) merupakan teknik dalam bidang ilmu komputer yang bertujuan untuk mengklasifikasikan objek-objek berdasarkan pada data pembelajaran yang memiliki jarak terdekat dengan objek yang akan diklasifikasikan [9], [10]. Algoritma ini mengoperasikan prinsip dasarnya dengan menggunakan konsep jarak terpendek dari sampel uji ke sampel latih sebagai landasan untuk menentukan nilai KNN-nya, yang memungkinkan untuk menentukan klasifikasi yang optimal berdasarkan pada data yang tersedia. Setelah dihasilkan nilai KNN, kemudian diambil mayoritas dari KNN untuk dijadikan prediksi dari sampel uji. Penentuan jarak K pada metode KNN, menggunakan metrik jarak *euclidian*, *minkowski*, dan *manhattan* [11]. Gambar 2 berikut merupakan ilustrasi sebaran data pada metode KNN.



Gambar 2. Ilustrasi Sebaran Data Dengan KNN

Pada metode KNN terdapat perhitungan jarak dalam klasifikasi citranya, dimana rumus-rumus yang berlaku tersebut, dapat dilihat persamaan dibawah ini:

a. *Euclidean distance*:

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \tag{6}$$

b. *Minkowski distance*:

$$d(p, q) = \sqrt[m]{\sum_{i=1}^n |p_i - q_i|^m} \tag{7}$$

c. *Manhattan distance*:

$$d(p, q) = \sum_{i=1}^n |p_i - q_i| \tag{8}$$

Keterangan:

- p dan q adalah dua piksel dalam gambar.
- p_1 dan q_1 adalah intensitas piksel pada koordinat i (misalnya, nilai RGB).
- $d(p, q)$ adalah mengukur jarak dari metrik (*euclidian*, *minkowski*, dan *manhattan*).
- m adalah parameter yang menentukan jenis jarak.

2.3. Metode *Confusion Matrix*

Metode *confusion matrix* adalah sebuah teknik yang menggunakan tabel untuk menyajikan jumlah data uji yang diklasifikasikan dengan benar dan jumlah data uji yang diklasifikasikan dengan salah [12]. Metode ini digunakan untuk mengevaluasi model yang akan dihasilkan nantinya. Dapat dilihat pada Tabel 1 yang merupakan tabel dari metode *confusion matrix*.

Tabel 1. Tabel Metode *Confusion Matrix*

K	-	Kelas Prediksi
-----	---	----------------

		Positif	Negatif
	Positif	TP	FP
	Negatif	FN	TN

Keterangan : TP (*true positive*) : Jumlah data positif yang benar.
 FP (*false positive*) : Jumlah data negatif yang benar.
 FN (*false negative*) : Jumlah data negatif yang salah.
 TN (*true negative*) : Jumlah data negatif yang salah.

Berikut adalah rumus *confusion matrix* untuk menghitung nilai akurasi, *precision*, *F1-score*, dan *recall*:

$$a. Accuracy = \frac{\text{Jumlah prediksi benar}}{\text{Total jumlah prediksi}} \quad (9)$$

$$b. Precision = \frac{TP}{(FP+TP)} \quad (10)$$

$$c. Recall = \frac{TP}{(FN+TP)} \quad (11)$$

$$d. F1-Score = \frac{(2 \times recall \times precision)}{(recall + precision)} \quad (12)$$

Berikut adalah tabel *confusion matrix* untuk setiap kelas yang digunakan dalam penelitian. Dimana seperti yang terlihat di Tabel 2 ilustrasi setiap kelas yang ada di *confusion matrix*.

Tabel 2. *Confusion Matrix* untuk 7 Kelas

Kelas Sebenarnya/Prediksi	Buah Sakit	Buah Sehat	Daun Gemini	Daun Keriting	Daun Mosaic	Daun Sehat	Kutu Kebul
Buah Sakit	TP	FP	FP	FP	FP	FP	FP
Buah Sehat	FP	TP	FP	FP	FP	FP	FP
Daun Gemini	FP	FP	TP	FP	FP	FP	FP
Daun Keriting	FP	FP	FP	TP	FP	FP	FP
Daun Mosaic	FP	FP	FP	FP	TP	FP	FP
Daun Sehat	FP	FP	FP	FP	FP	TP	
Kutu Kebul	FP	FP	FP	FP	FP	FP	TP

2.4. Segmentasi Warna RGB (*Red, Green, Blue*)

Segmentasi warna memanfaatkan berbagai model warna, salah satu yang paling umum adalah model RGB (*Red, Green, Blue*) [13]. Selain RGB, terdapat pula model normalisasi RGB yang merepresentasikan setiap piksel dalam citra digital dengan tiga komponen: r, g, dan b [14]. Model normalisasi RGB menghitung nilai-nilai r, g, dan b berdasarkan persamaan-persamaan tertentu. Persamaan-persamaan ini menentukan bagaimana nilai intensitas warna merah, hijau, dan biru dihitung untuk setiap piksel. Persamaan-persamaan untuk menghitung nilai r, g, dan b dapat dilihat di bawah ini:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (13)$$

2.5 Data Penelitian

Pada penelitian ini menggunakan data citra sebanyak 3.445 citra dari tanaman cabai merah. Data diambil langsung dari kebun pertanian cabai merah, serta juga diambil melalui media *open source* Kaggle.com. Dimana dari data tersebut akan dibagi menjadi data training dan data validasi. Pembagian data tersebut akan digunakan dalam pelatihan model CNN dan KNN, dengan proforsi pembagian datanya menjadi 80% untuk data training dan 20% untuk data validasi atau data testing. Pada Tabel 3 berikut, akan terlihat sebaran data yang digunakan dalam penelitian ini beserta kelas dari data tersebut.

Tabel 3. Sebaran Dataset

No.	Class	Jumlah Data	Data Training	Data Validasi	Data Primer	Data Sekunder
1	Daun Sehat	1.032	2.764	691	550	2.895
2	Buah Sakit	612				
3	Daun Mosaic	600				
4	Buah Sehat	500				
5	Daun Keriting	467				
6	Daun Gemini	144				
7	Kutu Kebul	100				
Total		3.455				

Dari Tabel 3 diatas, data pada setiap kelasnya berbeda-beda. Dimana data terbanyak dimiliki oleh *class* daun sehat dengan sebanyak 1.032 data citra, dan data tersedikit dimiliki oleh *class* kutu kebul dengan sebanyak 100 data citra saja. Dengan total keseluruhan *dataset* sebanyak 3.445 citra, akan dibagi 80% dari total keseluruhan data untuk menjadi data training dengan total data sebanyak 2.764 citra, dan 20% dari keseluruhan total data akan dibagi menjadi data testing atau validasi dengan jumlah data 691 citra.

3. Hasil dan Pembahasan

3.1 Hasil Metode Convolutional Neural Network (CNN)

Model CNN yang telah selesai dibangun, akan dilihat nilai *accuracy*-nya sebagai tolak ukur atau ketepatan model dalam memprediksi gambar baru. Nilai dari *accuracy* ini sangat penting untuk menunjukkan seberapa baik model dapat mengenali pola dan fitur baru dalam data testing yang kemudian akan diterapkan pada data uji atau data baru. Selain itu, nilai *accuracy* yang tinggi juga dapat mengindikasikan bahwa model dapat melakukan generalisasi secara baik pada gambar-gambar baru yang belum pernah dilihat sebelumnya. Dapat dilihat pada Tabel 4 hasil dari model CNN yang telah dibangun.

Tabel 4. Hasil dari Model CNN

No.	Model	Class	<i>precision</i>	<i>recall</i>	<i>F1-score</i>	<i>accuracy</i>
1	CNN	Buah Sakit	0.98	0.96	0.97	0.86
2		Buah Sehat	0.90	0.83	0.87	
3		Daun Gemini	0.60	0.45	0.51	
4		Daun Keriting	0.71	0.82	0.76	
5		Daun Mosaic	0.86	0.86	0.86	
6		Daun Sehat	0.86	0.90	0.88	
7		Kutu Kebul	0.80	0.55	0.65	

Dari Tabel 4 diatas, didapatkan hasil *accuracy* sebesar 0.86 jika dibulatkan menjadi 86%. Nilai dari masing-masing metrik diambil dari hasil *classification report* yang ada diprogram model CNN, dengan mengambil *history* model pada saat proses training.

3.2 Hasil Metode K-Nearest Neighbor (KNN)

Model *k-nearest neighbor* (KNN) yang telah selesai di bangun, akan dilihat atau dievaluasi terlebih dahulu untuk melihat nilai *accuracy* yang dihasilkan sudah sesuai dengan yang diinginkan atau tidak. Pada model KNN berbeda dengan CNN, dimana harus terlebih dahulu menentukan nilai jarak antar data atau nilai K. Nilai jarak yang digunakan adalah 3, dengan hasil yang dapat dilihat pada Tabel 5 hasil dari model KNN.

Tabel 5. Hasil dari Model KNN

No.	Model	Class	<i>precision</i>	<i>recall</i>	<i>F1-score</i>	<i>accuracy</i>
-----	-------	-------	------------------	---------------	-----------------	-----------------

1	KNN	Buah Sakit	0.91	0.92	0.92	0.81
2		Buah Sehat	0.86	0.78	0.82	
3		Daun Gemini	0.32	0.40	0.36	
4		Daun Keriting	0.67	0.66	0.66	
5		Daun <i>Mosaic</i>	0.95	0.76	0.84	
6		Daun Sehat	0.80	0.92	0.85	
7		Kutu Kebul	0.71	0.68	0.70	

Berdasarkan Tabel 5 diatas, dimana nilai *accuracy* yang dihasilkan oleh model KNN sebesar 0.81 atau sebesar 81%. Nilai masing-masing metrik juga diambil dari *classification report* berdasarkan *history* training model KNN.

3.3 Hasil Uji Model CNN Terhadap Objek Baru

Pengujian pada model CNN dilakukan dengan beberapa kali proses pengujian. Pengujian dilakukan untuk melihat seberapa akuratnya hasil prediksi yang diberikan model terhadap data baru. Uji terhadap objek baru dilakukan sebanyak 5 kali pengujian, dimana setiap proses pengujiannya memiliki 10 citra baru dari tanaman cabai merah. Secara keseluruhan, pengujian yang dilakukan dengan 5 kali uji memiliki total 50 gambar atau citra baru. Dapat dilihat dari Tabel 6 hasil uji model CNN terhadap objek baru.

Tabel 6. Hasil Uji Model CNN Terhadap Data Baru

No.	Model	Prediksi Benar	Prediksi Salah	Total Prediksi Benar	Total Prediksi Salah
1	CNN	6	4	35	15
2		7	3		
3		8	2		
4		6	4		
5		8	2		

Pada tabel 6 diatas, model CNN dilakukan pengujian terhadap citra baru sebanyak 5 kali pengujian. Dimana dari pengujian tersebut, model CNN mampu memprediksi citra baru dengan benar sebanyak 35 citra dari total 50 citra. Serta memprediksi dengan salah sebanyak 15 citra dari total 50 citra.

3.4 Hasil Uji Model KNN Terhadap Objek Baru

Pengujian model KNN terhadap data baru perlu dan penting untuk dilakukan. Hal tersebut berguna untuk melihat seberapa bagus model yang telah selesai di bangun dalam memprediksi citra tanaman cabai dengan benar. Selain itu, juga sebagai media evaluasi terhadap model jika hasil prediksi masih banyak yang salah. Data uji yang digunakan sama dengan pengujian yang ada di model CNN, dimana menggunakan 50 citra baru yang di bagi menjadi 5 kali pengujian. Hasil dari model KNN dalam memprediksi data baru, dapat dilihat pada Tabel 7 berikut.

Tabel 7. Hasil Uji Model KNN Terhadap Data Baru

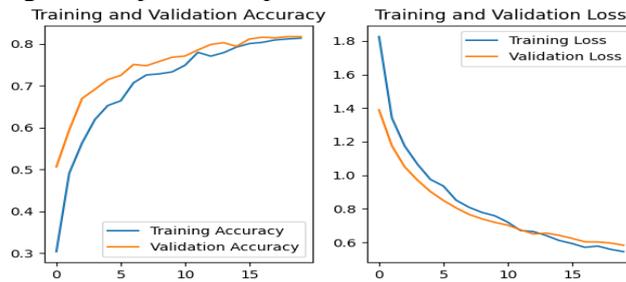
No.	Model	Prediksi Benar	Prediksi Salah	Total Prediksi Benar	Total Prediksi Salah
1	KNN	6	4	25	25
2		3	7		
3		5	5		
4		5	5		
5		6	4		

Berdasarkan Tabel 7 diatas, model KNN dilakukan pengujian terhadap data baru sebanyak 5 sesi. Setiap sesinya memiliki citra baru sebanyak 10 citra. Model KNN mampu memprediksi citra dengan

benar sebanyak 25 citra, dan memprediksi citra baru dengan salah sebanyak 25 citra dari total keseluruhan citra baru sebanyak 50 citra.

3.5 Evaluasi Training Model CNN

Pada saat proses training, *history* training akan disimpan guna melakukan tahapan evaluasi selama berjalannya proses training, atau dapat dilihat pada Gambar 3 dibawah ini.



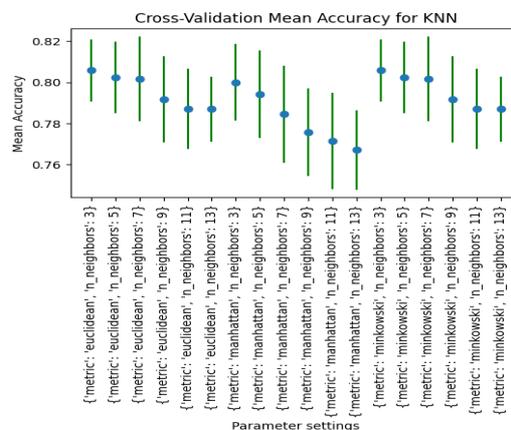
Gambar 3. Akurasi Training & Validasi, Training & Validasi Loss

Berdasarkan Gambar 3 diatas, gambar pada bagian sebelah kiri menunjukkan grafik perubahan akurasi selama training. Sumbu Y menunjukkan akurasi (nilai antara 0 dan 1). Sumbu X menunjukkan jumlah *epoch* (jumlah iterasi pelatihan model). Garis biru pada data pelatihan menunjukkan seberapa baik model dalam memprediksi label yang benar pada pelatihan di setiap *epoch*. Garis oranye akurasi pada data validasi, menunjukkan seberapa baik model memprediksi label yang benar pada data yang tidak terlihat selama pelatihan (data validasi) untuk setiap *epoch*. Dapat disimpulkan bahwa berdasarkan kedua garis tersebut, dengan seiring bertambahnya *epoch* menunjukkan model belajar dan meningkatkan kemampuannya dalam mengklasifikasikan data. Selain itu, model juga tidak terlalu mengalami *overfitting*.

Pada grafik disebelah kanan menunjukkan *loss* selama pelatihan. Dimana sumbu Y menunjukkan nilai *loss* (nilai antara 0 dan lebih besar). Sumbu X menunjukkan jumlah *epoch*. Garis biru pada data pelatihan menunjukkan seberapa besar kesalahan model dalam memprediksi label pada data pelatihan untuk setiap *epoch*. Garis oranye pada data validasi, menunjukkan seberapa besar kesalahan model dalam memprediksi label pada data yang tidak terlihat selama pelatihan (data validasi) untuk setiap *epoch*. Dapat disimpulkan penurunan *loss* baik pada data pelatihan maupun data validasi, menurun seiring bertambahnya *epoch*. Menunjukkan bahwa model semakin baik dalam memprediksi label yang benar dan kesalahannya berkurang. Kedua garis juga cenderung mendekati nilai yang sama setelah sejumlah *epoch*, yang menunjukkan bahwa model mengelompokan dengan baik dan tidak *overfitting*.

3.6 Evaluasi Training Model KNN

Evaluasi model KNN dilakukan dengan melihat nilai *mean* (rata-rata) hasil dari *cross-validation* dengan berbagai kondisi parameter. Dapat dilihat dari Gambar 4 berikut grafik *cross-validation* dari model KNN.



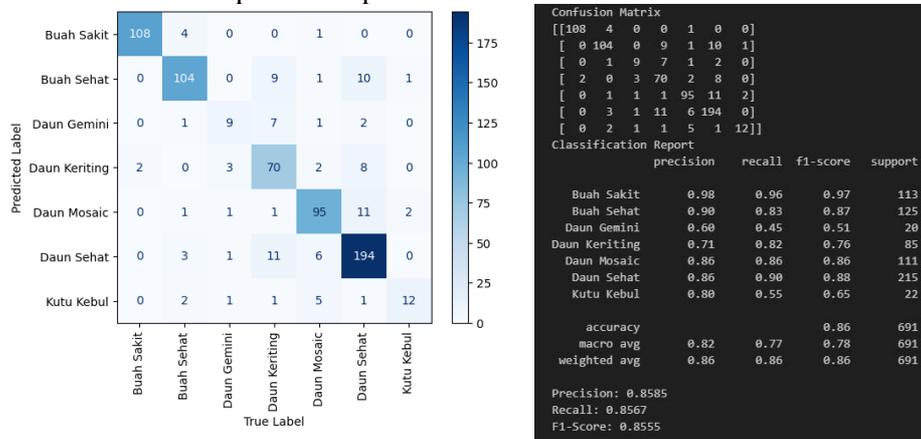
Gambar 4. Grafik Cross Validation Model KNN

Dari gambar 4 diatas, pada sumbu X terdapat berbagai pengaturan parameter yang diterapkan dalam model KNN. Sementara sumbu Y terdapat nilai *mean* atau rata-rata dari *accuracy*. Pada sumbu X sendiri, digunakan untuk mengatur metrik atau jarak yang digunakan seperti nilai jarak *euclidian*, *manhatan*, dan *minkowski*. Selain itu, *n_neighbors* adalah nilai K atau jumlah tetangga terdekat pada model KNN seperti 3, 5, 7, 9, 11, dan 13.

Sementara itu, sumbu Y merupakan rata-rata nilai dari *cross-validation* untuk setiap parameter. *Mean accuracy* ini adalah rata-rata dari akurasi yang diperoleh dari setiap lipatan (*fold*) dalam *cross-validation*. Garis *vertical* (*error bars*) yang menyertai di setiap titik data, menunjukkan rentang variasi seperti (*standard deviation* atau *standard error*), dari akurasi selama *cross-validation*. Semakin panjang garis, semakin besar pula variasi akurasi diantara lipatan. Titik biru merupakan rata-rata akurasi dari *cross-validation*. Untuk memilih parameter terbaik atau optimal, harus dicari kombinasi yang memberikan *mean accuracy* tertinggi dengan *error bars* yang relative kecil. Hal tersebut menunjukkan bahwa model KNN tidak hanya akurat tetapi juga konsisten di berbagai lipatan *cross-validation*.

3.7 Evaluasi Confusion Matrix Pada Model CNN

Evaluasi model CNN dengan menggunakan metode *confusion matrix* dalam memprediksi objek, sangat diperlukan untuk dapat mengetahui nilai dari *accuracy*, *precision*, *recall*, dan *F1-score*. Hasil evaluasi model CNN dapat dilihat pada Gambar 5 berikut.

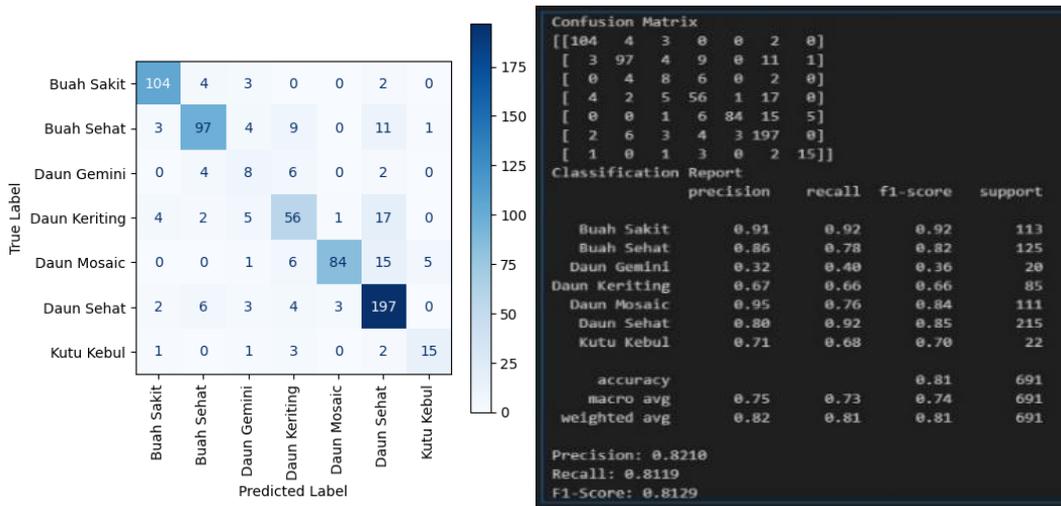


Gambar 5. Evaluasi Confusion Matrix & Classification Report CNN

Berdasarkan Gambar 5 diatas, *confusion matrix* menunjukkan nilai prediksi benar dan salah terhadap setiap kelas dari *dataset* yang digunakan. Setiap baris mewakili label sebenarnya, sedangkan kolom mewakili label yang diprediksi. Sedangkan *classification report* memberikan metrik evaluasi untuk setiap kelas, termasuk *precision*, *recall*, *f1-score*, dan *accuracy*. Hasil perhitungan dengan menggunakan metode *classification report* menghasilkan nilai akurasi berupa 86%. Nilai rata-rata dari *precision* sebesar 0.8585 (85.85%), nilai rata-rata dari *recall* sebesar 0.8567 (85.67%), dan nilai rata-rata dari *F1-score* sebesar 0.8555 (85.55%). Nilai rata-rata tersebut di dapat dari hasil perhitungan yang ada di *classification report*.

3.8 Evaluasi Confusion Matrix Pada Model KNN

Evaluasi model KNN dilakukan mempredikasi objek baru dengan menggunakan metode *confusion matrix*, untuk melihat nilai dari *accuracy*, *precision*, *recall*, dan *F1-score* dari setiap kelasnya. Hasil dari *confusion matrix* dan *classification report* dari model KNN dapat dilihat pada Gambar 6 berikut.



Gambar 6. Evaluasi *Confusion Matrix* & *Classification Report* KNN

Dari Gambar 6 diatas, nilai *confusion matrix* menunjukkan nilai prediksi yang benar dan salah dari prediksi gambar. Baris pada *confusion matrix* mewakili nilai sebenarnya, sedangkan kolom pada *confusion matrix* mewakili label yang diprediksi. *Classification report* digunakan untuk melihat nilai *accuracy*, *precision*, *recall*, dan *F1-score* dari setiap kelas. Dengan melihat hasil perhitungan yang berlaku pada metode *confusion matrix*, pada model KNN menghasilkan akurasi sebesar 81%. Nilai rata-rata dari *precision* sebesar 0.8210 (82.10%), nilai rata-rata *recall* sebesar 0.8119 (81.19%), dan nilai rata-rata dari *F1-score* sebesar 0.8129 (81.29%). Nilai rata-rata tersebut, didapat dari tabel *classification report* yang ada di Gambar 6.

3.9 Perbandingan Hasil Model CNN & KNN

Berdasarkan hasil perhitungan nilai *confusion matrix* untuk masing-masing model pada tahapan sebelumnya, didapatkan nilai akurasi terbaik berada pada model CNN. Dimana model CNN menghasilkan akurasi sebesar 0.86 atau 86%. Sedangkan model KNN menghasilkan akurasi sebesar 0.81 atau 81%. Perhatikan Tabel 8. perbandingan antara model CNN dan KNN berikut.

Tabel 8. Perbandingan Hasil Perhitungan *Confusion Matrix*

No.	Model	Class	precision	recall	F1-score	accuracy
1	CNN	Buah Sakit	0.98	0.96	0.97	0.86
2		Buah Sehat	0.90	0.83	0.87	
3		Daun Gemini	0.60	0.45	0.51	
4		Daun Keriting	0.71	0.82	0.76	
5		Daun Mosaic	0.86	0.86	0.86	
6		Daun Sehat	0.86	0.90	0.88	
7		Kutu Kebul	0.80	0.55	0.65	
1	KNN	Buah Sakit	0.91	0.92	0.92	0.81
2		Buah Sehat	0.86	0.78	0.82	
3		Daun Gemini	0.32	0.40	0.36	
4		Daun Keriting	0.67	0.66	0.66	
5		Daun Mosaic	0.95	0.76	0.84	
6		Daun Sehat	0.80	0.92	0.85	
7		Kutu Kebul	0.71	0.68	0.70	

Dari tabel 8 tersebut, dapat dilihat bahwa akurasi tertinggi dihasilkan pada model CNN. Setiap model dihitung masing-masing kelas untuk mengetahui nilai dari masing-masing metriknya.

Selanjutnya, perbandingan hasil prediksi antara model CNN dan KNN terhadap citra baru. Seperti yang terlihat pada Tabel 9 perbandingan hasil prediksi terhadap objek baru seperti dibawah ini.

Tabel 9. Perbandingan Hasil Uji Terhadap Objek Baru Pada Model CNN & KNN

No.	Model	Prediksi Benar	Prediksi Salah	Total Prediksi Benar	Total Prediksi Salah
1	CNN	6	4	35	15
2		7	3		
3		8	2		
4		6	4		
5		8	2		
1	KNN	6	4	25	25
2		3	7		
3		5	5		
4		5	5		
5		6	4		

Berdasarkan Tabel 9 diatas, model CNN berhasil memprediksi citra dengan benar sebanyak 35 citra, dan memprediksi citra dengan salah sebanyak 15 citra. Sedangkan model KNN berhasil memprediksi citra dengan benar sebanyak 25 citra, dan memprediksi dengan salah sebanyak 25 citra. Setiap model dilakukan pada data pengujian yang sama. Pengujian dilakukan sebanyak 5 kali, dengan setiap sesi pengujiannya memiliki citra sebanyak 10 citra, dengan total keseluruhan sebanyak 50 citra.

Dengan melihat tabel perbandingan hasil perhitungan *confusion matrix*, dan tabel perbandingan pengujian terhadap objek baru, penelitian telah berhasil dalam menganalisa penyakit yang ada di tanaman cabai merah. Dimana akurasi tertinggi diperoleh oleh model CNN sebesar 86%, sedangkan model KNN berhasil memperoleh akurasi sebesar 81%.

4. Kesimpulan

Berdasarkan hasil penelitian dalam “Analisa Penyakit Pada Tanaman Cabai Merah (*Capsicum annum L*) Dengan Membandingkan Tingkat Akurasi Menggunakan Metode *Convolutional Neural Network* (CNN) Dan *K-Nearest Neighbor* (KNN)” dapat disimpulkan bahwa, model CNN dan KNN telah berhasil dibuat dengan tingkat akurasi yang cukup memuaskan. Dimana model CNN mampu menghasilkan akurasi sebesar 86%, sedangkan model KNN menghasilkan tingkat akurasi sebesar 81%. Selanjutnya, masing-masing model dibangun dengan menambahkan arsitektur VGG16 dalam pembuatannya. VGG16 di tambahkan agar model dapat lebih mengenal secara mendalam dari data. Hasil tingkat akurasi pada model, dibandingkan dengan menggunakan metode *confusion matrix*. Dimana pada metode tersebut dilihat persentasi dari *accuracy*, *F1-score*, *recall*, dan *precision*.

Daftar Pustaka

- [1] I. Verawati and A. Lucky Rebecca, ‘Klasterisasi Jenis Tanah pada Tanaman Cabai Menggunakan Algoritma K-Means’, vol. 7, no. 3, pp. 1414–1422, 2023, doi: 10.30865/mib.v7i3.6132.
- [2] M. Anwar, Y. Kristian, and E. Setyati, ‘Classification Of Chili Plant Diseases Equipped With Leaf And Fruit Image Segmentation Using Yolo V7’, *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 6, no. 1, 2023.
- [3] R. Kelikualiq, N. B. A. Karna, and B. Aditya, ‘Sistem Monitoring Kesehatan Tanaman Cabai Pada Smart Greenbox Menggunakan Algoritma *Convolutional Neural Network* (CNN) Berbasis *Internet Of Things (IOT) Dan Website*’, *eProceedings of Engineering*, vol. 9, no. 6, Jan. 2023, Accessed: Feb. 16, 2024.
- [4] L. Setiana Riva, ‘Deteksi Penyakit Tanaman Cabai Menggunakan Algoritma YOLOv5 Dengan Variasi Pembagian Data’, vol. 8, no. 3, 2023.
- [5] S. Taliki, J. Nur, and I. Colanus Rally Drajana, ‘Aplikasi Diagnosa Penyakit Tanaman Cabai Merah menggunakan Algoritma *K-Nearest Neighbor*’, 2022.
- [6] I. E. Handayani and D. Avianto, ‘Klasifikasi Penyakit Antraknosa Pada Cabai Merah Teropong “*Inko Hor*” Dengan Metode *Convolutional Neural Network*’, 2023, [Online]. Available: <https://doi.org/10.31598>

- [7] F. Ilham and N. Rochmawati, 'Transliterasi Aksara Jawa Tulisan Tangan ke Tulisan Latin Menggunakan CNN', *Journal of Informatics and Computer Science*, vol. 01, no. 04, 2020.
- [8] Y. A. Suwitono and F. J. Kaunang, 'Implementasi Algoritma *Convolutional Neural Network* (CNN) Untuk Klasifikasi Daun Dengan Metode Data Mining SEMMA Menggunakan Keras', *Jurnal Komtika (Komputasi dan Informatika)*, vol. 6, no. 2, pp. 109–121, Nov. 2022, doi: 10.31603/komtika.v6i2.8054.
- [9] S. Raysyah, V. Arinal, and D. I. Mulyana, 'Klasifikasi Tingkat Kematangan Buah Kopi Berdasarkan Deteksi Warna Menggunakan Metode Knn Dan Pca', *JSiI (Jurnal Sistem Informasi)*, vol. 8, no. 2, pp. 88–95, Sep. 2021, doi: 10.30656/JSiI.V8I2.3638.
- [10] R. L. Hasanah *et al.*, 'Klasifikasi Penerima Dana Bantuan Desa Menggunakan Metode Knn (*K-Nearest Neighbor*)', *Jurnal Techno Nusa Mandiri*, vol. 16, no. 1, pp. 1–6, Mar. 2019, doi: 10.33480/TECHNO.V16I1.25.
- [11] A. Tangkelayuk and E. Mailoa, 'Klasifikasi Kualitas Air Menggunakan Metode KNN, *Naïve Bayes*, dan *Decision Tree*', *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 2, pp. 1109–1119, Jun. 2022, doi: 10.35957/JATISI.V9I2.2048.
- [12] D. Normawati and S. A. Prayogi, 'Implementasi *Naïve Bayes Classifier* Dan *Confusion Matrix* Pada Analisis Sentimen Berbasis Teks Pada Twitter', *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 5, no. 2, pp. 697–711, Sep. 2021, Accessed: Jul. 15, 2024. [Online]. Available: <https://ejournal.tunasbangsa.ac.id/index.php/jsakti/article/view/369>
- [13] I. Wulandari, H. Yasin, and T. Widiharah, 'Klasifikasi Citra Digital Bumbu Dan Rempah Dengan Algoritma *Convolutional Neural Network* (Cnn)', *Jurnal Gaussian*, vol. 9, no. 3, pp. 273–282, Aug. 2020, doi: 10.14710/J.GAUSS.9.3.273-282.
- [14] I. Perlindungan, 'Pengenalan Tanaman Cabai Dengan Teknik Klasifikasi Menggunakan Metode Cnn. 2020.