

Optimasi Data Preprocessing dan Hyperparameter Tuning pada Klasifikasi Penyakit Daun Apel menggunakan DenseNet169

Gilang Satria Putra Ramadhan, Maimunah*, Setiya Nugroho

Fakultas Teknik, Program Studi Teknik Informatika, Universitas Muhammadiyah Magelang, Magelang, Indonesia

Email: ¹gilangsatriags2@gmail.com, ^{2,*}maimunah@unimma.ac.id, ³setiya@ummgl.ac.id

Email Penulis Korespondensi: maimunah@unimma.ac.id

Submitted: 27/10/2024; Accepted: 01/12/2024; Published: 03/12/2024

Abstrak—Apel merupakan salah satu komoditas hortikultura di Indonesia dengan produksi mencapai 5.235.955 kuintal pada tahun 2022, namun mengalami penurunan menjadi 3.925.628 kuintal pada tahun 2023. Salah satu penyebab penurunan ini adalah penyakit pada tanaman apel yang terjadi pada bagian daun, seperti *scab*, *black rot*, dan *cedar rust*, yang dapat mengakibatkan penurunan kualitas dan kuantitas hasil produksi. Oleh karena itu, diperlukan teknologi untuk klasifikasi penyakit pada daun apel yang cepat dan akurat. Penelitian ini menggunakan model CNN dengan DenseNet169 dengan optimasi pada *data preprocessing* dan *hyperparameter tuning* untuk meningkatkan akurasi pada model klasifikasi penyakit daun apel. Sebanyak 36 kombinasi skenario *data preprocessing* dan *hyperparameter tuning* diujicobakan pada dataset citra daun apel yang terdiri dari 4 kelas: *scab*, *black rot*, *cedar rust*, dan *healthy*. Skenario optimal diperoleh dari kombinasi RGB + CLAHE dengan *optimizer* RMSprop dan *learning rate* 0.0001 (P6 + H4), yang menghasilkan *accuracy* 99.39%, *precision* 99.4%, *recall* 99.39%, dan *f1-score* 99,39%. Hasil penelitian ini menunjukkan bahwa pemilihan *data preprocessing* dan *hyperparameter tuning* yang tepat sangat berpengaruh terhadap performa model klasifikasi penyakit daun apel.

Kata Kunci: CNN; DenseNet169; Penyakit Daun Apel; Praproses Data; Hyperparameter

Abstract—Apples are one of the horticultural commodities in Indonesia with production reaching 5,235,955 quintals in 2022, but decreasing to 3,925,628 quintals in 2023. One of the causes of this decline is diseases in apple plants that occur on the leaves, such as scab, black rot, and cedar rust, which can result in a decrease in the quality and quantity of production. Therefore, technology is needed for fast and accurate classification of diseases on apple leaves. This study uses a CNN model with DenseNet169 with optimization on data preprocessing and hyperparameter tuning to improve the accuracy of the apple leaf disease classification model. A total of 36 combinations of data preprocessing and hyperparameter tuning scenarios were tested on the apple leaf image dataset consisting of 4 classes: scab, black rot, cedar rust, and healthy. The optimal scenario is obtained from a combination of RGB + CLAHE with RMSprop optimizer and a learning rate of 0.0001 (P6 + H4), which results in 99.39% accuracy, 99.4% precision, 99.39% recall, and 99.39% f1-score. The results of this study show that the selection of the right preprocessing data and hyperparameter tuning greatly affects the performance of the apple leaf disease classification model.

Keywords: CNN; DenseNet169; Apple Leaf Disease; Data Preprocessing; Hyperparameter

1. PENDAHULUAN

Apel merupakan salah satu produk hortikultura dengan potensi besar yang dapat dikembangkan sebagai bahan utama dalam pembuatan makanan dan minuman di sektor pertanian [1]. Apel tidak hanya dikenal karena kandungan gizinya yang kaya akan serat, vitamin, dan antioksidan [2], tetapi juga karena fleksibilitasnya dalam berbagai produk makanan dan minuman. Menurut data Badan Pusat Statistik (BPS) tahun 2022, Indonesia memiliki produksi apel sebanyak 5.235.955 kuintal, di mana Jawa Timur menyumbang 5.233.263 kuintal dari total produksi apel sepanjang tahun tersebut [3]. Namun pada tahun 2023, produksi apel mengalami penurunan menjadi 3.925.628 kuintal. Salah satu penyebabnya adalah penyakit pada tanaman apel yang semakin bervariasi [4]. Penyakit seperti *scab*, *black rot* dan *cedar rust* yang terjadi pada daun apel dapat secara signifikan mengurangi kualitas dan jumlah hasil panen [5]. *Black rot*, yang disebabkan oleh jamur *Diplodia seriata*, ditandai dengan munculnya bintik-bintik seperti mata katak dengan tepi kemerahan atau ungu yang muncul pada daun yang terinfeksi. *Scab*, yang disebabkan oleh jamur *Venturia inaequalis*, dapat dikenali dengan munculnya bintik-bintik kuning pucat atau hijau zaitun di permukaan atas daun, serta lesi beludru berwarna gelap di permukaan bawahnya. Sementara itu, *Cedar rust*, yang disebabkan oleh jamur *Gymnosporangium juniperi-virginianae*, ditandai dengan munculnya lesi melingkar berwarna kemerahan atau kuning pucat dan secara bertahap membesar menjadi bintik-bintik oranye-kuning cerah [6].

Untuk mencegah, mengidentifikasi, dan mengatasi penyakit pada daun apel, penting meningkatkan pemahaman dalam mengenali tanaman apel yang sehat dan yang terinfeksi penyakit [7]. Penyakit ini biasanya diidentifikasi melalui bercak pada daun. Meski sudah era digital, petani masih mengandalkan deteksi manual dengan mata. Namun, metode ini memiliki kendala seperti kurangnya pemahaman, bersifat subjektif, dan memakan waktu [8]. Oleh karena itu, dibutuhkan teknologi untuk klasifikasi penyakit daun apel. Teknologi pengolahan citra digital menjadi solusi yang efektif, karena dapat mengekstrak informasi melalui proses digital [9]. Salah satu metode untuk melakukan pengolahan citra adalah *Convolutional Neural Network* (CNN) [10]. CNN dapat secara otomatis memproses fitur kompleks dalam citra dan membantu mesin mengenali serta membedakan citra dengan akurat [11]. CNN dibangun dengan beberapa lapisan yang terdiri dari: (1) *Input Layer*, (2) *Convolutional Layer*, (3) *ReLU (Rectified Linear Unit)*, (4) *Pooling Layer*, (5) *Fully Connected Layer*, (6) *Softmax Layer* untuk menjalankan tugas klasifikasi citra.

Transfer learning adalah teknik pada CNN yang memanfaatkan bobot dari pelatihan sebelumnya, sehingga mempercepat komputasi dan meningkatkan akurasi model [12]. DenseNet169 merupakan salah satu model *transfer learning* yang unggul. Dalam berbagai skenario klasifikasi COVID-19 menggunakan *X-ray*, DenseNet169 terbukti lebih baik dibandingkan *transfer learning* lainnya [13]. Selain itu, DenseNet memiliki keunggulan menarik lainnya, yaitu kemampuannya dalam mengurangi masalah *vanishing gradient*, memperkuat propagasi fitur, mendorong penggunaan kembali fitur yang sudah dipelajari, dan secara signifikan mengurangi jumlah parameter yang diperlukan [14].

Terdapat beberapa penelitian yang membahas klasifikasi penyakit daun apel menggunakan CNN atau *transfer learning*. Pada penelitian yang dilakukan [5] tentang deteksi hama pada daun apel dengan CNN dengan citra yang terdiri dari 4 kelas yaitu, *scab*, *black rot*, *cedar rust* dan *healthy* didapatkan akurasi 99.66%. Penelitian lainnya tentang daun apel yang dilakukan oleh [6] mengenai deteksi penyakit tanaman apel menggunakan citra daun melalui CNN mendapatkan akurasi sebesar 98%. Penelitian terkait DenseNet169 untuk deteksi COVID-19 menggunakan citra *chest X-ray* menunjukkan bahwa arsitektur ini mencapai akurasi 99.25% dan *f1-score* 99.24%, menjadikannya yang terbaik dibandingkan arsitektur CNN lainnya [13]. Selain itu penelitian yang dilakukan [15] terkait perbandingan *transfer learning* pada klasifikasi tumor otak, didapatkan hasil akurasi 0.98 dengan DenseNet169 dibanding dengan VGG16 0.75. Penelitian [16] tentang klasifikasi sampah dengan *ensemble* DenseNet169 mendapatkan akurasi 0.91 pada *balance dataset*.

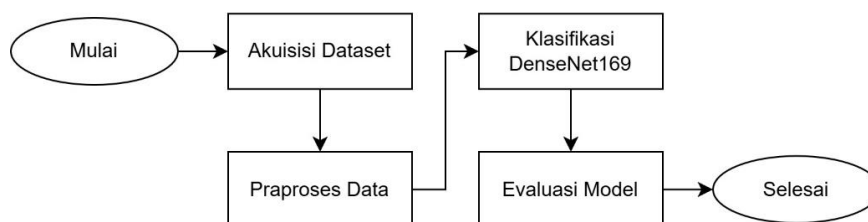
Sebelum model CNN di-*training*, biasanya dilakukan *data preprocessing* terhadap citra untuk mengatasi beberapa *noise* yang ada pada citra. Proses ini dapat mempengaruhi nilai akurasi dari model yang dibuat. Selain *data preprocessing*, *hyperparameter tuning* juga merupakan langkah penting dalam meningkatkan performa model, seperti pemilihan *optimizer* dan *learning rate* yang tepat [17]. Pemilihan *hyperparameter* yang tepat dapat meningkatkan kecepatan konvergensi dan akurasi model, serta mencegah masalah seperti *overfitting* dan *underfitting* [18].

Berdasarkan latar belakang di atas, peneliti melakukan studi yang berfokus pada klasifikasi jenis penyakit yang menyerang daun apel, yaitu *scab*, *black rot*, *cedar rust*, dan *healthy*, dengan memanfaatkan *transfer learning* DenseNet169. Untuk mencapai hasil optimal pada model, peneliti melakukan berbagai tahap *data preprocessing* dan *hyperparameter tuning* selama proses pelatihan model. Pengukuran kinerja menggunakan *confusion matrix* sebagai metode evaluasi untuk menganalisis tingkat *accuracy*, *precision*, *recall*, dan *f1-score* yang dihasilkan. Penelitian ini diharapkan menghasilkan model optimal yang dapat menjadi solusi praktis bagi petani dalam mengidentifikasi penyakit pada tanaman apel, sehingga memudahkan penerapan langkah penanganan yang tepat dan efektif.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian mengenai optimasi *data preprocessing* dan *hyperparameter tuning* pada klasifikasi penyakit daun apel menggunakan DenseNet169 melibatkan beberapa tahapan, sebagaimana pada Gambar 1.



Gambar 1. Tahapan Penelitian

Gambar 1 menunjukkan tahapan penelitian yang dilakukan. Tahap pertama adalah akuisisi *dataset*, dilanjutkan dengan praproses data atau *data preprocessing*, yang mencakup penanganan *imbalanced dataset*, perubahan ukuran citra, penyesuaian warna dan kontras, serta pembagian data menjadi *training*, *validation*, dan *testing*. Hasil dari tahap *data preprocessing* ini menghasilkan 6 skenario. Setelah itu, dilakukan *fine tuning* dan *hyperparameter tuning* untuk menemukan pengaturan optimal bagi model. Tahap ini menghasilkan 6 skenario yang nantinya dikombinasikan dengan skenario *data preprocessing*. Selanjutnya, data *training* digunakan untuk klasifikasi menggunakan metode CNN dengan *transfer learning* DenseNet169. Setelah itu, model terbaik hasil dari kombinasi skenario yang telah di-*training* dievaluasi untuk mengukur kinerjanya.

2.2 Akuisisi Dataset

Dalam penelitian ini, citra yang digunakan merupakan data sekunder mengenai penyakit pada tanaman apel, yang mencakup penggunaan citra bagian daun. *Dataset* ini terdiri dari 3171 citra dalam format JPG dengan ukuran 256x256 piksel dan menggunakan ruang warna RGB. Citra-citra ini diklasifikasikan ke dalam empat kelas yaitu *scab*, *black rot*, *cedar rust*, dan *healthy*. Sumber data citra berasal dari laman Kaggle.com [19].

2.3 Praproses Data

Proses klasifikasi diawali dengan tahap praproses data atau *data preprocessing* untuk mengurangi *noise* dan meningkatkan kualitas citra [20]. Langkah pertama dalam proses ini adalah menangani *imbalanced dataset* dengan *undersampling*. Pada teknik *undersampling*, jumlah data dari kelas mayoritas dikurangi sehingga seimbang dengan jumlah data di kelas minoritas [21]. *Undersampling* digunakan karena pada penelitian [22] skenario tersebut dikombinasikan dengan augmentasi menghasilkan performa yang baik dibandingkan dengan kombinasi *oversampling* augmentasi. Proses berikutnya melibatkan perubahan ukuran citra menjadi 224x224 piksel untuk meningkatkan kecepatan proses klasifikasi, seperti yang disarankan oleh [23]. Pembagian skenario yang akan diimplementasikan dalam penelitian ini disajikan pada Tabel 1, dengan skenario *data preprocessing* P1 hingga P6 mewakili tahapan skenario *data preprocessing* 1 hingga 6 secara berurutan.

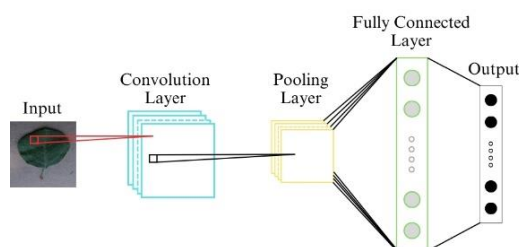
Tabel 1. Skenario *Data Preprocessing*

No	Skenario	Keterangan
1	P1	<i>Green Channel</i> + CLAHE + Unsharp Masking
2	P2	<i>Green Channel</i> + CLAHE + Gaussian Blur
3	P3	<i>Green Channel</i> + CLAHE
4	P4	RGB + CLAHE + Unsharp Masking
5	P5	RGB + CLAHE + Gaussian Blur
6	P6	RGB + CLAHE

Tahap selanjutnya dalam proses ini adalah mengambil *green channel* dari citra. *Green channel* dipilih karena memiliki tingkat *noise* yang lebih rendah serta menghasilkan citra yang lebih jelas dibandingkan dengan *red* dan *blue channel* [24]. Langkah selanjutnya adalah penerapan CLAHE (*Contrast Limited Adaptive Histogram Equalization*) pada semua skenario. CLAHE adalah teknik dalam *preprocessing* citra yang berfungsi untuk meningkatkan kontras pada citra [25] dengan cara memperbaiki distribusi intensitas piksel dalam citra. CLAHE dapat menangani masalah peningkatan kontras yang berlebihan dengan mengontrol atau membatasi nilai intensitas piksel pada histogram. Beberapa studi yang mengaplikasikan CLAHE sebagai metode untuk meningkatkan kualitas citra, sebagaimana ditunjukkan dalam literatur [26] menunjukkan bahwa CLAHE efektif dalam meningkatkan kualitas citra. Dalam skenario tersebut, *unsharp masking* juga diterapkan untuk meningkatkan ketajaman citra, dengan cara mengurangi citra asli dengan versi citra yang telah diburamkan (*unsharp*) [27]. Selain itu, teknik *gaussian blur* juga digunakan untuk mengurangi *noise* dan detail yang tidak diinginkan pada citra, menciptakan efek fokus otomatis, serta memberikan efek kabut [28]. *Dataset* citra kemudian dibagi menjadi tiga subset data *training*, *validation*, dan *testing*. Data *training* digunakan selama proses pelatihan model dan umumnya memiliki jumlah data yang lebih banyak daripada data *validation* dan *testing*. Untuk meningkatkan generalisasi dan variasi data, augmentasi diterapkan menggunakan teknik seperti *rescale*, *shear range*, *zoom range*, *vertical flip*, *horizontal flip*, *rotation*, dan *fill mode*.

2.3 Klasifikasi dengan DenseNet169

Setelah *data preprocessing* tahap selanjutnya adalah klasifikasi citra daun apel menggunakan model CNN dengan *transfer learning* DenseNet169.



Gambar 2. Arsitektur CNN

Gambar 2 menunjukkan arsitektur CNN sederhana dengan beberapa lapisan, dimulai dari lapisan pertama yaitu *convolution*. Lapisan ini berfungsi untuk mengekstraksi fitur-fitur dari citra *input* daun dengan menggeser filter (kernel) di seluruh citra. Hasil dari lapisan *convolution* kemudian diproses oleh *pooling layer*, di mana ukuran piksel citra diperkecil tanpa mengurangi kualitasnya. Lapisan selanjutnya adalah *fully connected*, yang mengambil fitur dari lapisan *convolution* dan *pooling* untuk menghasilkan keputusan akhir. Lapisan terakhir adalah *output*, yang memberikan hasil prediksi berupa klasifikasi kelas pada tugas pengenalan citra.

Penelitian ini menggunakan *transfer learning* DenseNet169 untuk meningkatkan performa model. Arsitektur ini memiliki keunggulan di mana setiap *layer* terhubung satu sama lain dalam pola *feed-forward*. Setiap *layer* menerima *feature maps* dari layer sebelumnya. Dengan cara ini, fitur tetap terjaga meskipun melalui banyak *layer*. Dalam arsitektur DenseNet, terjadi pengurangan jumlah parameter karena fungsi yang digunakan diulang [16].

Model DenseNet169 diinisialisasi dengan bobot yang sudah dilatih sebelumnya, kemudian dilakukan *fine-tuning* pada beberapa lapisan untuk menyesuaikan model dengan *dataset* citra daun apel. Setelah proses *fine-tuning*, langkah selanjutnya adalah melakukan *hyperparameter tuning*. Dalam penelitian ini, dilakukan 6 skenario *hyperparameter tuning* untuk memperoleh hasil yang optimal. Pembagian skenario *hyperparameter tuning* yang akan diimplementasikan dalam penelitian ini disajikan pada Tabel 2, dengan skenario H1 hingga H6 masing-masing mewakili kombinasi *optimizer* dan *learning rate* yang berbeda.

Tabel 2. Skenario *Hyperparameter Tuning*

No	Skenario	Optimizer	Learning Rate
1	H1	Adam	0.001
2	H2	Adam	0.0001
3	H3	RMSprop	0.001
4	H4	RMSprop	0.0001
5	H5	SGD	0.001
6	H6	SGD	0.0001

Setiap skenario menggunakan kombinasi *optimizer* dan nilai *learning rate* yang bervariasi untuk menentukan konfigurasi model terbaik. *Optimizer* yang digunakan terdiri dari Adam, RMSprop, dan SGD, yang masing-masing diatur pada dua nilai *learning rate* berbeda yaitu 0.001 dan 0.0001.

2.4 Evaluasi Model

Dalam penelitian ini, evaluasi model yang dibuat dilakukan menggunakan *confusion matrix*. *Confusion matrix* adalah matriks yang menunjukkan performa suatu model atau algoritma, di mana setiap baris merepresentasikan kelas sebenarnya dan setiap kolom merepresentasikan kelas yang diprediksi atau dapat sebaliknya [29]. Tabel 3 menunjukkan *confusion matrix* yang digunakan dalam penelitian ini.

Tabel 3. *Confusion Matrix*

<i>Confusion matrix</i>	Kelas Prediksi		
	Positive	Negative	
Kelas Asli	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Pada Tabel 3, *confusion matrix* memiliki empat komponen utama yang digunakan untuk mengevaluasi performa model klasifikasi, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). *True Positive* merepresentasikan jumlah data positif yang diklasifikasikan dengan benar. *True Negative* adalah jumlah data negatif yang diklasifikasikan dengan benar. *False Positive* menunjukkan jumlah data negatif yang diklasifikasikan sebagai positif, sedangkan *False Negative* merepresentasikan jumlah data positif yang diklasifikasikan sebagai negatif.

Dari *confusion matrix*, nilai *accuracy*, *precision*, *recall*, dan *f1-score* dapat dihitung menggunakan rumus (1), (2), (3), dan (4).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$





$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{4}$$

3. HASIL DAN PEMBAHASAN

3.1 Hasil Akuisisi Dataset

Tahapan awal dari penelitian ini adalah akuisisi *dataset* citra daun apel yang diambil dari *open data* Kaggle. Hasil dari pengumpulan citra terbagi menjadi 4 kelas, yaitu *scab*, *black rot*, *cedar rust* dan *healthy* seperti disajikan pada Tabel 4.

Tabel 4. Rincian *Dataset*

No	Kelas	Citra	Jumlah Citra
1	<i>Scab</i>		630
2	<i>Black Rot</i>		621
3	<i>Cedar Rust</i>		275
4	<i>Healthy</i>		1645
Total			3171

Tabel 4 menyajikan rincian *dataset* yang digunakan dalam penelitian ini. *Dataset* terdiri dari 3171 citra yang terbagi dalam empat kelas. Kelas *scab* memiliki 630 citra, *black rot* memiliki 621 citra, *cedar rust* memiliki 275 citra, dan *healthy* memiliki 1645 citra.

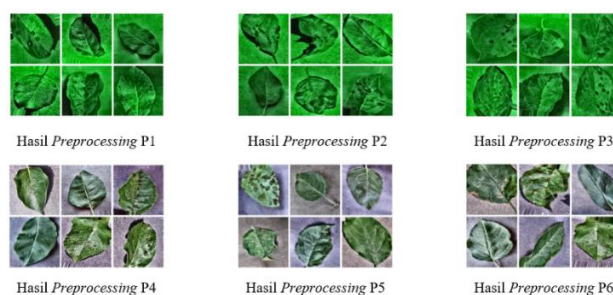
3.2 Hasil Praproses Data

Langkah pertama dalam praproses data atau *data preprocessing* adalah menangani masalah *imbalanced dataset* dengan menerapkan *undersampling* pada kelas mayoritas. Proses ini mengurangi jumlah citra pada kelas mayoritas agar seimbang dengan kelas minoritas, sehingga setiap kelas memiliki jumlah data yang sebanding. Hal ini bertujuan mencegah bias model terhadap kelas mayoritas dan memastikan representasi yang seimbang selama *training*. Rincian *dataset* hasil *undersampling* disajikan pada Tabel 5.

Tabel 5. Rincian *Dataset* Setelah *Undersampling*

Kelas	Jumlah Citra
<i>Scab</i>	275
<i>Black Rot</i>	275
<i>Cedar Rust</i>	275
<i>Healthy</i>	275
Total	1100

Tabel 5 menyajikan hasil *undersampling*, di mana setiap kelas memiliki jumlah citra yang disamakan menjadi 275 citra. Ukuran citra kemudian diubah menjadi 224x224 piksel dan diproses sesuai dengan 6 skenario *data preprocessing* yang telah dibuat sebelumnya. Tahap selanjutnya data dibagi menjadi 3, yaitu data *training*, *validation*, dan *testing*. Data *training* terdiri dari 70% dengan total 770 citra, data *validation* terdiri dari 15% dengan total 165 citra, dan data *testing* terdiri dari 15% dengan total 165 citra. Augmentasi diterapkan untuk meningkatkan variasi data dengan teknik *rescale*, *shear range*, *zoom range*, *vertical flip*, *horizontal flip*, *rotation*, dan *fill mode*. Berikut merupakan hasil dari tahap *data preprocessing* ditampilkan pada Gambar 3.



Gambar 3. Hasil *Data Preprocessing*

Gambar 3 menunjukkan hasil *preprocessing* skenario P1 setelah *data preprocessing* dan augmentasi. Hasil untuk P2 hingga P6 juga mewakili *output data preprocessing* dan augmentasi sesuai teknik masing-masing skenario.

3.3 Hasil Klasifikasi DenseNet169

Fine-tuning dilakukan untuk menyesuaikan model dengan *dataset* yang digunakan sebelum klasifikasi. Pada tahap awal *fine-tuning*, semua lapisan dari *base model* DenseNet169 dibekukan untuk mencegah *overfitting* akibat

penggunaan parameter berlebihan. Tahap berikutnya, beberapa lapisan baru ditambahkan, termasuk Conv2D untuk ekstraksi fitur, MaxPooling2D untuk mengurangi dimensi fitur, Dropout untuk mengurangi risiko *overfitting*, serta lapisan Dense untuk menggabungkan fitur. Sebagai lapisan akhir, *fully connected layer* ditambahkan dengan jumlah neuron sesuai dengan jumlah kelas. Hasil *fine-tuning* arsitektur DenseNet169 ditampilkan dalam Tabel 6.

Tabel 6. Arsitektur DenseNet169 Setelah *Fine-tuning*

Layer	Output shape	Param
densenet169 (Functional)	(None, 7, 7, 1664)	12642880
conv2d (Conv2D)	(None, 4, 4, 32)	479264
max_pooling2d (MaxPooling2D)	(None, 2, 2, 32)	0
dropout (Dropout)	(None, 2, 2, 32)	0
FC2 (Dense)	(None, 2, 2, 64)	2112
dropout_1 (Dropout)	(None, 2, 2, 64)	0
flatten (Flatten)	(None, 256)	0
dropout_2 (Dropout)	(None, 256)	0
dense (Dense)	(None, 4)	1028
Total params: 13,125,284		
Trainable params: 482,404		
Non-trainable params: 12,642,880		

Setelah proses *fine-tuning*, dilakukan *training* model untuk melihat performanya sebelum optimasi dengan *data preprocessing* dan *hyperparameter tuning*. Hasil akurasi dari model yang dilatih tanpa optimasi ditampilkan pada Tabel 7.

Tabel 7. Hasil Akurasi Sebelum Optimasi

Akurasi		
<i>Training</i>	<i>Validation</i>	<i>Testing</i>
0.8844	0.8424	0.8424

Pada proses *training* awal, diterapkan *batch size* sebesar 32 dan jumlah *epoch* sebanyak 30. Dari hasil yang ditunjukkan pada Tabel 7, terlihat bahwa akurasi model sebelum optimasi hanya 0.8424. Hasil ini kurang optimal dan menunjukkan bahwa model belum sepenuhnya mampu generalisasi dengan baik pada data yang tidak terlihat sebelumnya. Oleh karena itu, langkah selanjutnya adalah menerapkan proses optimasi untuk meningkatkan performa model. *Training* model dilakukan menggunakan arsitektur DenseNet169 yang telah di *fine-tuning* dengan pengaturan *batch size* dan *epoch* yang sama, serta dioptimalkan melalui kombinasi skenario *data preprocessing* dan *hyperparameter tuning*. Semua hasil skenario percobaan yang diterapkan ditampilkan pada Tabel 8.

Tabel 8. Hasil Akurasi Seluruh Kombinasi Skenario

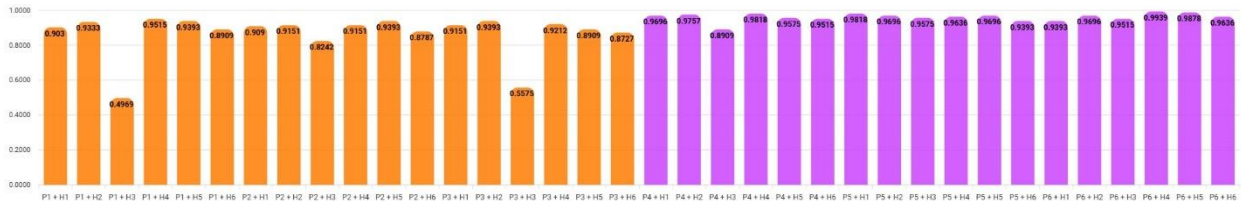
No	Skenario	Akurasi			No	Skenario	Akurasi		
		<i>Training</i>	<i>Validation</i>	<i>Testing</i>			<i>Training</i>	<i>Validation</i>	<i>Testing</i>
1	P1 + H1	0.9169	0.9152	0.9030	19	P4 + H1	0.9623	0.9758	0.9696
2	P1 + H2	0.8701	0.9273	0.9333	20	P4 + H2	0.9468	0.9758	0.9757
3	P1 + H3	0.4675	0.5212	0.4969	21	P4 + H3	0.8623	0.9091	0.8909
4	P1 + H4	0.9675	0.9091	0.9515	22	P4 + H4	0.9857	0.9879	0.9818
5	P1 + H5	0.8481	0.9455	0.9393	23	P4 + H5	0.9727	0.9636	0.9575
6	P1 + H6	0.7351	0.8788	0.8909	24	P4 + H6	0.8961	0.9455	0.9515
7	P2 + H1	0.9571	0.9576	0.9090	25	P5 + H1	0.9571	0.9758	0.9818
8	P2 + H2	0.8831	0.9091	0.9151	26	P5 + H2	0.9442	0.9758	0.9696
9	P2 + H3	0.7792	0.8424	0.8242	27	P5 + H3	0.9195	0.9818	0.9575
10	P2 + H4	0.9390	0.9212	0.9151	28	P5 + H4	0.9779	0.9879	0.9878
11	P2 + H5	0.9026	0.8970	0.9393	29	P5 + H5	0.9610	0.9939	0.9696
12	P2 + H6	0.7896	0.8667	0.8787	30	P5 + H6	0.8558	0.9515	0.9393
13	P3 + H1	0.9052	0.9212	0.9151	31	P6 + H1	0.9649	0.9636	0.9393
14	P3 + H2	0.8779	0.8909	0.9393	32	P6 + H2	0.9208	0.9879	0.9696
15	P3 + H3	0.4532	0.5030	0.5575	33	P6 + H3	0.9442	0.9636	0.9515
16	P3 + H4	0.9429	0.9394	0.9212	34	P6 + H4	0.9844	0.9818	0.9939
17	P3 + H5	0.8870	0.9273	0.8909	35	P6 + H5	0.9714	0.9818	0.9878
18	P3 + H6	0.7792	0.9091	0.8727	36	P6 + H6	0.8675	0.9636	0.9636

Tabel 8 menunjukkan data hasil akurasi model klasifikasi penyakit daun apel pada berbagai kombinasi skenario *data preprocessing* dan *hyperparameter tuning*. Secara keseluruhan, terdapat 36 skenario yang diuji, dan hasil akurasi yang

diperoleh bervariasi di antara skenario-skenario tersebut. Perbedaan akurasi tersebut dipengaruhi oleh kombinasi *data preprocessing* dan *hyperparameter tuning* yang diterapkan.

3.3.1 Pengaruh *Data Preprocessing* Terhadap Akurasi

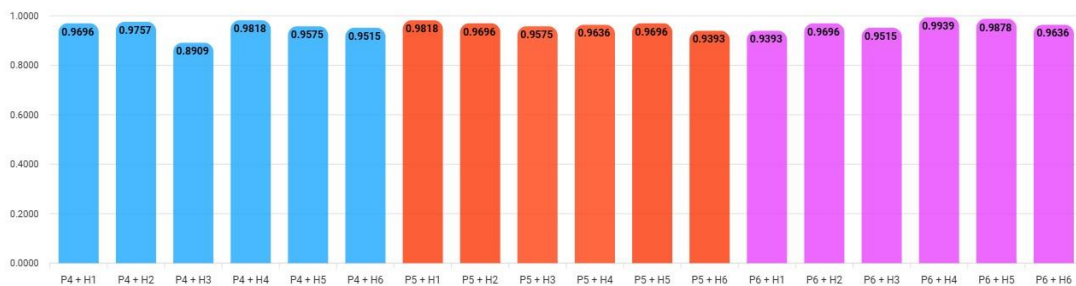
Hasil akurasi dari semua kombinasi skenario percobaan dalam Tabel 8 kemudian divisualisasikan dengan menggunakan diagram batang yang disajikan pada Gambar 4.



Gambar 4. Perbandingan *Green Channel* dan RGB

Pada Gambar 4 menunjukkan hasil akurasi dari seluruh skenario yang diuji menggunakan data *testing* sebesar 15% dari total data. Data tersebut dikelompokkan berdasarkan skenario *channel* warna pada citra *input*. Pada diagram tersebut, batang berwarna oranye mewakili skenario di mana citra *input* menggunakan *channel* warna hijau (*green channel*) (P1, P2, P3), sedangkan batang berwarna ungu mewakili skenario dengan citra *input* yang menggunakan *channel* warna lengkap (RGB) (P4, P5, P6). Semua skenario tersebut dikombinasikan dengan *hyperparameter tuning* H1 - H6.

Pada skenario *green channel* mendapatkan rata-rata akurasi sebesar 0.8662 dengan akurasi terendahnya 0.4969 dan akurasi tertinggi 0.9515. Sementara itu, pada skenario RGB mendapatkan rata-rata akurasi sebesar 0.9618 dengan akurasi terendahnya 0.8909 dan akurasi tertingginya mencapai 0.9939. Dalam visualisasi diagram, skenario RGB juga menunjukkan akurasi yang stabil dibandingkan dengan *green channel*, selain itu perbedaan rata-rata akurasi antara dua skenario tersebut juga sangat signifikan. Hasil ini menunjukkan bahwa skenario dengan *channel* warna RGB secara signifikan meningkatkan akurasi dan stabilitas performa model baik dikombinasikan dengan skenario *hyperparameter tuning* H1 - H6.



Gambar 5. Perbandingan Unsharp Masking, Gaussian Blur, dan Tanpa Filter

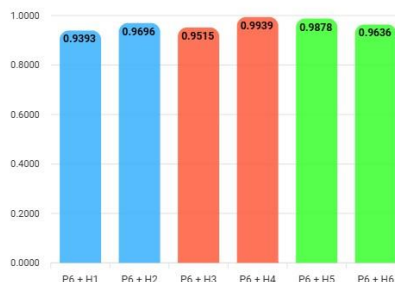
Selain itu studi ini juga melakukan perbandingan terkait tingkat akurasi berdasarkan penerapan filter dalam skenario yang berbeda. Pada perbandingan filter digunakan skenario *channel* warna RGB saja dikarenakan lebih unggul dibandingkan dengan *green channel*. Pada Gambar 5 menunjukkan diagram distribusi perbandingan skenario filter. Batang diagram berwarna biru adalah skenario yang menggunakan filter penajaman dengan unsharp masking (P4), batang berwarna merah merupakan skenario yang menggunakan filter penghalusan dengan gaussian blur (P5), dan batang berwarna ungu mewakili skenario yang tidak menggunakan filter (P6).

Pada skenario yang menggunakan unsharp masking, diperoleh rata-rata akurasi sebesar 0.9545, dengan akurasi terendah 0.8909 dan tertinggi 0.9818. Skenario dengan filter gaussian blur menghasilkan rata-rata akurasi sebesar 0.9635, dengan akurasi terendah 0.9393 dan tertinggi 0.9818. Sementara itu, skenario tanpa penggunaan filter mendapatkan rata-rata akurasi sebesar 0.9676, dengan akurasi terendah 0.9393 dan tertinggi 0.9939.

Dengan perbandingan ini, skenario tanpa filter memiliki rata-rata akurasi tertinggi, yaitu sebesar 0.9676, diikuti oleh skenario dengan filter gaussian blur yang memiliki rata-rata akurasi 0.9635, dan skenario menggunakan filter unsharp masking dengan rata-rata akurasi 0.9545. Skenario tanpa filter juga menunjukkan akurasi tertinggi yang lebih baik, yaitu 0.9939, dibandingkan dengan skenario gaussian blur yang hanya mencapai 0.9818. Hal ini menunjukkan bahwa meskipun perbedaan rata-rata akurasi antara skenario tanpa filter dan gaussian blur tidak terlalu signifikan, skenario tanpa filter lebih unggul dalam mencapai akurasi maksimal. Hal ini dikarenakan penggunaan filter gaussian blur dapat mengurangi beberapa detail penting pada citra, yang berpengaruh pada kemampuan model dalam mengenali pola.

3.3.2 Pengaruh *Hyperparameter Tuning* Terhadap Akurasi

Penelitian ini juga melakukan perbandingan performa pada skenario *hyperparameter tuning* untuk mencapai hasil terbaik. Skenario yang digunakan dalam perbandingan ini adalah skenario dengan menggunakan RGB + CLAHE tanpa filter (P6), karena menunjukkan akurasi yang lebih unggul dibandingkan skenario *data preprocessing* lainnya saat dikombinasikan dengan H1 - H6. Hasil visualisasi perbandingan ditampilkan pada Gambar 6.

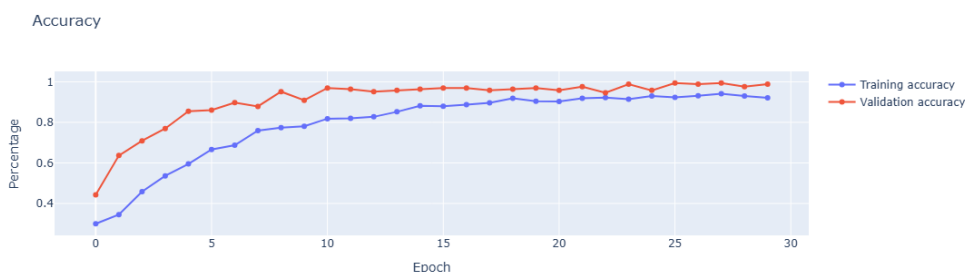


Gambar 6. Perbandingan *Optimizer* dan *Learning Rate*

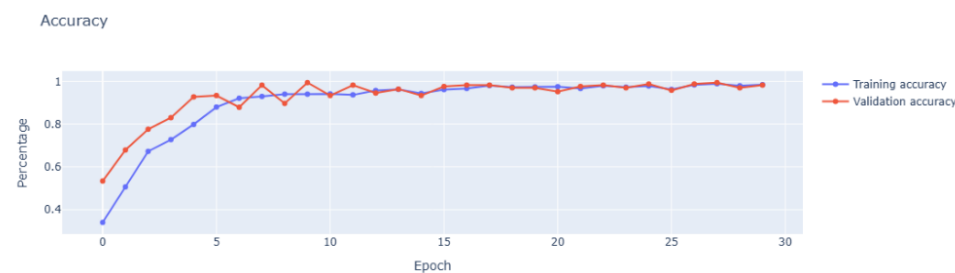
Pada Gambar 6, data tersebut dikelompokkan berdasarkan skenario *optimizer* dalam *hyperparameter tuning*. Batang berwarna biru menunjukkan skenario dengan *optimizer* Adam dan *learning rate* 0.001 serta 0.0001 (H1, H2). Batang berwarna merah menunjukkan *optimizer* RMSprop dengan *learning rate* 0.001 dan 0.0001 (H3, H4), sedangkan batang berwarna hijau menunjukkan *optimizer* SGD dengan *learning rate* 0.001 dan 0.0001 (H5, H6).

Dari visualisasi diagram batang tersebut, penggunaan *optimizer* Adam dengan *learning rate* 0.001 (H1) mendapatkan akurasi sebesar 0.9393, sedangkan dengan *learning rate* 0.0001 akurasi meningkat menjadi 0.9696. Ini berarti pengurangan *learning rate* pada Adam membantu model mencapai akurasi yang lebih baik. Pada penggunaan *optimizer* RMSprop dengan *learning rate* 0.001 memberikan akurasi 0.9515, yang lebih tinggi dari Adam pada *learning rate* yang sama. Saat *learning rate* diturunkan menjadi 0.0001, akurasi meningkat menjadi 0.9939, yang merupakan akurasi tertinggi dari semua skenario. *Learning rate* yang lebih rendah memungkinkan RMSprop untuk membuat penyesuaian lebih baik pada pembobotan model. Untuk *optimizer* SGD menggunakan *learning rate* 0.001 memberikan hasil yang cukup baik, yaitu 0.9878, menunjukkan bahwa SGD dapat memberikan performa yang hampir optimal pada *learning rate* ini. Saat *learning rate* dikurangi menjadi 0.0001, akurasi sedikit menurun menjadi 0.9636. Ini menunjukkan bahwa SGD tidak merespons sebaik Adam dan RMSprop terhadap *learning rate* yang lebih rendah, serta membutuhkan waktu yang lebih lama untuk konvergen dengan *learning rate* yang lebih kecil.

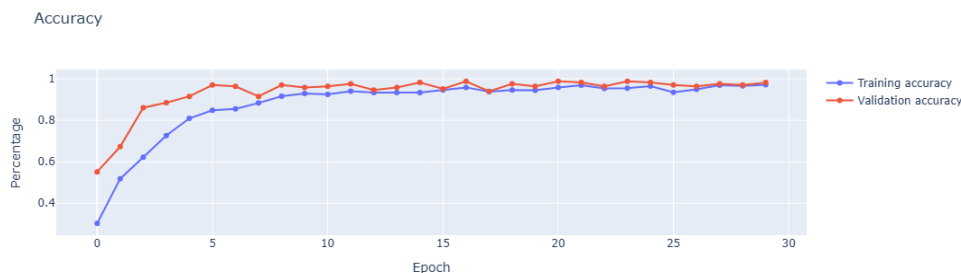
Berdasarkan hasil akurasi, RMSprop dengan *learning rate* 0.0001 (H4) adalah kombinasi *hyperparameter tuning* terbaik, dengan akurasi 0.9939, menjadikannya juga akurasi tertinggi dari seluruh kombinasi skenario. Ini menunjukkan bahwa RMSprop lebih optimal untuk skenario ini (RGB + CLAHE), terutama dengan pengaturan *learning rate* yang lebih rendah. Penggunaan *learning rate* yang lebih rendah (0.0001) memberikan hasil yang lebih baik dibandingkan *learning rate* yang lebih tinggi (0.001), khususnya pada *optimizer* seperti Adam dan RMSprop. Pengaturan *learning rate* rendah membantu model dalam mencapai konvergensi yang lebih baik dan stabilitas dalam pelatihan.



Gambar 7. Grafik Skenario P6 + H2



Gambar 8. Grafik Skenario P6 + H4

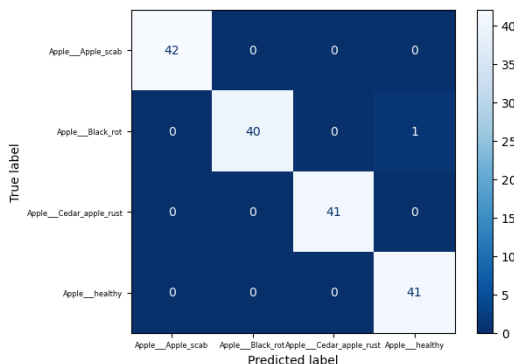


Gambar 9. Grafik Skenario P6 + H5

Gambar 7, Gambar 8, dan Gambar 9 menunjukkan grafik proses *training* dan *validation* model untuk skenario P6 + H2, P6 + H4, dan P6 + H5, yang masing-masing mencatat akurasi tertinggi untuk setiap *optimizer* yang digunakan. Dari ketiga skenario tersebut, Gambar 8, yang menggambarkan P6 + H4, menunjukkan performa terbaik. Pada *epoch* ke-22, nilai akurasi *training* dan *validation* telah mencapai kesetaraan atau *converge*, menunjukkan bahwa model telah mencapai titik optimal dan tidak mengalami peningkatan yang signifikan lagi. Di sisi lain, untuk skenario P6 + H2 (Gambar 7) dan P6 + H5 (Gambar 9), meskipun akurasi yang dihasilkan cukup mendekati P6 + H4, kedua skenario ini belum mencapai kondisi *converge*. Grafik menunjukkan adanya perbedaan yang jelas antara akurasi *training* dan *validation*, yang mengindikasikan bahwa model masih memiliki potensi untuk terus meningkat, atau bahkan rentan terhadap *overfitting* jika proses pelatihan diteruskan lebih lama.

3.4 Hasil Evaluasi Model

Untuk menganalisis kinerja model terbaik yang akan diterapkan (P6 + H4), dilakukan evaluasi dengan menggunakan *confusion matrix* pada data *testing*. Model dievaluasi untuk mendapatkan nilai *accuracy*, *precision*, *recall*, dan *f1-score*. Hasil dari *confusion matrix* disajikan pada Gambar 10.



Gambar 10. Confusion Matrix Hasil Evaluasi Model

Confusion matrix pada Gambar 10 menunjukkan kinerja model dalam mengklasifikasikan empat kelas penyakit daun apel yaitu *scab*, *black rot*, *cedar rust*, dan *healthy*. Pada kelas *scab* model berhasil mengklasifikasikan semua 42 sampel citra dengan benar tanpa ada kesalahan. Untuk kelas *black rot*, model mengklasifikasikan 40 sampel citra dengan benar, namun terdapat satu sampel yang salah diklasifikasikan sebagai *healthy*. Pada kelas *cedar rust*, semua 41 sampel berhasil diklasifikasikan dengan tepat untuk semua 41 sampel. Kelas *healthy* semua 41 sampel juga berhasil diklasifikasikan dengan benar semua. Berdasarkan hasil dari *confusion matrix*, diperoleh nilai-nilai evaluasi model seperti *accuracy*, *precision*, *recall*, dan *f1-score*, yang ditampilkan pada Tabel 9.

Tabel 9. Hasil Evaluasi Model

<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
0.9939	0.994	0.9939	0.9939

Hasil evaluasi model menunjukkan bahwa *accuracy* model P6 + H4 mencapai 99.39%, dengan *precision* dan *recall* masing-masing 99.4% dan 99.39%, serta *f1-score* sebesar 99.39%. Jika dibandingkan dengan hasil akurasi model sebelum optimasi, di mana akurasi hanya mencapai 0.8424 atau 84.24%, peningkatan ini menunjukkan bahwa optimasi yang diterapkan berhasil secara signifikan meningkatkan performa model. Ini menunjukkan bahwa model sangat akurat dalam mengklasifikasikan citra daun apel, dengan keseimbangan yang baik antara kemampuan mengenali dan mengklasifikasikan setiap kelas penyakit dengan tepat.

4. KESIMPULAN

Penelitian ini berfokus pada optimasi model klasifikasi penyakit daun apel dengan menerapkan berbagai skenario *data preprocessing* dan *hyperparameter tuning*. Tujuan utama dari penelitian ini adalah untuk menemukan kombinasi terbaik antara teknik *data preprocessing* dan *hyperparameter tuning*. Dalam penelitian ini, model CNN digunakan untuk pelatihan dengan metode *transfer learning* DenseNet169. Hasil percobaan menunjukkan bahwa pemilihan *data preprocessing* dan *hyperparameter tuning* memiliki dampak signifikan terhadap akurasi model. Model hanya mencapai akurasi sebesar 0.8424 atau 84.24% sebelum dilakukan optimasi, yang menunjukkan bahwa kinerjanya belum optimal. Dari total 36 kombinasi skenario untuk optimasi, skenario P6 + H4, yang menggabungkan RGB + CLAHE dengan *optimizer* RMSprop dan *learning rate* 0.0001, menunjukkan hasil optimal dengan *accuracy* 99.39%, *precision* 99.4%, *recall* 99.39%, dan *f1-score* 99.39%. Selain itu, skenario P6 + H4 juga berhasil mencapai kondisi *converge* selama proses *training*. Hal ini mengindikasikan bahwa model mampu mengklasifikasikan data uji dengan akurat sesuai dengan kelas yang relevan selama evaluasi. Di sisi lain, meskipun hasil yang diperoleh sudah sangat baik, ada beberapa bagian yang masih dapat ditingkatkan untuk penelitian selanjutnya. Penelitian selanjutnya dapat mengeksplorasi variasi lain dalam teknik *preprocessing* yang dapat membantu model untuk lebih optimal. Selain itu, eksperimen dengan arsitektur model lain atau teknik *ensemble* dapat dijadikan opsi untuk meningkatkan akurasi lebih lanjut.

REFERENCES

- [1] R. Prastiyaningtiyas and N. I. Hidayati, "Analisis Kelayakan Usaha Minuman Sari Buah Apel Di Desa Wonosari Nongkojajar Pasuruan," *J. Ekon. Manaj. dan Akunt.*, vol. 2, no. 4, pp. 680–690, 2023, doi: 10.572349/mufakat.v2i4.976.
- [2] M. N. Naben and E. Pani, "Pengaruh Berbagai Konsentrasi Gula terhadap Aktivitas Antioksidan pada Sari Buah Apel (Malus Sylvestris)," *J. Pendidik. Tambusai*, vol. 7, no. 3, pp. 20410–20414, 2023, doi: 10.31004/jptam.v7i3.9495.
- [3] BPS, "Produksi Buah-buahan dan Sayuran Menurut Jenis Tanaman Menurut Provinsi, 2023," www.bps.go.id, 2024. <https://www.bps.go.id/id/statistics-table/3/U0dKc1owczVSa1J5VFdOMWVETnlVRVJ6YIRJMFp6MDkjMw==/produksi-buah-buahan-menurut-jenis-tanaman-menurut-provinsi--2023.html?year=2023>
- [4] L. R. Farida, Tazkia; Susilowati, Dwi; Maula, "Fenomena Peralihan Usahatani Apel Ke Komoditas Lain Di Kecamatan Bumiaji Kota Batu," *J. Sos. Ekon. Pertan. dan Agribisnis*, vol. 1, no. 12, pp. 2439–2450, 2023.
- [5] D. Husen, K. Kusri, and K. Kusnawi, "Deteksi Hama Pada Daun Apel Menggunakan Algoritma Convolutional Neural Network," *J. Media Inform. Budidarma*, vol. 6, no. 4, pp. 2103–2110, 2022, doi: 10.30865/mib.v6i4.4667.
- [6] V. K. Vishnoi, K. Kumar, B. Kumar, S. Mohan, and A. A. Khan, "Detection of Apple Plant Diseases Using Leaf Images Through Convolutional Neural Network," *IEEE Access*, vol. 11, no. November 2022, pp. 6594–6609, 2023, doi: 10.1109/ACCESS.2022.3232917.
- [7] A. M. Lesmana, R. P. Fadhillah, and C. Rozikin, "Identifikasi Penyakit pada Citra Daun Kentang Menggunakan Convolutional Neural Network (CNN)," *J. Sains dan Inform.*, vol. 8, no. 1, pp. 21–30, 2022, doi: 10.34128/jsi.v8i1.377.
- [8] Q. N. Azizah and Andreyestha, "Klasifikasi Penyakit Daun Apel Menggunakan Convolutional Neural Network," *J. Ilmu Komput. dan Inform.*, vol. 3, no. 1, pp. 242–249, 2020, doi: 10.24127/jiki.v3i1.2379.
- [9] H. P. Hadi and E. H. Rachmawanto, "Ekstraksi Fitur Warna Dan Gcm Pada Algoritma Knn Untuk Klasifikasi Kematangan Rambutan," *J. Inform. Polinema*, vol. 8, no. 3, pp. 63–68, 2022, doi: 10.33795/jip.v8i3.949.
- [10] M. Wahid Islahfari, A. Lawi, and A. M. A. Siddik, "Perbandingan Kinerja Model Ensembled Transfer Learning Pada Klasifikasi Penyakit Daun Tomat," *Semin. Nas. Tek. Elektro dan Inform.*, vol. 8, no. 1, pp. 286–291, 2022.
- [11] S. Sahibu and I. Taufik, "Implementation of the Convolutional Neural Network Algorithm for Classifying Types of Organic and Non-Organic Waste," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 3, pp. 840–852, 2024, doi: 10.57152/malcom.v4i3.1346.
- [12] A. P. Syahputra, A. C. Siregar, and R. W. S. Insani, "Comparison of CNN Models With Transfer Learning in the Classification of Insect Pests," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 17, no. 1, pp. 103–114, 2023, doi: 10.22146/ijccs.80956.
- [13] F. A. Breve, "COVID-19 detection on Chest X-ray images: A comparison of CNN architectures and ensembles," *Expert Syst. Appl.*, vol. 204, 2022, doi: 10.1016/j.eswa.2022.117549.
- [14] A. Solihin, D. I. Mulyana, and M. B. Yel, "Klasifikasi Jenis Alat Musik Tradisional Papua menggunakan Metode Transfer Learning dan Data Augmentasi," *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 5, no. 2, pp. 36–44, 2022, doi: 10.47970/siskom-kb.v5i2.279.
- [15] N. Pratama, M. Liebenlito, and Y. Irene, "Perbandingan Model Klasifikasi Transfer Learning Convolutional Neural Network Tumor Otak menggunakan Citra Magnetic Resonance Imaging," *J. Sehat Indones.*, vol. 6, no. 01, pp. 308–318, 2024, doi: 10.59141/jsi.v6i01.81.
- [16] Ulfah Nur Oktaviana and Yufis Azhar, "Garbage Classification Using Ensemble DenseNet169," *J. RESTI (Rekayasa Sist. dan Teknol. Informatika)*, vol. 5, no. 6, pp. 1207–1215, 2021, doi: 10.29207/resti.v5i6.3673.
- [17] K. R. R. Wardani, H. Suryalim, V. J. L. Engel, and H. Christian, "Analisis Pemilihan Optimizer dalam Arsitektur Convolution Neural Network VGG16 dan Inception untuk Sistem Pengenalan Wajah," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 2, pp. 186–194, 2023, doi: 10.26418/jp.v9i2.60432.
- [18] A. K. Putri and A. S. Handayani, "Penerapan Arsitektur EfficientNet Untuk Pembuatan Model Algoritma Convolutional Neural Network Pada Klasifikasi Bahasa Isyarat," *Build. Informatics, Technol. Sci.*, vol. 6, no. 2, pp. 758–766, 2024, doi: 10.47065/bits.v6i2.5592.
- [19] H. Abbaszadeh, "Apple diseases 4 Classes (PlantVillage)," *Kaggle.com*, 2023. <https://www.kaggle.com/datasets/hamidabbaszadeh/apple-leave-diseases-dataset-without-augmentation>
- [20] A. Arjun, "Klasifikasi Citra Pada Tingkat Kematangan Buah Pisang Menggunakan Algoritma Deep Learning," *J. Ekon.*



Manaj. Sist. Inf., vol. 5, no. 3, pp. 203–208, 2024, doi: 10.31933/jemsi.v5i3.1786.

- [21] Nurul Chamidah, Mayanda Mega Santoni, and Nurhafifah Matondang, “Oversampling Method on Classifying Hypertension Using Naive Bayes, Decision Tree, and Artificial Neural Network,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 4, pp. 635–641, 2020, doi: 10.29207/resti.v4i4.2015.
- [22] M. Resa Arif Yudianto, P. Sukmasetya, R. Abul Hasani, and D. Sasongko, “Pengaruh Data Preprocessing terhadap Imbalanced Dataset pada Klasifikasi Citra Sampah menggunakan Algoritma Convolutional Neural Network,” *Build. Informatics, Technol. Sci.*, vol. 4, no. 3, pp. 1367–1375, 2022, doi: 10.47065/bits.v4i3.2575.
- [23] N. P. Ekananda and D. Rimirasih, “Identifikasi Penyakit Pneumonia Berdasarkan Citra Chest X-Ray Menggunakan Convolutional Neural Network,” *J. Ilm. Inform. Komput.*, vol. 27, no. 1, pp. 79–94, 2022, doi: 10.35760/ik.2022.v27i1.6487.
- [24] A. Desiani, D. A. Zayanti, R. Primartha, F. Efriliyanti, and N. A. C. Andriani, “Variasi Thresholding untuk Segmentasi Pembuluh Darah Citra Retina,” *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 2, pp. 255–262, 2021, doi: 10.26418/jp.v7i2.47205.
- [25] S. A. Widiarto, W. A. Saputra, and A. R. Dewi, “Klasifikasi Citra X-Ray Toraks Dengan Menggunakan Contrast Limited Adaptive Histogram Equalization Dan Convolutional Neural Network (Studi Kasus: Pneumonia),” *JIP1 (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 6, no. 2, pp. 348–359, 2021, doi: 10.29100/jipi.v6i2.2102.
- [26] S. Saifullah, “Segmentasi Citra Menggunakan Metode Watershed Transform Berdasarkan Image Enhancement Dalam Mendeteksi Embrio Telur,” *Syst. Inf. Syst. Informatics J.*, vol. 5, no. 2, pp. 53–60, 2020, doi: 10.29080/systemic.v5i2.798.
- [27] J. E. Widyaya and S. Budi, “Pengaruh Preprocessing Terhadap Klasifikasi Diabetic Retinopathy dengan Pendekatan Transfer Learning Convolutional Neural Network,” *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 110–124, 2021, doi: 10.28932/jutisi.v7i1.3327.
- [28] M. Ikhsan and A. Wiranda Hakiki, “Analisis Perbandingan Metode Histogram Equalization Dan Gaussian Filter Untuk Perbaikan Kualitas Citra,” *J. Sci. Soc. Res.*, vol. 4307, no. 2, pp. 487–492, 2024, doi: 10.54314/jssr.v7i2.1865.
- [29] M. M. Sugiman and H. D. Purnomo, “Prediksi Kegagalan Transformator Daya dengan Metode DGA (Dissolved Gas Analysis) Menggunakan Random Forest Berbasis TDCG,” *J. Media Inform. Budidarma*, vol. 8, no. 1, pp. 441–449, 2024, doi: 10.30865/mib.v8i1.7036.