KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)
Terakreditasi Nomor 204/E/KPT/2022 | Vol. 6, No. 1, Januari (2025), pp. 38-47

Automated Detection of Black Pod Disease in Cocoa Fruits Using Convolutional Neural Network

Febbi Sena Lestari[1], Harliana[2], Fatra Nonggala Putra[3]*
*[1,*2,3]Universitas Nahdlatul Ulama Blitar, Indonesia*
E-mail: febbisena.l@gmail.com[1], harliana@unublitar.ac.id[2], fatranp@unublitar.ac.id[3]

Abstract

Black pod disease is a severe disease affecting cocoa fruit, caused by the Phytophthora Palmivora fungus. This infection turns the fruit's surface dark brown to black, while the inside becomes rotten. Currently, identifying infected cocoa fruits is done manually through visual observation, which is prone to errors and inconsistency. This study aims to implement a Convolutional Neural Network (CNN) algorithm to classify images of black pod disease in cocoa fruits. The dataset consists of 1,500 images obtained through documentation and literature review, with 750 images of healthy cocoa fruits and 750 images of infected fruits. To determine the optimal configuration, the CNN model was tested across 15 scenarios with varying batch sizes and epochs. The results show that the fifth scenario, with a batch size of 32 and 50 epochs, achieved the best performance, with an accuracy of 97.33%, precision of 97.41%, recall of 97.33%, and an f1-score of 97.33%. Additionally, the model was further tested using 20 original images, achieving an accuracy of 90%. These results demonstrate that the CNN model developed effectively classifies cocoa fruit images affected by black pod disease, highlighting its potential for use in developing more accurate and efficient cocoa disease detection applications.

Keywords: *Black Pod Disease, Cocoa Pod, Convolutional Neural Network (CNN).*

Abstrak

Black pod disease adalah salah satu penyakit serius yang menyerang buah kakao, disebabkan oleh jamur Phytophthora Palmivora. Infeksi jamur ini membuat permukaan buah kakao berubah warna menjadi hitam kecokelatan, dan bagian dalam buah menjadi busuk. Saat ini, identifikasi buah kakao yang terinfeksi masih dilakukan secara manual melalui pengamatan visual, yang rentan terhadap kesalahan dan ketidakkonsistenan. Penelitian ini bertujuan untuk mengimplementasikan algoritma Convolutional Neural Network (CNN) dalam klasifikasi citra penyakit black pod pada buah kakao. Dataset yang digunakan terdiri dari 1.500 citra yang diperoleh melalui dokumentasi dan studi literatur, dengan 750 citra buah kakao sehat (Healthy) dan 750 citra buah kakao terinfeksi (Black Pod). Model CNN diuji menggunakan 15 skenario berbeda dengan variasi nilai batch size dan epoch untuk mencari konfigurasi terbaik. Hasil pengujian menunjukkan bahwa skenario ke-5, dengan batch size 32 dan epoch 50, memberikan performa terbaik, dengan akurasi mencapai 97,33%, presisi 97,41%, recall 97,33%, dan f1-score 97,33%. Selain itu, model diuji lebih lanjut menggunakan 20 citra asli dan menghasilkan akurasi sebesar 90%. Dari hasil ini, model CNN yang dikembangkan menunjukkan kemampuan yang baik dalam mengklasifikasikan citra buah kakao yang terinfeksi black pod disease, sehingga model ini memiliki potensi untuk digunakan dalam pengembangan aplikasi deteksi penyakit kakao yang lebih akurat dan efisien.

Keywords: *Black Pod Disease, Buah Kakao, Convolutional Neural Network (CNN).*

1. Pendahuluan

Kakao (*Theobroma cacao L*) merupakan tanaman penting dari family *Sterculiaceae*, yang sering dimanfaatkan sebagai bahan baku produk makanan, obat-obatan, dan

ISSN: 2720-992X
Copyright © KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)

kosmetik. Di Indonesia, kakao termasuk komoditas perkebunan yang signifikan setelah kelapa sawit, karet, dan kopi, dengan nilai ekonomi yang cukup tinggi[1]. Namun, data dari BPS RI pada tahun 2022 menunjukkan bahwa produksi kakao nasional mengalami penurunan sebesar 0,6% atau sekitar 60,9 ribu ton selama periode 2018 hingga 2021. Salah satu penyebab penurunan ini adalah serangan penyakit tanaman, khususnya *black pod disease* atau penyakit busuk buah, yang berdampak besar pada produktivitas tanaman kakao. *Black pod disease* merupakan salah satu penyakit paling merusak bagi kakao, yang disebabkan oleh jamur *Phytophthora Palmivora*. Infeksi ini menyebabkan permukaan buah kakao berubah menjadi hitam kecoklatan, dan dalam waktu singkat isi buah akan membusuk[2][3]. Penyakit ini memiliki tingkat penyebaran yang cepat dan dapat menyebabkan kerugian langsung hingga 30-50% dari hasil panen[4]. Jika tidak segera diatasi, infeksi *black pod disease* dapat menjalar ke tangkai dan kulit batang, berkembang menjadi kanker batang yang mampu menginfeksi seluruh bagian tanaman kakao dan menyebabkan kematian tanaman[2].

Di Indonesia proses deteksi penyakit *black pod disease* masih dilakukan secara manual oleh petani melalui pengamatan visual. Metode ini mengandalkan penilaian subjektif yang rentan terhadap ketidakkonsistenan karena perbedaan kualitas penglihatan, perspektif, kondisi fisik petani, dan kondisi lingkungan saat penilaian dilakukan. Proses manual ini sering kali menyebabkan deteksi terlambat, sehingga penyakit dapat menyebar dan mengakibatkan kerugian yang lebih besar. Oleh karena itu, diperlukan solusi otomatis yang dapat mendeteksi penyakit secara cepat dan akurat untuk membantu petani dalam mengambil tindakan pencegahan yang efektif.

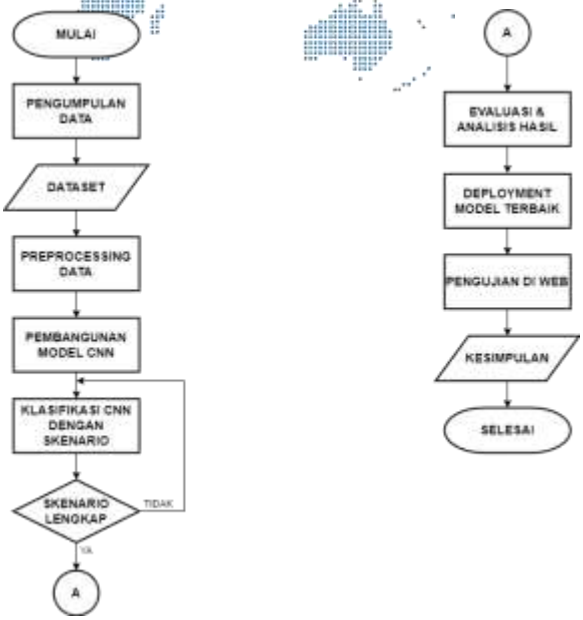
Pada penelitian sebelumnya, deteksi penyakit kakao telah diupayakan dengan berbagai pendekatan. Beberapa metode sistem pakar seperti logika fuzzy[5] yang menghasilkan akurasi 100% dengan 4 pengujian dari 4 dataset penyakit, *Certainty Factor* [6] terhadap 2 gejala penyakit, dan *Dempster Shafer* [7] untuk 10 penyakit dan 21 gejala untuk mendeteksi penyakit ini. Pada ranah *machine learning* metode *K-Nearest Neighbor* (KNN)[8] dan *Support Vector Machine* [9] juga pernah diaplikasikan untuk klasifikasi penyakit kakao. Meskipun metode-metode tersebut mampu mencapai hasil yang cukup baik, seperti akurasi SVM sebesar 89%, performa model masih sangat dipengaruhi oleh pemilihan parameter dan cenderung kurang optimal untuk klasifikasi citra dengan detail visual yang kompleks, seperti yang dimiliki oleh citra buah kakao yang terinfeksi.

Berbeda dengan beberapa penelitian tersebut, pada penelitian ini peneliti akan menggunakan Algoritma *Convolutional Neural Network* (CNN) untuk klasifikasi citra penyakit *black pod disease* pada buah kakao. CNN dipilih karena kemampuannya yang sangat baik dalam mengenali objek melalui pembelajaran berbasis fitur visual, yang memungkinkan model untuk mengidentifikasi pola dan tekstur pada citra secara otomatis[10]. Dengan struktur lapisan konvolusi dan pooling, CNN mampu mengekstrak fitur-fitur penting dari piksel citra secara bertahap. Selain itu, CNN mengintegrasikan proses ekstraksi fitur dan klasifikasi dalam satu model, sehingga lebih efisien dan tidak memerlukan pemisahan proses antara tahap-tahap ini[11]. Dalam melakukan klasifikasi, CNN pernah dilakukan untuk mengklasifikasi buah kiwi yang dilatih dengan 25 *epoch* dan menghasilkan nilai akurasi sebesar 0,987[12]. Ketika dibandingkan dengan SVM dan KNN dalam melakukan klasifikasi citra cuaca, akurasi yang dihasilkan oleh CNN pun cukup tinggi yaitu sekitar 0,942 dengan presisi 0,943 dan *recall* 0,942[13]. Dengan demikian, penelitian ini bertujuan untuk mengembangkan sistem deteksi penyakit *black pod disease* yang andal dan akurat menggunakan CNN, yang diharapkan dapat membantu petani dalam meningkatkan efektivitas dan konsistensi proses deteksi penyakit.

2. Metodologi Penelitian

Tahapan dalam penelitian ini terdiri dari 7 tahapan sebagaimana ditunjukkan dalam *flowchart* pada Gambar 1, yaitu pengumpulan data, *preprocessing* data, pembangunan model CNN, klasifikasi CNN dengan skenario, evaluasi dan analisis hasil, *deployment*

model terbaik, dan pengujian di web. Tahapan tersebut diharapkan dapat menghasilkan suatu model CNN terbaik dalam skenario yang selanjutnya diimplementasikan ke dalam web agar dapat dilakukan pengujian berbasis GUI (*Graphical User Interface*).



Gambar 1. *Flowchart* penelitian

2.1. Pengumpulan data

Dataset yang digunakan dalam penelitian ini peneliti ambil dari situs Kaggle yang berada pada Alamat: *https://www.kaggle.com/datasets/zaldyjr/cacao-diseases*. Jumlah dataset yang digunakan berjumlah 1.500 data citra buah kakao yang terdiri dari 750 citra buah kakao sehat (*healthy*) dan 750 citra penyakit busuk buah kakao (*black pod*). Selanjutnya citra akan disimpan ke dalam folder yang sesuai dengan nama kelasnya yaitu *healthy* untuk dataset buah kakao sehat dan *black pod* untuk dataset buah kakao berpenyakit yang ditandai dengan adanya bercak / warna kehitaman pada permukaan kulit buah. Gambar 2 adalah contoh citra dari setiap kelas.

Gambar 2. Sample citra yang akan digunakan

Karena dataset dalam penelitian ini merupakan citra buah kakao yang hampir sebagian besar masih menggantung pada pohon dan berlatarbelakang perkebunan maka akan dilakukan proses *preprocessing* untuk menghilangkan objek-objek yang tidak dibutuhkan.

2.2. Preprocessing data

Tahapan *preprocessing* data pada citra yang digunakan adalah:

a) *Cropping*, pada tahapan ini akan dihilangkan latar belakang dan objek yang tidak dibutuhkan dalam proses klasifikasi.
b) *Resize*, pada tahapan ini akan dilakukan perubahan ukuran citra menjadi 150x150 piksel agar proses klasifikasi yang dilakukan tidak terlalu berat. Meskipun ukuran resolusinya tidak terlalu besar, tetapi citra kakao masih dapat terlihat dengan jelas.
c) Augmentasi data, pada tahapan ini akan dilakukan proses *rescale* (untuk mengubah skala nilai piksel citra guna mendapatkan rentang nilai piksel citra antara 0-1), *zoom range* (untuk melakukan zoom range guna menghasilkan variasi citra dengan memperbesar/memperkecil tampilan citra secara acak tanpa mengubah ukuran resolusinya), *rotation range* (untuk menghasilkan variasi citra dengan memutar tampilan secara acak), *flip* (untuk menghasilkan cerminan dari tampilan citra, flip akan dilakukan secara horizontal sehingga objek yang sebelumnya ada di sebelah kiri akan menjadi di sebelah kanan), *shear range* (untuk menghasilkan variasi citra dengan mentransformasi tampilan secara acak), *brightness range* (untuk mengubah kecerahan citra dimana nilai yang akan digunakan yaitu 80-120% atau [0,8 1,2] karena akan menghasilkan citra yang tidak terlalu gelap ataupun terang), dan *fill* (untuk mengisi piksel citra yang kosong akibat proses *zoom range*, *rotation range* dan *shear range* pada posisi yang sama di sisi berlawanan) terhadap data latih. Gambar 3 merupakan proses augmentasi yang dilakukan.

Gambar 3. Contoh hasil augmentasi

2.3. Pembangunan model CNN

Pada tahapan ini akan ditentukan susunan lapisan terhadap model CNN yang digunakan, Adapun rangkuman mengenai lapisan ini terangkum pada Tabel 1.

Tabel 1. Susunan lapisan model CNN

Nama lapisan	Ukuran
Convolutional layer (Conv2D)	32 filter ukuran 3x3 dan aktivasi ReLu
Pooling (Maxpooling)	Filter 2x2
Convolutional layer (Conv2D)	32 filter ukuran 3x3 dan aktivasi ReLu
Pooling (Maxpooling)	Filter 2x2
Convolutional layer (Conv2D)	64 filter ukuran 3x3 dan aktivasi ReLu
Pooling (Maxpooling)	Filter 2x2
Convolutional layer (Conv2D)	128 filter ukuran 3x3 dan aktivasi ReLu
Pooling (Maxpooling)	Filter 2x2
Dropout	0,2
Flatten	-
Dense	128 neuron dan aktivasi ReLu
Dense	1 neuron dan aktivasi signoid

Pembangunan model CNN ini akan menggunakan library Keras dan Trensorflow dengan 4 lapisan konvolusi (*Convolution Layer*) berukuran 3x3, hal ini bertujuan agar jumlah parameter yang dihasilkan tidak terlalu besar karena jumlah parameter yang sesuai akan membuat pelatihan model menjadi lebih cepat dan mengurangi resiko *overfitting*.

2.4. Klasifikasi CNN

Pada tahapan ini akan dilakukan beberapa skenario untuk memperoleh model CNN terbaik dengan nilai akurasi tertinggi. Skenario yang dilakukan akan menggunakan 15 skenario pengujian dengan mengubah nilai *batch size* menjadi 3 yaitu 32, 64 dan 128 serta 5 nilai *epoch* yaitu 10, 20, 30, 40 dan 50. Adapun rangkuman mengenai 15 skenario pengujian tersebut dengan nilai *batch size* dan *epoch* berbeda terangkum pada Tabel 2.

Tabel 2. Skenario pengujian

Skenario ke-	Nilai *batch size*	Nilai *epoch*
1	32	10
2	32	20
3	32	30
4	32	40
5	32	50
6	64	10
7	64	20
8	64	30
9	64	40
10	64	50
11	128	10
12	128	20
13	128	30
14	128	40
15	128	50

2.5. Evaluasi dan analisis hasil

Pada tahapan ini akan dilakukan analisis terhadap hasil akurasi dan *loss*, *confutional matriks*, akurasi, *recall*, dan *f1-score*.

2.6. Deployment model terbaik

Pada tahapan ini akan dilakukan proses *deployment* menjadi web. Dalam penggunaannya, user akan menggunggah foto buah kakao untuk selanjutnya diprediksi masuk ke dalam kelas *Healthy* ataukah *Black pod*.

2.7. Pengujian

Pada tahapan ini akan dilakukan pengujian dengan menggunakan 20 data citra baru yang tidak termasuk dataset. Jumlah dataset tersebut adalah 10 citra untuk kelas *Healthy* dan 10 citra untuk kelas *Black pod*.

3. Hasil dan Pembahasan

3.1. Hasil Analisis

Hasil pengujian terhadap 15 skenario pengujian yang telah dilakukan terangkum pada Tabel 3.

Tabel 3. Rangkuman hasil pengujian 15 skenario

Skenario ke-	Akurasi	Presisi	*Recall*	F1-score	Grafik
1	91,67%	92,67%	91,67%	91,62%	Overfitting
2	96,00%	96,13%	96,00%	96,00%	Good fit
3	96,33%	96,58%	96,33%	96,33%	Good fit
4	96,67%	96,70%	96,67%	96,67%	Good fit

Skenario ke-	Akurasi	Presisi	*Recall*	F1-score	Grafik
5	97,33%	97,41%	97,33%	97,33%	Good fit
6	91,33%	91,94%	91,33%	91,30%	Good fit
7	92,67%	93,04%	92,67%	92,65%	Good fit
8	94,67%	94,74%	94,67%	94,66%	Good fit
9	95,67%	95,77%	95,67%	95,66%	Overfitting
10	96,67%	96,80%	96,67%	96,66%	Good fit
11	89,33%	89,45%	89,33%	89,33%	Overfitting
12	90,67%	90,67%	90,67%	90,67%	Good fit
13	94,33%	94,34%	94,34%	94,33%	Overfitting
14	95,67%	96,01%	95,67%	95,66%	Overfitting
15	96,33%	96,43%	96,33%	96,33%	Good fit

Berdasarkan Tabel 3, terlihat bahwa variasi ukuran batch (*batch size*) dan jumlah *epoch* mempengaruhi performa model dalam klasifikasi, khususnya pada nilai akurasi, presisi, recall, dan f1-score. Pada skenario 1 hingga 5, yang menggunakan *batch size* 32, terdapat peningkatan tren terhadap performa seiring dengan bertambahnya jumlah *epoch* dari 10 hingga 50. Pada skenario 1 dengan 10 *epoch*, akurasi model mencapai 91,67%. Ketika jumlah *epoch* meningkat menjadi 20 di skenario 2, akurasi naik signifikan menjadi 96,00%. Tren peningkatan ini terus berlanjut hingga skenario 5, di mana model dengan 50 *epoch* mendapatkan akurasi tertinggi sebesar 97,33%. Hal ini menunjukkan bahwa model dengan *batch size* 32 mampu belajar lebih efektif saat jumlah *epoch* ditingkatkan, sehingga semakin banyak pola yang bisa dipelajari dari data.

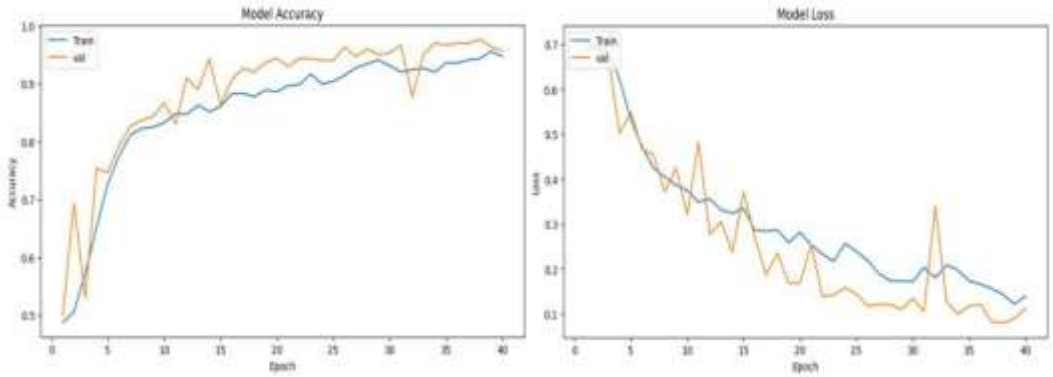
Pada skenario 6 hingga 10, di mana *batch size* dinaikkan menjadi 64, tren peningkatan performa juga meningkat, meskipun sedikit lebih rendah dibandingkan dengan *batch size* 32. Di skenario 6, dengan 10 *epoch*, didapatkan akurasi model sebesar 91,33%. Dengan penambahan *epoch*, akurasi meningkat secara bertahap, mencapai 96,67% pada skenario 10 (*epoch* 50). Berdasarkan hal tersebut, terlihat bahwa *batch size* 64 juga efektif meningkatkan akurasi, tetapi memiliki performa yang tidak setinggi *batch size* 32 pada jumlah *epoch* yang sama. Namun, pada *batch size* 64 tetap memberikan hasil yang stabil, dengan akurasi, presisi, recall, dan f1-score yang cukup tinggi dan konsisten.

Pada skenario 11 hingga 15, dengan *batch size* yang lebih besar yaitu 128, peningkatan akurasi juga terjadi seiring bertambahnya jumlah *epoch*, tetapi hasilnya cenderung lebih rendah dibandingkan *batch size* yang lebih kecil. Pada skenario 11 dengan 10 *epoch*, akurasi yang dihasilkan hanya mencapai 89,33%, lebih rendah bila dibandingkan dengan *batch size* 32 dan 64 pada *epoch* yang sama. Bahkan pada skenario 15 dengan 50 *epoch*, akurasi mencapai 96,33%, tetapi masih sedikit lebih rendah bila dibandingkan *batch size* 32 pada *epoch* 50. Hal ini menunjukkan bahwa *batch size* yang lebih besar cenderung membuat model lebih sulit untuk mencapai akurasi yang optimal pada jumlah *epoch* yang sama, hal tersebut mungkin disebabkan karena model perlu melakukan pengolahan data yang lebih banyak secara sekaligus, sehingga proses pembelajarannya menjadi lebih lambat. Selain itu, overfitting juga terjadi pada beberapa skenario, seperti pada skenario 1, 9, dan 11. Berdasarkan analisis grafik menunjukkan adanya tanda-tanda overfitting. Sebaliknya, sebagian besar skenario lain justru memberikan hasil grafik dengan kondisi good fit, yang berarti model dapat mempelajari pola dengan baik pada data pelatihan. Kondisi good fit ini terlihat pada skenario 2 hingga 8 dan skenario 10 hingga 15, di mana akurasi, presisi, recall, dan f1-score semuanya menunjukkan nilai yang tinggi dan stabil.

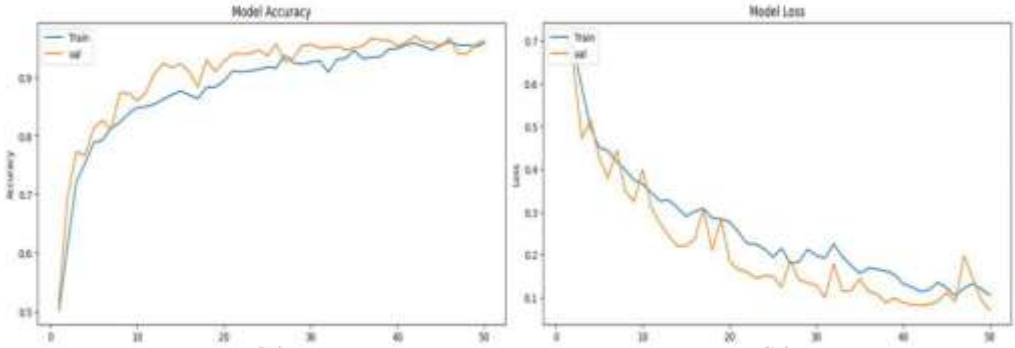
Secara keseluruhan, berdasarkan hasil analisis menunjukkan bahwa *batch size* yang lebih kecil yaitu 32, memungkinkan model untuk mencapai akurasi yang lebih tinggi pada jumlah *epoch* yang sama dibandingkan *batch size* yang lebih besar.

Peningkatan jumlah *epoch* pada setiap *batch size* umumnya meningkatkan performa model, tetapi justru membatasi performa pada *batch size* lebih besar seperti 128. Berdasarkan pada Tabel 3, maka dapat disimpulkan bahwa performa model ternyata dipengaruhi oleh kombinasi antara ukuran batch dan jumlah *epoch* yang digunakan. Ukuran batch yang lebih kecil mungkin dapat meningkatkan performa yang lebih konsisten pada berbagai *epoch*, sedangkan *batch size* yang lebih besar seperti 128 membutuhkan lebih banyak *epoch* untuk mendapatkan akurasi yang mendekati *batch size* yang lebih kecil. Namun, dengan nilai *epoch* yang semakin tinggi pada *batch size* yang besar memungkinkan terjadinya *overfitting*.

Adapun gambaran mengenai grafik akurasi dan loss yang dihasilkan dari skenario 14 dan skenario 15 terdapat pada Gambar 4 dan Gambar 5, yang sama-sama menggunakan *batch size* 128 tetapi dengan jumlah *epoch* yang berbeda (40 dan 50).



Gambar 4. Grafik akurasi (kiri) dan grafik loss (kanan) pada skenario 14



Gambar 5. Grafik akurasi (kiri) dan grafik loss (kanan) pada skenario 15

Dari Gambar 4 dan Gambar 5, merupakan hasil grafik akurasi dan loss pada dua skenario 14 dan skenario 15. Dimana kedua skenario ini memiliki *batch size* yang sama, yaitu 128, tetapi berbeda dalam jumlah *epoch* yang digunakan (40 untuk skenario 14 dan 50 untuk skenario 15). Berikut adalah analisis mendetail terkait grafik akurasi dan loss, serta hubungannya dengan hasil yang ada pada Tabel 3.

a. Pada skenario 14 (Gambar 4), model menunjukkan grafik akurasi yang stabil dengan nilai akurasi validasi yang mendekati akurasi data pelatihan seiring bertambahnya *epoch*. Akurasi validasi dan pelatihan bergerak naik dengan cepat di awal *epoch* dan mulai stabil setelah sekitar *epoch* ke-20. Di akhir proses, akurasi validasi dan pelatihan hampir menyentuh 95%, yang mencerminkan good fit, sebagaimana yang tercantum di Tabel 3. Sedangkan pada grafik loss di sisi kanan, terlihat bahwa nilai loss untuk data pelatihan dan validasi mengalami penurunan yang cukup signifikan di awal, kemudian mulai stabil pada sekitar *epoch* ke-20, meskipun terdapat sedikit fluktuasi. Nilai loss validasi cenderung lebih rendah dibandingkan nilai loss pelatihan pada beberapa titik, yang

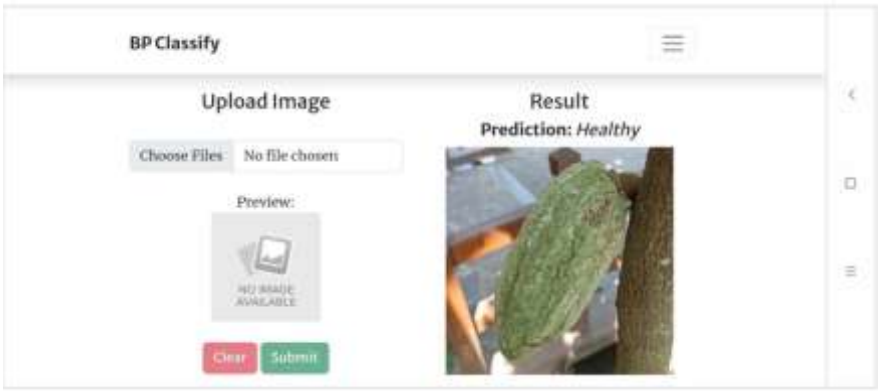
mengindikasikan bahwa model mampu belajar dengan baik tanpa mengalami overfitting. Tren ini menguatkan hasil pada Tabel 3, di mana skenario 14 menunjukkan akurasi tinggi dengan nilai presisi, recall, dan f1-score yang konsisten di angka 95,67%.

b. Pada skenario 15 (Gambar 5), model akan dilatih hingga 50 *epoch*, dan hasilnya menunjukkan peningkatan yang sedikit lebih tinggi dibandingkan skenario 14. Pada grafik akurasi, terlihat bahwa akurasi validasi dan pelatihan meningkat dengan cepat hingga sekitar *epoch* ke-20, dan kemudian stabil di sekitar angka 96%. Ini menunjukkan bahwa menambah jumlah *epoch* hingga 50 membantu model mencapai akurasi yang sedikit lebih tinggi dibandingkan skenario 14, yang tercermin di Tabel 3 dengan akurasi 96,33%, presisi 96,43%, recall 96,33%, dan f1-score 96,33%. Sedangkan untuk grafik loss pada skenario 15 juga menunjukkan tren penurunan yang stabil hingga sekitar *epoch* ke-20, dan kemudian fluktuasi kecil terjadi. Meski ada sedikit peningkatan di beberapa titik, nilai loss validasi tetap lebih rendah atau setara dengan nilai loss pelatihan, hal ini menunjukkan bahwa model tetap berada dalam kondisi good fit dan ini mengindikasikan bahwa model mampu mempelajari data dengan baik tanpa overfitting meskipun menggunakan jumlah *epoch* yang lebih tinggi.

Apabila hasil Tabel 3 dibandingkan dengan grafik pada Gambar 4 dan 5 tersebut terlihat bahwa dengan *batch size* 128 model akan mencapai akurasi yang stabil pada skenario 14 (*epoch* 40) dan sedikit meningkat pada skenario 15 (*epoch* 50). Namun, peningkatan performa dari *epoch* 40 ke *epoch* 50 pada *batch size* yang sama (128) tidak terlalu signifikan, yang menunjukkan bahwa menambah *epoch* di atas 40 hanya memberikan sedikit manfaat dalam hal akurasi dan metrik performa lainnya. Secara keseluruhan, dari analisis ini dapat disimpulkan bahwa penggunaan *epoch* antara 40 hingga 50 pada *batch size* 128 memberikan hasil yang cukup optimal, dengan nilai akurasi, presisi, recall, dan f1-score yang tinggi dan stabil, serta grafik loss yang menurun secara konsisten tanpa indikasi overfitting.

3.2. Pengujian

Seperti yang telah dijelaskan sebelumnya, maka pada tahapan pengujian ini akan dilakukan pengujian terhadap 20 citra asli di luar dataset citra yang digunakan dari Kaggle. Citra ini didapatkan dari hasil pemotretan langsung di Kampung Coklat yang merupakan salah satu destinasi wisata edukasi kakao yang berada di Kabupaten Blitar Jawa Timur. Gambar 6 merupakan salah satu hasil pengujian yang dilakukan.



Gambar 6. Hasil pengujian

Rangkuman terhadap hasil pengujian 20 citra baru tersebut terdapat pada Tabel 4.

Tabel 4. Hasil pengujian 20 citra asli

Pengujian ke-	Citra asli	Hasil pengujian	Kesimpulan
1	*Black pod*	*Black pod*	Citra terprediksi secara benar sebagai *black pod*
2	*Black pod*	*Black pod*	Citra terprediksi secara benar sebagai *black pod*
3	*Black pod*	*Healthy*	citra terprediksi secara salah sebagai *Healthy*
4	*Black pod*	*Black pod*	citra terprediksi secara benar sebagai *Black pod*
5	*Black pod*	*Black pod*	citra terprediksi secara benar sebagai *Black pod*
6	*Black pod*	*Black pod*	citra terprediksi secara benar sebagai *Black pod*
7	*Black pod*	*Healthy*	citra terprediksi secara salah sebagai *Healthy*
8	*Black pod*	*Black pod*	citra terprediksi secara benar sebagai *Black pod*
9	*Black pod*	*Black pod*	citra terprediksi secara benar sebagai *Black pod*
10	*Black pod*	*Black pod*	citra terprediksi secara benar sebagai *Black pod*
11	*Healthy*	*Healthy*	citra terprediksi secara benar sebagai *healthy*
12	*Healthy*	*Healthy*	citra terprediksi secara benar sebagai *healthy*
13	*Healthy*	*Healthy*	citra terprediksi secara benar sebagai *healthy*
14	*Healthy*	*Healthy*	citra terprediksi secara benar sebagai *healthy*
15	*Healthy*	*Healthy*	citra terprediksi secara benar sebagai *healthy*
16	*Healthy*	*Healthy*	citra terprediksi secara benar sebagai *healthy*
17	*Healthy*	*Healthy*	citra terprediksi secara benar sebagai *healthy*
18	*Healthy*	*Healthy*	citra terprediksi secara benar sebagai *healthy*
19	*Healthy*	*Healthy*	citra terprediksi secara benar sebagai *healthy*
20	*Healthy*	*Healthy*	citra terprediksi secara benar sebagai *healthy*

Berdasarkan hasil pengujian terhadap 20 citra didapatkan bahwa terdapat 8 citra *black pod* yang terprediksi sebagai *black pod*, 10 citra *healthy* terprediksi sebagai *healthy*, 2 citra *black pod* terprediksi sebagai *healthy* dan 0 citra *healthy* terprediksi sebagai *black pod*. Dengan demikian, pengujian di web menghasilkan nilai akurasi sebesar 90% sehingga model Algoritma CNN yang telah dibuat cukup baik untuk klasifikasi citra *Black pod disease* buah kakao.

4. Kesimpulan

Implementasi algoritma CNN untuk klasifikasi citra penyakit *black pod* pada buah kakao dilakukan dengan menggunakan 1.500 data citra RGB buah kakao berukuran 150×150 piksel. Berdasarkan 15 skenario pengujian yang dilakukan menunjukkan hasil bahwa peningkatan jumlah *epoch* secara umum dapat meningkatkan nilai akurasi, sedangkan peningkatan *batch size* justru cenderung membuat nilai akurasi yang lebih kecil. Selain itu model terbaik juga diperoleh pada skenario ke-5 dengan *batch size* terkecil (32) dan *epoch* terbesar (50) dengan hasil akurasi 97,33%, presisi 97,41%, recall 97,33%, dan f1-score 97,33%. Ketika diuji di web dengan dataset baru sebanyak 20 kali pengujian, didapatkan hasil bahwa model yang digunakan mampu mencapai akurasi sebesar 90%. Secara keseluruhan, model Convolutional Neural Network yang dikembangkan mampu mendeteksi penyakit *black pod disease* dengan kinerja yang baik dalam klasifikasi citra penyakit busuk buah kakao.

Daftar Pustaka

[1] V. Zikria, S. Sofyan, S. Safrida, and E. Susanti, "Analisis Wilayah dan Kontribusi Kakao Terhadap Pembangunan Daerah di Kabupaten Parigi Moutong," *J. Agriuma*, vol. 4, no. 1, pp. 22–30, 2022, doi: 10.31289/agri.v4i1.7173.

[2] W. Rumahlewang, H. R.D. Amanupunyo, and B. S. Tomia, "Kerusakan Buah Kakao Akibat Penyakit Busuk Buah (Phytopthora palmivora Butlher)," *COMSERVA J. Penelit. dan Pengabdi. Masy.*, vol. 2, no. 7, pp. 956–962, 2022, doi: 10.59141/comserva.v2i7.427.

[3] M. Minarni, E. Novira, E. Yulianti, and G. Yoga Swara, "Sistem Pakar Diagnosa Penyakit dan Hama Tanaman Kakao Dengan Metode Certainty Factor," *SATIN - Sains dan Teknol. Inf.*, vol. 8, no. 2, pp. 79–88, 2022, doi: 10.33372/stn.v8i2.901.

[4] P. N. Indah, S. Sugiyarto, C. A. Putra, L. Endahwati, and H. Maulana, *Pasca Panen Kakao & Pembuatan Sabun Kecantikan*, 2nd ed. Surabaya: Unggul Pangestu Nirmala, 2021.

[5] F. P. Soge, B. Deta, and D. B. Watomakin, "Penerapan Logika Fuzzy dalam Sistem Pakar Deteksi Hama dan Penyakit Tanaman Kakao Kecamatan Wulanggitang Flores Timur," *J. Kridatama Sains Dan Teknol.*, vol. 6, no. 01, pp. 212–234, 2024, doi: 10.53863/kst.v6i01.1115.

[6] S. Syaifuddin, O. Maliki, and H. W. Kamase, "Implementasi Sistem Pakar Android untuk Deteksi Penyakit Tanaman Coklat dengan Metode Certainty Factor," *Bull. Inf. Technol.*, vol. 4, no. 4, pp. 456–464, 2023, doi: 10.47065/bit.v4i4.1019.

[7] A. Heri, W. Wahdaniah, and H. AP, "Sistem Pakar Diagnosa Penyakit Pada Tanaman Kakao Menggunakan Metode Dempster Shafer," *J. Comput. Inf. Syst. (J-CIS)*, vol. 6, no. 2, pp. 9–18, 2023, doi: 10.31605/jcis.v6i2.3428.

[8] G. H. Yogiswara, R. Magdalena, H. F. T. S. P, F. T. Elektro, and U. Telkom, "Identifikasi Jenis Penyakit Pada Kakao Dengan Pengolahan Citra Digital Dan K-Nearest Neighbor," *E-Proceding Eng.*, vol. 3, no. 1, pp. 371–377, 2016.

[9] D. S. Tan *et al.*, "A framework for measuring infection level on cacao pods," *Proc. - 2016 IEEE Reg. 10 Symp. TENSYMP 2016*, pp. 384–389, 2016, doi: 10.1109/TENCONSpring.2016.7519437.

[10] A. J. Rozaqi, A. Sunyoto, and M. rudyanto Arief, "Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network," *Creat. Inf. Technol. J.*, vol. 8, no. 1, p. 22, 2021, doi: 10.24076/citec.2021v8i1.263.

[11] I. Supiyani and N. Arifin, "Identifikasi Nomor Rumah Pada Citra Digital Menggunakan Neural Network," *Method. J. Tek. Inform. dan Sist. Inf.*, vol. 8, no. 1, pp. 18–21, 2022, doi: 10.46880/mtk.v8i1.921.

[12] A. Jakaria, S. Mu'minah, D. Riana, and S. Hadianti, "Klasifikasi Varietas Buah Kiwi dengan Metode Convolutional Neural Networks Menggunakan Keras," *J. Media Inform. Budidarma*, vol. 5, no. 4, p. 1309, 2021, doi: 10.30865/mib.v5i4.3166.

[13] M. F. Naufal, "Analisis Perbandingan Algoritma SVM, KNN, dan CNN untuk Klasifikasi Citra Cuaca," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 2, pp. 311–318, 2021, doi: 10.25126/jtiik.2021824553.