



Implementasi Metode *Convolutional Neural Network* Untuk Deteksi Penyakit Pada Tanaman Kopi Arabika Melalui Citra Daun Berbasis Android

Andre Ilham Saputra¹⁾, Indra Weni¹⁾, Ulfa Khaira¹⁾

¹⁾Program Studi Sistem Informasi, Universitas Jambi, Indonesia.

Artikel Info

Kata Kunci:

Convolutional Neural Network;
Hyperparameter;
Images;
Klasifikasi;
Penyakit Kopi Arabika.

Keywords:

Convolutional Neural Network;
Classification;
Images;
Hyperparameter;
Arabica Coffee Diseases.

Riwayat Article:

Submitted: 3 Juni 2023
Accepted: 19 Juli 2023
Published: 4 Oktober 2023

Abstrak: Penyakit pada tanaman kopi arabika merupakan masalah serius yang dapat menyebabkan kerugian bagi petani jika tidak terdeteksi dan ditangani secara tepat waktu. Dalam penelitian ini, digunakan metode *Convolutional Neural Network* (CNN) untuk mendeteksi penyakit pada tanaman kopi arabika. Metode CNN dipilih karena kemampuannya dalam memproses data citra dan mempelajari pola yang kompleks, terutama dalam konteks identifikasi penyakit pada tanaman. Dataset yang digunakan berupa citra daun kopi arabika yang dikategorikan menjadi tiga kelas: Penyakit Bercak Daun, Karat Daun, dan Daun Sehat dengan jumlah total 2829 citra daun yang kemudian dibagi dengan skenario 80:20 untuk data latih dan data uji. Pada penelitian ini dilakukan pengujian *hyperparameter* dengan memvariasikan jumlah *epoch*, nilai *batch size* dan optimizer. Hasil penelitian menunjukkan bahwa peningkatan jumlah *epoch* dan nilai *batch size* berdampak terhadap akurasi model CNN. Dari pengujian *hyperparameter* didapatkan model optimum dengan menggunakan 50 *epoch*, *batch size* 32, dan optimizer Adam yang mencetak akurasi sebesar 94.33% pada proses pengujian.

Abstract: Diseases in Arabica coffee plants are a serious problem that can cause losses to farmers if not detected and treated in a timely manner. In this research, the Convolutional Neural Network (CNN) method is used to detect diseases in arabica coffee plants. The CNN method was chosen because of its ability to process image data and learn complex patterns, especially in the context of plant disease identification. The dataset used is an image of Arabica coffee leaves categorized into three classes: Leaf Spot Disease, Leaf Rust, and Healthy Leaves with a total of 2829 leaf images which are then divided by an 80:20 scenario for training data and test data. In this study, hyperparameter testing was carried out by varying the number of epochs, batch size value and optimizer. The results showed that increasing the number of epochs and batch size value had an impact on the accuracy of the CNN model. From hyperparameter testing, the optimum model was obtained using 50 epochs, batch size 32, and Adam optimizer which scored an accuracy of 94.33% in the testing process.

Corresponding Author:

Ulfa Khaira

Program Studi Sistem Informasi, Universitas Jambi, Indonesia.

Alamat: Jl. Jambi – Muara Bulian No.KM. 15, Mendalo Darat, Kec. Jambi Luar Kota, Kabupaten

Muaro Jambi, Jambi, Indonesia

Email: ulfa.ilkom@gmail.com

PENDAHULUAN

Kopi merupakan komoditas tropis utama dalam subsektor perkebunan yang memiliki kontribusi signifikan bagi perekonomian Indonesia, baik di pasar domestik maupun internasional (Putri & Sodik, 2019). Indonesia merupakan negara produsen kopi terbesar keempat di dunia dan mengalami peningkatan produksi dari tahun ke tahun jumlah produksi kopi Indonesia mencapai 774,60 ribu ton pada 2021. Jumlah itu meningkat sekitar 1,62% dari tahun sebelumnya yang sebanyak 762,20 ribu ton (BPS, 2022) dan akan diprediksi terus meningkat pada tahun-tahun berikutnya. Kopi arabika (*Coffea arabica*) mempunyai mutu cita rasa terbaik dibandingkan jenis kopi yang lain, dengan ciri biji berbentuk putih kehijauan dan daun hijau tua. Kopi arabika memerlukan waktu 6-8 bulan sejak dari kuncup sampai matang (Prastowo et al., 2010). Harga yang lebih tinggi dari jenis kopi lainnya juga seharusnya menjadikan kopi arabika sebagai peluang besar dalam industri perekonomian. Namun, tanaman kopi rentan terhadap serangan penyakit, yang dapat menyebabkan kerugian yang besar jika tidak ditangani dengan baik (Ramadhan et al., 2021). Oleh karena itu, deteksi penyakit pada tanaman kopi menjadi penting dalam menjaga produktivitas dan kualitas hasil produksi (Singh & Misra, 2017).

Proses identifikasi penyakit pada tanaman kopi umumnya dilakukan secara manual oleh petani, namun metode ini memiliki beberapa keterbatasan, seperti waktu yang diperlukan yang relatif lama, perbedaan persepsi tentang jenis penyakit, dan keterbatasan visual manusia. Kurangnya pengetahuan dan informasi petani tentang jenis penyakit juga dapat mengakibatkan penanganan yang tidak tepat, yang berpotensi memperburuk penyebaran penyakit dan menurunkan kualitas dan tingkat produktivitas kopi (Gunawan et al., 2020).

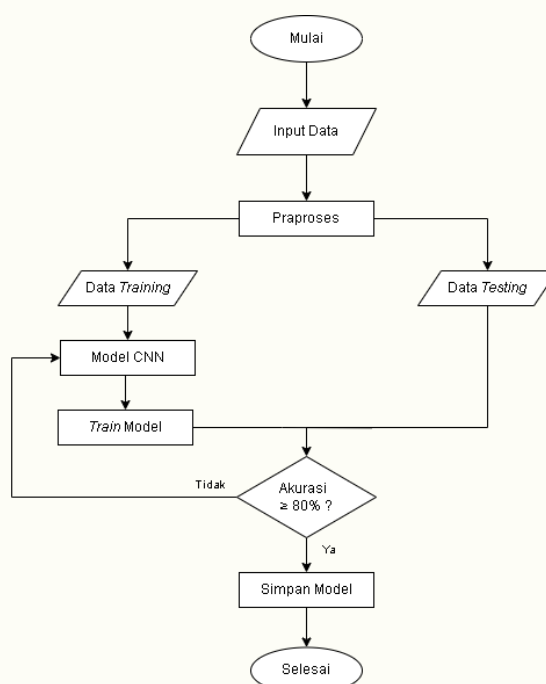
Dalam beberapa tahun terakhir, kecerdasan buatan atau *artificial intelligence* (AI) telah menjadi topik hangat yang banyak diterapkan dalam berbagai sektor industri, termasuk pertanian. Saat ini, teknologi *deep learning* telah mengalami kemajuan yang luar biasa dalam hal klasifikasi objek pada gambar atau pengenalan pola (Lecun, Bengio, & Hinton, 2015). Penggunaan teknologi AI, khususnya metode *deep learning* dengan menggunakan *Convolutional Neural Network* (CNN), dapat memberikan solusi dalam identifikasi penyakit tanaman. CNN merupakan salah satu algoritma *deep learning* yang umum digunakan untuk menyelesaikan masalah identifikasi penyakit tanaman dalam kurun waktu lima tahun terakhir (Sabrina, 2022).

Pengenalan pola untuk identifikasi penyakit pada tanaman sudah banyak menjadi fokus penelitian dengan berbagai tujuan dan metode yang berbeda. Seperti penelitian yang dilakukan oleh Anggi Yhurinda Perdana Putri dan Anwar Sodik pada 2019 menggunakan algoritma *K-Nearest Neighbor* (K-NN) untuk mengidentifikasi penyakit pada tanaman kopi arabika, akan tetapi akurasi yang didapatkan masih terbilang rendah dengan persentase yaitu 65,99% (Putri & Sodik, 2019). Kemudian penelitian dengan metode yang sama juga dilakukan oleh Serawork Wallelign, Mihai Polceanu, Towfik Jemal, dan Cédric Buche di tahun 2019 yang menggunakan CNN untuk mengidentifikasi mutu atau 12 *grade* biji kopi, Hasil penelitian ini memberi akurasi sebesar 89,1%. Penelitian ini juga menunjukkan bahwa CNN dapat mengekstrak fitur dan mengklasifikasikan gambar *input* dengan pemrosesan awal yang minimal (Wallelign, Polceanu, Jemal, & Buche, 2019). Metode CNN memiliki hasil paling signifikan dalam klasifikasi citra, hal tersebut dikarenakan CNN berusaha meniru sistem pengenalan citra pada *visual cortex* manusia (Putra, 2016).

Berdasarkan latar belakang diatas fokus penelitian ini adalah bagaimana mengidentifikasi penyakit pada tanaman kopi arabika (*coffea arabica*) berdasarkan citra daun dengan menerapkan implementasi dari metode *deep learning* menggunakan CNN dan akan diterapkan ke dalam aplikasi *mobile* berbasis Android. Dengan adanya penelitian ini, diharapkan petani dan masyarakat umum dapat memanfaatkan teknologi AI untuk meningkatkan kemampuan dalam mengidentifikasi penyakit pada tanaman kopi. Hal ini diharapkan dapat mengurangi kerugian yang disebabkan oleh penyakit, menjaga produktivitas tanaman, dan meningkatkan kualitas hasil produksi kopi.

METODE

Pada penelitian ini, dilakukan penelitian mengenai citra daun pada kopi arabika (*coffee arabica*) yang mengalami penyakit karat daun dan penyakit bercak daun (*brown eye-spot*). Data dikumpulkan dari dua sumber, pertama-tama data dikumpulkan langsung di lokasi perkebunan kopi arabika PTPN VI yang berlokasi di Kayu Aro, Kabupaten Kerinci, Provinsi Jambi, Indonesia dengan menggunakan kamera *smartphone* sesuai dengan kelas yang ditentukan dan kemudian dihilangkan noise atau latar belakang pada tiap citra menggunakan layanan daring *remove.bg* sehingga menghasilkan citra yang berfokus pada objek daun. Selain itu, data juga dikumpulkan dari internet melalui situs *data.mendeley.com*. Data tersebut juga terdiri dari tiga kategori, yaitu citra daun yang mengalami penyakit bercak daun, karat daun, dan citra daun yang dalam kondisi sehat. Waktu penelitian dilakukan mulai dari bulan Desember 2022 hingga April 2023. Secara umum langkah-langkah perancangan sistem dapat dilihat pada gambar 1 berikut.



Gambar 1. Langkah -Langkah Perancangan Sistem

Dataset yang telah dikumpulkan akan diubah dan diseragamkan menjadi ukuran piksel 256x256 dengan tiga *channel* (*Red*, *Green*, dan *Blue*) dan dilakukan proses augmentasi untuk mencegah *overfitting* dan kemudian dilakukan penyemimbangan jumlah dataset. Hal ini merupakan tahapan dari tahap praproses dataset. Dataset yang telah melalui tahapan *preprocessing* akan dibagi menjadi dua bagian menggunakan metode *train test split*, dengan perbandingan 80:20 untuk data *training* dan data *validation*. Data *training* digunakan untuk membangun model, sedangkan data *validation* digunakan untuk menguji dan memvalidasi model yang telah dilatih.

Menurut (Ashshiddeqy et al., 2020) Augmentasi merupakan suatu proses yang bertujuan untuk mencegah terjadinya *overfitting*, dengan kata lain augmentasi bertujuan untuk menambah varian data pada saat proses pelatihan atau *training* model. Dengan dilakukannya augmentasi pada data, maka gambar citra daun akan dimodifikasi sedemikian rupa sehingga komputer mendeteksi bahwa gambar yang diubah merupakan gambar yang berbeda (Perez & Wang, 2017). Pada penelitian ini proses augmentasi dilakukan dengan mengubah *brightness* dan *contrast* dari citra daun secara acak, melakukan rotasi, translasi, dan *crop* secara acak. Kemudian data yang telah diaugmentasi dilakukan pembagian atau *split dataset* dengan skenario pembagian yaitu 80:20 untuk data *training* dan *validation*.

Selanjutnya, dilakukan desain arsitektur sistem menggunakan *Convolutional Neural Network* (CNN). Data *training* dengan ukuran 256x256 piksel digunakan sebagai *input* untuk model CNN. Model ini terdiri dari beberapa lapisan konvolusi dengan *filter* yang berbeda, dilanjutkan dengan proses *flatten layer* yang digunakan untuk mengubah dimensi peta fitur menjadi nilai vektor satu dimensi (Ng et al., 2019), dan kemudian dilanjutkan dengan lapisan *fully connected layer* untuk proses klasifikasi. Lapisan akhir menggunakan fungsi aktivasi *softmax* untuk mengklasifikasikan citra daun menjadi tiga kelas.

Untuk mendapatkan model yang optimal, pada penelitian ini akan dilakukan pengujian *hyperparameter*. Tidak ada pendekatan yang konsisten untuk mengidentifikasi konfigurasi *hyperparameter* yang optimal, meskipun *hyperparameter* memainkan peran krusial dalam menentukan struktur keseluruhan jaringan (Neary, 2018). Pada penelitian ini *hyperparameter* yang digunakan untuk percobaan adalah sebagai berikut.

1. *Batch size* dengan skenario 32 dan 64.
2. *Epoch* dengan skenario data 10, 30, dan 50.
3. Optimizer yang digunakan adalah Adam, Adamax, dan RMSProp.

Implementasi model CNN dilakukan menggunakan *Jupyter Notebook* dengan bahasa pemrograman Python 3.8 dan paket *TensorFlow*. Sedangkan untuk mengintegrasikan model ke dalam antarmuka pengguna berbasis android, digunakan *framework* Flutter dengan bahasa pemrograman Dart dan *Visual Studio Code* sebagai IDE.

Setelah mendapatkan nilai prediksi, dilakukan perhitungan akurasi keseluruhan dengan menggunakan persamaan-persamaan berikut.

$$\text{Overall Accuracy} = \frac{TTP_{all}}{\text{Total Number of Testing Entries}} \quad (1)$$

Akurasi digunakan sebagai parameter untuk mengukur seberapa akurat model dalam melakukan klasifikasi. Persamaan diatas memaparkan TTPall sebagai *True Positives* yaitu banyak kejadian yang benar dibagi dengan *n* atau jumlah keseluruhan data diprediksi (Rasyidi & Bariyah, 2020).

$$\text{Precision} = \frac{TP_{class}}{TP_{class} + \sum FP_{class}} \quad (2)$$

$$\text{recall} = \frac{TTP_{class}}{TP_{class} + \sum FP_{class}} \quad (3)$$

$$f1 - \text{score} = \frac{\sum(\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \quad (4)$$

Precision menggambarkan seberapa tepat model memprediksi kejadian positif dalam serangkaian kegiatan prediksi. Selain itu, untuk melihat lebih detail lagi kinerja model, *recall* atau sensitifitas dan nilai *F1-score* juga dapat digunakan.

Selanjutnya, dirancang antarmuka pengguna grafis (GUI) pada aplikasi Android. Tujuan dari aplikasi Android ini adalah memudahkan pengguna atau petani kopi arabika dalam mendeteksi penyakit pada tanaman kopi arabika. Arsitektur aplikasi Android ini memproses citra daun kopi arabika yang diinputkan ke dalam aplikasi. Citra tersebut kemudian diproses dan diklasifikasikan menggunakan model CNN yang terintegrasi dalam aplikasi Android melalui *framework* TensorFlow dan tflite. Hasilnya adalah prediksi tentang jenis penyakit yang diduga oleh sistem terhadap citra daun tersebut.

HASIL DAN PEMBAHASAN

Hasil dari penelitian ini adalah untuk mengukur akurasi yang diperoleh menggunakan metode *Convolutional Neural Network* dalam mengidentifikasi penyakit pada tanaman kopi arabika menggunakan yang diambil langsung di perkebunan kopi arabika milik PT Perkebunan Nusantara VI

dan data.mendeley.com. Berikut merupakan sebaran data citra setelah dilakukan *data augmentation* dan *data balancing* data pada tahap *data preprocessing*.

Tabel 1. Jumlah Dataset Setelah Dilakukan Penyeimbangan Data Citra

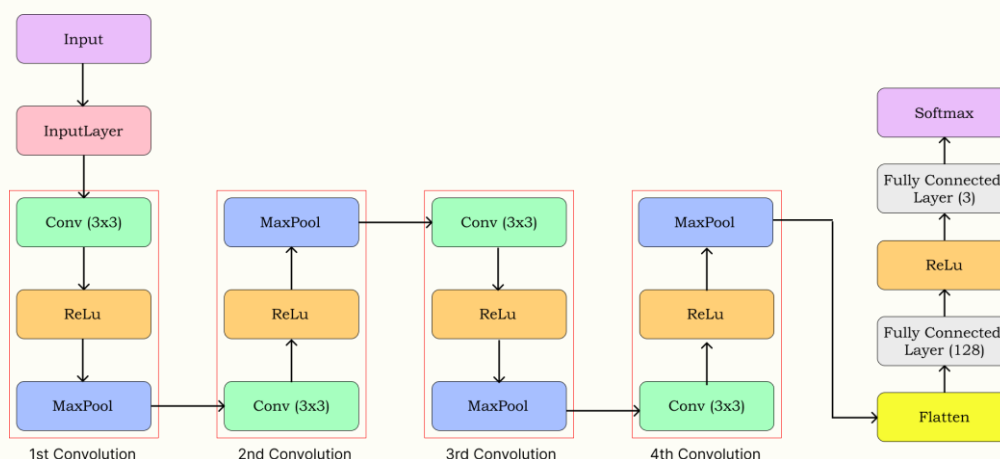
No.	Nama Kelas	Jumlah Data Awal	Setelah diaugmentasi	Setelah <i>balanced</i>
1.	Bercak Daun	100	944	943
2.	Karat Daun	100	950	943
3.	Sehat	100	943	943
	Total	300	2837	2829

Menurut (Wang, Sun, & Wang, 2017) jumlah dataset untuk tiap kelas haruslah seimbang (*balanced*) untuk mengurangi bias yang mungkin dimiliki jaringan terhadap kelas yang memiliki lebih banyak sampel, inilah alasan mengapa terjadi pengurangan dari data augmentasi dengan strategi memangkas dan menyamaratakan tiap kelas dengan jumlah terkecil dari ketiga kelas sebelumnya yaitu 943 citra pada kelas sehat, sehingga akan dilakukan penghapusan beberapa citra pada kelas bercak daun dan karat daun. Data citra selanjutnya akan dibagi kedalam dua folder direktori yaitu *train* dan *validation* dengan skema skenario pembagian yaitu 80:20, 80% untuk data *training* dan 20% untuk data *validation* secara acak (Ahmad & Khan, 2021).

Tabel 2. Jumlah Sebaran Data Training dan Validation

No.	Nama Kelas	<i>training</i>	<i>validation</i>
1.	Bercak Daun	754	188
2.	Karat Daun	754	188
3.	Sehat	754	188
	Total	2265	564

Setelah dilakukan berbagai tahapan hingga pada pembagian data, langkah selanjutnya adalah menentukan arsitektur dan parameter CNN yang akan digunakan pada proses *training*. Pada penelitian ini model CNN yang digunakan terdiri dari tiga lapisan konvolusi, tiga lapisan *pooling*, dan dua lapisan *Fully Connected Layer* serta satu lapisan *output*.



Gambar 2. Rancangan Arsitektur CNN

Tahap pertama adalah *Convolution Layer*, di mana dilakukan konvolusi untuk mengekstraksi ciri pada citra *input* dan menghasilkan feature maps. Pada penelitian ini juga digunakan *padding same*, yaitu *padding* yang nilai *output*nya sama dengan nilai *input*. Dengan menggunakan *padding* ini maka dilakukan penambahan nilai 0 di sekeliling nilai matriks *input* agar nilai *input* dan *output* memiliki

matriks yang sama dan tidak mengurangi informasi pada gambar. Berikut ini merupakan contoh perhitungan untuk nilai *output* yang menggunakan *padding same*.

$$\begin{aligned} o &= i + 2p - (k - 1) \\ o &= 256 + 2(1) - (3-1) \\ o &= 256 + 2 - 2 = 256 \end{aligned}$$

Dengan O adalah *output*, i adalah *input*, p adalah *padding*, dan k adalah nilai kernel (Dumoulin & Visin, 2016). Dengan menggunakan *padding*, dimensi *output* dapat diatur agar tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara drastis. Sehingga bisa menggunakan *convolution layer* yang lebih dalam sehingga lebih banyak *features* yang berhasil di-extract.

Tahap kedua adalah *Pooling*, di mana dilakukan *Convolution Layer* untuk mengurangi dimensi *feature maps* dan mempercepat komputasi. *Convolution Layer* dengan ukuran 2x2 digunakan untuk mengambil nilai terbesar dari setiap blok piksel. Tahap terakhir adalah *Fully Connected Layer*, yang bertanggung jawab atas klasifikasi gambar dengan menghubungkan *output* dari tahap sebelumnya. Di sini, vektor *input* diratakan dan dikalikan dengan bobot dan bias yang sesuai untuk menghasilkan *output* pada setiap *neuron*. Teknik regularisasi seperti *dropout* dan *L2 regularization* juga digunakan untuk menghindari *overfitting* pada model. *Fully Connected Layer* merupakan komponen penting dalam CNN dan penggunaan teknik regularisasi dapat meningkatkan akurasi dan mencegah *overfitting* pada model.

Untuk mendapatkan model yang optimal, pada penelitian ini juga dilakukan pengujian *hyperparameter* yang meliputi *EPOCH*, *optimizer*, dan *batch size*. Hasil pengujian ini terlampir pada tabel 3 berikut.

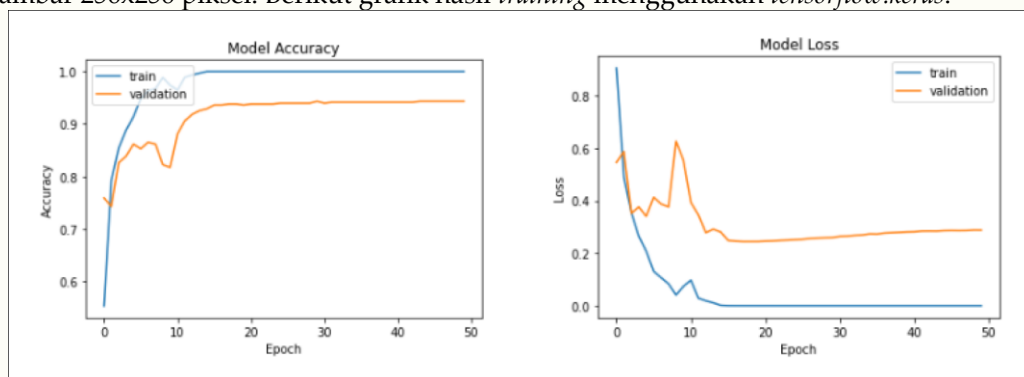
Tabel 3. Hasil Pengujian Hyperparameter

<i>Epoch</i>	<i>Optimizer</i>	<i>Batch size</i>	<i>Train Acc</i>	<i>Val Acc</i>	<i>Train Loss</i>	<i>Val Loss</i>
10	Adam	32	0.9982	0.9149	0.01	0.41
		64	0.9890	0.8670	0.05	0.34
	Adamax	32	0.9898	0.8918	0.04	0.28
		64	0.9479	0.8369	0.15	0.40
	RMSprop	32	0.9987	0.8901	0.01	0.40
		64	0.9608	0.8460	0.16	0.40
30	Adam	32	1.0000	0.8457	0.00	1.05
		64	1.0000	0.8564	0.00	0.87
	Adamax	32	1.0000	0.9060	0.00	0.52
		64	1.0000	0.8301	0.00	0.50
	RMSprop	32	1.0000	0.8901	0.00	0.80
		64	1.0000	0.8901	0.00	0.40
50	Adam	32	1.0000	0.9433	0.00	0.29
		64	1.0000	0.8599	0.00	1.11
	Adamax	32	1.0000	0.8457	0.00	1.24
		64	1.0000	0.8387	0.00	0.97
	RMSprop	32	1.0000	0.8493	0.00	1.86
		64	1.0000	0.8209	0.00	1.69

Berdasarkan hasil pengujian ini didapatkan bahwa kombinasi *hyperparameter* terbaik pada model CNN adalah dengan menggunakan jumlah *EPOCH* sebanyak 50, algoritma *optimizer* Adam, dan *batch size* sebanyak 32. Penggunaan kombinasi *hyperparameter* ini mampu memberikan performa terbaik pada model CNN yang dibangun.

Setelah melalui beberapa tahapan skenario dalam algoritma CNN dan pengujian *hyperparameter* untuk menentukan model terbaik, didapatkan hasil *training*. Proses *train* akan menghasilkan sebuah model yang selanjutnya akan digunakan untuk proses pengujian atau *testing*, model terbaik didapatkan

dengan skenario *hyperparameter* paling optimal yaitu jumlah *epoch* 50, *batch size* 32, optimasi Adam, *input* gambar 256x256 piksel. Berikut grafik hasil *training* menggunakan *tensorflow.keras*:



Gambar 3. Grafik Hasil Training

Selanjutnya berdasarkan hasil pelatihan model CNN terhadap *data training* didapatkan juga hasil *confusion matrix* pada data ras, berikut ini merupakan hasil dari *confusion matrix* sebagai berikut.

Tabel 4. Confusion Matrix

		Actual Values		
		Bercak Daun	Karat Daun	Daun Sehat
Predicted Values	Bercak Daun	176	1	11
	Karat Daun	1	185	2
	Daun Sehat	14	3	171

Tabel 4 menunjukkan hasil prediksi terhadap data *validation* selama proses *training*. Prediksi terhadap penyakit bercak daun diklasifikasikan benar sebanyak 170 dan *missing* data dari *input* bercak daun diklasifikasikan sebagai karat daun sebanyak 1 dan daun sehat sebanyak 11 data. Prediksi kedua untuk penyakit karat daun diklasifikasikan benar sebanyak 187 dan 1 salah klasifikasi atau *missing* data yang memprediksi sebagai bercak daun dan 2 data sebagai daun sehat. Terakhir prediksi terhadap daun sehat 171 terklasifikasi benar, 3 terklasifikasi sebagai karat daun dan 14 data citra terklasifikasi sebagai bercak daun. Perhitungan akurasi matrix tersebut adalah sebagai berikut.

$$\begin{aligned}
 \text{Overall Accuracy} &= \frac{TTP_{all}}{\text{Total Number of Testing Entries}} \times 100\% \\
 &= \frac{176 + 185 + 171}{564} \times 100\% = 94.3\%
 \end{aligned}$$

Dari perhitungan di atas didapatkan akurasi model dengan jumlah data *validation* 188 citra yaitu sebesar 100%. Pada penggunaan *confusion matrix* tidak hanya berpatokan pada seberapa besar atau kecil nilai akurasi, tetapi juga terdapat *precisions*, *recall* dan *F1-score*.


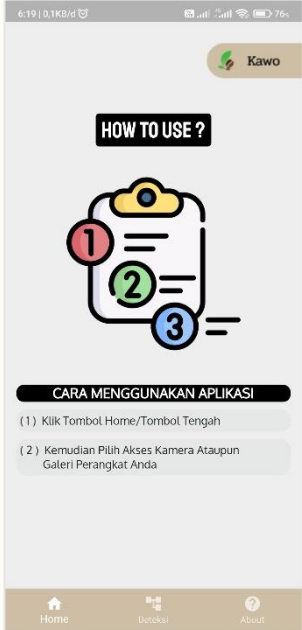
Berikut perhitungan untuk mendapatkan nilai *precisions*, *recall*, dan *F1-score* pada kelas 'karat_daun'.

$$\begin{aligned}
 \text{Precision} &= \frac{TP_{class}}{TP_{class} + \sum FP_{class}} = \frac{185}{185 + 1} \times 100\% = 99.46\% \\
 \text{recall} &= \frac{TTP_{class}}{TP_{class} + \sum FN_{class}} = \frac{185}{185 + 2} \times 100\% = 98.84\% \\
 f1 - score &= \frac{2x \sum (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} = \frac{2 \times (0.9946 \times 0.9894)}{(0.9946 + 0.9894)} \times 100\% = 99.2\%
 \end{aligned}$$


Hasil akurasi dari *precision* atau tingkat ketepatan antara data yang diminta dengan hasil yang diberikan oleh model yaitu sebesar 99.46%. Kemudian juga terdapat nilai *recall* yang merupakan tingkat keberhasilan model dalam menemukan kembali sebuah informasi adalah sebesar 98.84%. Terakhir terdapat akurasi *F1-score* yang merupakan tingkat kedekatan antara nilai prediksi dengan nilai aktual dengan nilai sebesar 99.2%.

Setelah model CNN dibangun, langkah berikutnya adalah menerapkan model CNN ke dalam aplikasi android menggunakan *framework* flutter dengan bahasa pemrograman dart. Aplikasi android untuk deteksi penyakit pada tanaman kopi arabika ini di beri nama "KAWO" yang memiliki 3 halaman, pertama ialah halaman *how to use* yang berisi deskripsi singkat terkait tata cara penggunaan aplikasi Kawo. Kedua terdapat halaman *about* yang berisi informasi kegunaan, nama dan versi aplikasi. Terakhir terdapat halaman inti dari aplikasi yaitu halaman deteksi penyakit. Berikut merupakan tampilan aplikasi Kawo.

Tabel 5. Tampilan Aplikasi Android KAWO

GUI	Deskripsi
	<p><i>Landing page</i> atau <i>splash screen</i> merupakan tampilan awal yang akan dijalankan pertama kali ketika aplikasi dibuka. Halaman ini berisi logo dan nama aplikasi yang merupakan identitas dari aplikasi deteksi penyakit pada tanaman kopi arabika ini.</p>
	<p><i>How to use page</i> atau <i>Halaman panduan</i> merupakan tampilan yang berisi panduan bagi user terkait bagaimana cara menggunakan aplikasi Kawo.</p>

GUI	Deskripsi
	<p>Halaman about berisi informasi kegunaan aplikasi dalam bentuk deskripsi singkat, selain itu juga tertera identitas aplikasi seperti logo aplikasi, nama aplikasi, dan versi aplikasi.</p>
	<p>Pada halaman deteksi digunakan untuk pengujian gambar. Dalam proses deteksi terdapat dua <i>button</i> sebagai pilihan pengujian yaitu <i>button</i> Ambil Gambar dan <i>button</i> Pilih Dari Galeri. <i>Button</i> ambil gambar digunakan untuk mengambil gambar atau citra daun kopi arabika secara langsung. Sedangkan, <i>button</i> pilih dari galeri digunakan untuk memilih gambar atau citra daun kopi arabika yang sebelumnya telah dipotret atau tersimpan dalam galeri penyimpanan.</p>

GUI	Deskripsi
	<p>Gambar ini menunjukkan hasil deteksi penyakit pada tanaman kopi arabika dengan hasil bahwa daun yang dipotret terdeteksi penyakit karat daun lengkap dengan penyebabnya yang ditampilkan dalam bentuk deskripsi singkat. Selain itu pada halaman hasil ini juga terdapat satu <i>button</i> lainnya yaitu <i>button reset</i>. <i>Button reset</i> digunakan untuk membersihkan halaman hasil deteksi dan kembali ke halaman deteksi awal.</p>

KESIMPULAN

Berdasarkan penelitian ini didapatkan hasil bahwa Penelitian ini menggunakan metode *Convolutional Neural Network* dan berhasil mencapai tingkat akurasi sebesar 100% pada proses *training* model. Proses *training* data dilakukan dengan mengatur beberapa *hyperparameter* dengan metode *trial and error*, dan didapatkan akurasi terbaik dengan *epoch* sebanyak 50, *batch size* 32 dengan menggunakan Optimizer Adam. Penelitian ini menggunakan pembagian dataset 80:20 dan berhasil menghasilkan nilai akurasi yang baik sebesar 94.3% pada data *validation*. Oleh karena itu, dapat disimpulkan bahwa model yang telah dibuat mampu melakukan identifikasi dan klasifikasi penyakit pada tanaman kopi arabika. Untuk penelitian selanjutnya, disarankan untuk menambah jumlah kelas penyakit pada tanaman kopi arabika agar lebih banyak jenis penyakit yang dapat diidentifikasi. Selain itu, penelitian juga dapat mempertimbangkan penambahan *hyperparameter* seperti variasi ukuran piksel, jumlah *epoch*, *learning rate*, *optimizer*, skenario variasi pembagian dataset dan faktor-faktor lainnya sebagai pembandingan untuk mencari arsitektur CNN yang menghasilkan model yang lebih optimal. Selanjutnya, penelitian dapat dikembangkan dalam hal implementasi atau penerapan langsung aplikasi android di perkebunan kopi agar lebih memudahkan dalam mengidentifikasi penyakit pada tanaman kopi arabika.

DAFTAR PUSTAKA

- Ahmad, Z., & Khan, N. (2021). CNN-Based Multistage Gated Average Fusion (MGAF) for Human Action Recognition Using Depth and Inertial Sensors. *IEEE Sensors Journal*, 21(3), 3623–3634. <https://doi.org/10.1109/JSEN.2020.3028561>
- Ashshiddieqy, M. H., Jondri, & Rizal, A. (2020). Klasifikasi Suara Paru Dengan Convolutional Neural Network (CNN). *EProceedings of Engineering*, 07(02), 8506–8512.
- Badan Pusat Statistik Nasional, 2022. Jumlah Produksi Kopi di Indonesia (2017-2021). Jakarta: Badan Pusat Statistik.
- Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. *ArXiv*, 1–31. <https://doi.org/https://doi.org/10.48550/arXiv.1603.07285>

- Gunawan, M. D., Franz, A., & Manullang, R. R. (2020). Sistem Pakar Penyakit Tanaman Kopi (*Coffea Sp*) Metode Forward Chaining Berbasis Web. *Buletin Poltanesa*, 21(1), 26–31. <https://doi.org/10.51967/tanesa.v21i1.321>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Neary, P. L. (2018). Automatic hyperparameter tuning in deep convolutional neural networks using asynchronous reinforcement learning. *Proceedings - 2018 IEEE International Conference on Cognitive Computing, ICC3 2018 - Part of the 2018 IEEE World Congress on Services*, (December), 73–77. <https://doi.org/10.1109/ICCC.2018.00017>
- Ng, W., Minasny, B., Montazerolghaem, M., Padarian, J., Ferguson, R., Bailey, S., & McBratney, A. B. (2019). Convolutional neural network for simultaneous prediction of several soil properties using visible/near-infrared, mid-infrared, and their combined spectra. *Geoderma*, 352, 251–267. <https://doi.org/10.1016/j.geoderma.2019.06.016>
- Perez, L., & Wang, J. (2017). *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. Retrieved from <http://arxiv.org/abs/1712.04621>
- Prastowo, B., Karmawati, E., Siswanto, R., & Munarso, C. I. S. J. (2010). *Budidaya dan Pasca Panen Kopi* (Yusniarti & A. Budiharto, eds.). Kementerian Pertanian.
- Putra, W. S. E. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *J. Tek. ITS*, 5.
- Putri, A. Y. P., & Sodik, A. (2019). Identifikasi Penyakit Tanaman Kopi Arabika dengan Metode K-Nearest Neighbor (K-NN). *Seminar Nasional Sains Dan Teknologi Terapan, VII*, 759–764.
- Ramadhan, M., Anwar, B., Gunawan, R., & Kustini, R. (2021). Pada Tanaman Kopi Menggunakan Metode. *Sistem Pakar Untuk Mendiagnosa Penyakit Pada Tanaman Kopi Menggunakan Metode Teorema Bayes*, 4307(June), 115–121.
- Rasyidi, M. A., & Bariyah, T. (2020). *Batik pattern recognition using convolutional neural network*. 9(4), 1430–1437. <https://doi.org/10.11591/eei.v9i4.2385>
- Sabrina, A. (2022). Klasifikasi Penyakit pada Tanaman Kopi Robusta Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *E-Proceeding of Engineering*, 9(3), 1919–1927.
- Singh, V., & Misra, A. K. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information Processing in Agriculture*, 4(1), 41–49. <https://doi.org/10.1016/j.inpa.2016.10.005>
- Walleign, S., Polceanu, M., Jemal, T., & Buche, C. (2019). Coffee grading with convolutional neural networks using small datasets with high variance. *Journal of WSCG*, 27(2), 113–120. <https://doi.org/10.24132/JWSCG.2019.27.2.4>
- Wang, G., Sun, Y., & Wang, J. (2017). Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning. *Hindawi Computational Intelligence and Neuroscience*, 2017, 8. <https://doi.org/https://doi.org/10.1155/2017/2917536>