

Pemanfaatan Convolutional Neural Network Dalam Klasifikasi Penyakit Tanaman Singkong Menggunakan Arsitektur Densenet

Use Of Convolutional Neural Networks On Classifying Cassava Diseases With Densenet Architecture

1st Aditya Fahrezantara
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
adityafahrezantara@student.telkomuniversity.ac.id

2nd Syamsul Rizal
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
syamsul@telkomuniversity.ac.id

3rd Nor Kumalasari Caesar Pratiwi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
caecarnkcp@telkomuniversity.ac.id

Abstrak—Singkong (*Manihot esculenta*) adalah tanaman yang memiliki permintaan yang tinggi di Indonesia, ditandai dengan peningkatan tingkat produksi seiring waktu. Selain kuantitas, kualitas tanaman harus dijaga, salah satunya dengan memperhatikan gejala penyakit. Gejala penyakit yang timbul pada daun singkong dapat dideteksi dengan inspeksi visual. Tetapi, diperlukan pengetahuan lebih untuk membedakan gejala suatu penyakit dengan penyakit lainnya. Salah satu solusi dari masalah ini adalah pemanfaatan *convolutional neural networks* (CNN) untuk klasifikasi penyakit. Penulis menggunakan model CNN berbasis arsitektur DenseNet untuk masalah ini. Parameter penilai kinerja model CNN yang digunakan adalah akurasi, presisi, *recall*, dan *F1-score*. Model CNN dilatih dengan *dataset* gambar berwarna berukuran 224×224 *pixel* dengan 5 kelas (CBSD, CMD, CBB, CGM, dan daun sehat) berjumlah 9430 gambar. Pengujian dilakukan untuk menentukan konfigurasi *hyperparameter* terbaik untuk model CNN. *Hyperparameter* yang divariasikan adalah *optimizer* dan *learning rate*. Dari hasil pengujian, model CNN berbasis DenseNet-169 dengan *optimizer* RMSprop dan *learning rate* 0,0001 memiliki hasil terbaik dengan akurasi *training* sebesar 98,73% dan akurasi *testing* sebesar 82%.

Kata Kunci — penyakit singkong, *convolutional neural network*, *computer vision*, *image classification*, DenseNet

Abstract—Cassava (*Manihot esculenta*) is a plant with high demand in Indonesia, shown by its increase in production rate over time. Other than quantity, quality of plant, especially in terms of disease infection, must be a

concern. Several diseases on cassava can be detected using visual inspection, but further knowledge is needed to identify the disease. To make detection and classification easier, convolutional neural networks (CNN) could be used. The writer used DenseNet architecture for this problem. Accuracy, precision, recall, and F1-score are the metrics used to measure CNN model's performance. The model is trained on colored image dataset with 224×224 pixels resolution and consists of 9430 images in five classes (CBSD, CMD, CBB, CGM, and healthy leaves). Testing was done to find the best combination of optimizer and learning rate for the CNN model to use. From the testing result, DenseNet-169 based model with RMSprop optimizer and 0,0001 learning rate has the best result with 98,73% training accuracy and 82% testing accuracy.

Keywords—cassava diseases, convolutional neural networks, computer vision, image classification, DenseNet

I. PENDAHULUAN

Singkong (*Manihot esculenta*), disebut juga sebagai ketela pohon dan ubi kayu, adalah salah satu tumbuhan berumbi yang menjadi salah satu makanan pokok bagi masyarakat di Indonesia. Oleh karena itu, produksi tanaman ini meningkat seiring waktu [1]. Untuk mengimbangi kuantitas produksi, kualitas produksi pun harus diperhatikan, salah satu caranya adalah dengan melakukan deteksi dini penyakit pada tanaman.

Penyakit pada singkong umumnya menginfeksi bagian daun, batang, atau akar tanaman tersebut [2].

Beberapa penyakit dapat dengan didiagnosis dengan inspeksi secara visual, namun diperlukan pengetahuan lebih lanjut untuk membedakan satu penyakit dengan penyakit lainnya. Dengan bantuan pembelajaran mesin (*machine learning*), penyakit pada singkong dapat diklasifikasikan jenisnya dengan mudah.

Suatu model *machine learning* dengan metode tertentu dapat dilatih untuk mengenali arti dari suatu gambar dengan tingkat akurasi yang baik [3]. Dalam kasus klasifikasi penyakit pada singkong, model *machine learning* yang dibuat dapat mengenali beragam penyakit dan dapat membedakan satu penyakit dengan penyakit lainnya. Dalam penelitian ini, model *machine learning* digunakan untuk mengklasifikasikan penyakit singkong yang tampak pada daun. Penelitian menggunakan salah satu cabang dari *machine learning*, yakni *deep learning* dengan metode *convolutional neural networks* (CNN). Penggunaan CNN didasarkan pada performa CNN yang lebih baik dari metode *machine learning* tradisional pada klasifikasi gambar [3]. Klasifikasi gambar dengan objek berupa daun juga pernah dilakukan dengan hasil yang sangat baik. Penelitian yang dilakukan oleh J. Yi et al. [4] dalam mendeteksi defisiensi nutrisi pada daun tanaman *sugar beet* menggunakan CNN menghasilkan akurasi maksimum sebesar 97,2% yang diraih dengan arsitektur DenseNet. B. R. D. Prá, R. N. De Mesquita, M. O. De Menezes, dan D. A. De Andrade [5] juga melakukan penelitian mengenai pemanfaatan CNN untuk mendeteksi defisiensi nutrisi pada rerumputan *Brachiaria brizantha* cv. marandu. Akurasi maksimum yang didapatkan sebesar 96%. Google [6] dengan model CNN yang dinamakan CropNet berbasis arsitektur MobileNetV3 memiliki akurasi maksimum sebesar 88% dalam kasus klasifikasi penyakit pada singkong. Dapat disimpulkan bahwa klasifikasi dengan objek berupa daun dapat dilakukan oleh CNN dengan akurasi tinggi.

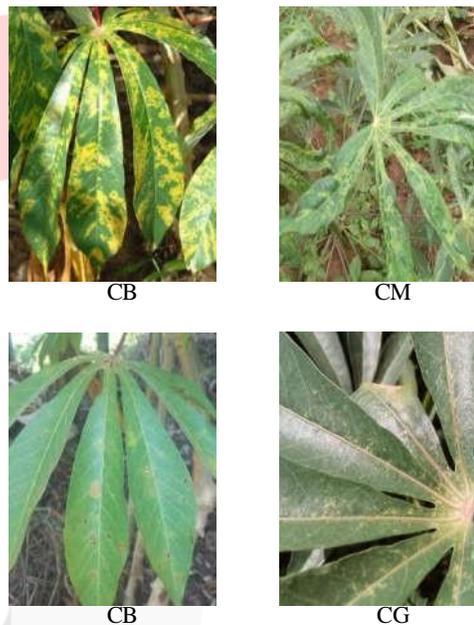
Pada penelitian ini, digunakan model CNN berbasis arsitektur DenseNet untuk melakukan klasifikasi penyakit pada singkong. Model CNN dilatih menggunakan *dataset* berisi 9430 gambar daun singkong berwarna dengan lima kelas yakni daun yang terpapar Cassava Brown Streak Disease (CBSD), Cassava Mosaic Disease (CMD), Cassava Bacterial Blight (CBB), Cassava Green Mite (CGM), dan daun yang sehat. Model CNN dilatih menggunakan *hyperparameter optimizer* dan *learning rate* yang berbeda. Kombinasi *hyperparameter* dengan akurasi terbaik kemudian dianalisis kinerjanya menggunakan parameter presisi, *recall*, dan *F1-score*.

II. KAJIAN TEORI

A. Penyakit Tanaman Singkong

Pendeteksian gejala beberapa penyakit pada singkong dapat dilakukan secara visual. Pada daun,

gejala dapat terlihat dari perubahan warna dan bentuk daun. Salah satu gejala yang muncul pada batang adalah tampilan batang terlihat rusak dan teksturnya berbeda. Gejala yang tampak pada akar dapat dilihat pada akar yang telah dicabut, di mana terdapat perubahan di bagian dalam atau luar umbi. Diagnosis yang mudah dilakukan diikuti dengan edukasi mengenai penyebab penyakit dapat memudahkan proses mitigasi. Fokus utama penelitian ini adalah klasifikasi penyakit dengan gejala yang muncul pada daun singkong, yakni Cassava Brown Streak Disease (CBSD), Cassava Mosaic Disease (CMD), Cassava Bacterial Blight (CBB), dan Cassava Green Mite (CGM).



GAMBAR 1
GEJALA PENYAKIT PADA DAUN SINGKONG

B. Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) adalah metode *deep learning* yang merupakan pengembangan lebih lanjut dari *artificial neural networks* (ANN). ANN adalah sekumpulan elemen pemroses (disebut unit, neuron, atau *node*) yang saling terhubung yang cara kerjanya didasarkan pada jaringan saraf yang sebenarnya, aktif hanya saat mendapatkan respons yang memadai [7]. Sekumpulan *node* pada posisi yang sama disebut sebagai *layer* dengan penghubung antar *layer* disebut *weight*. *Neural network* setidaknya memiliki dua *layer*, yakni *layer input* dan *layer output*. Jika hubungan antara *node* di *layer input* dan *node* di *layer output* cukup kuat, ANN akan menghasilkan suatu nilai keluaran.

Deep learning [8] adalah pengembangan dari ANN dengan dua atau lebih layer non-output. Layer-layer yang terletak di antara layer input dan output disebut sebagai hidden layers. Dengan meningkatnya kapabilitas komputer dalam melakukan komputasi, kini deep learning menjadi salah satu metode populer dalam machine learning [3], [9]. Salah satu pengembangan dari deep learning adalah convolutional neural networks (CNN).

CNN adalah pengembangan lebih lanjut dari ANN. Perbedaan utama dari ANN dengan CNN terletak pada satu komponen terpenting CNN, yakni convolution layer (lapisan konvolusi). Convolution layer bekerja sebagai filter pada input, dalam hal ini gambar latih, yang menghasilkan suatu activation (aktivasi) [10]. Filter ini dapat dilatih untuk mengenali suatu karakteristik, sehingga disebut sebagai filter trainable [8]. Filter ini diaplikasikan secara bertahap pada bagian-bagian gambar. Setelah filter diaplikasikan berulang kali pada gambar yang sama, akan dihasilkan pemetaan dari activation, disebut juga feature map, yang berisi posisi dan kekuatan dari fitur yang terdeteksi dari gambar tersebut. Proses konvolusi ini menyebabkan transformasi pada beberapa bagian gambar, bagian yang kurang penting akan hilang dan bagian yang penting akan sangat tampak [11].

Selain convolution layer, terdapat pooling layer dan fully connected layer. Pooling layer bekerja sebagai filter seperti convolution layer, hanya saja menggunakan filter berupa operator tetap, umumnya nilai maksimum atau nilai rata-rata [8]. Fully connected layer, disebut juga dense layer, sesuai namanya adalah layer dengan node yang terhubung dengan semua node pada layer yang mendahuluinya, layer ini bekerja seperti ANN [12].

C. DenseNet

Dense Convolutional Network (DenseNet) [13] pertama kali diperkenalkan pada tahun 2017 oleh Gao Huang, Zhuang Liu, Laurens van der Maaten, dan Kilian Q. Weinberger pada konferensi CVPR 2017 dan berhasil memenangkan penghargaan Best Paper Award [14]. Fitur utama dari arsitektur ini adalah dense block, DenseNet menghubungkan langsung satu convolution layer dengan semua convolution layer setelahnya, di mana semua layer mengambil input berupa feature map dari semua layer sebelumnya. Dengan semakin banyak hubungan langsung, arsitektur ini dapat menjadi lebih akurat dan lebih efisien karena koneksi antara layer dekat input dan layer dekat output semakin sedikit.

TABEL 1

STRUKTUR ARSITEKTUR DENSENET

Layers	Output Size	DenseNet-101	DenseNet-169	DenseNet-201
Convolution	112 × 112	7 × 7 conv, stride 2		
Pooling	56 × 56	3 × 3 max pool, stride 2		
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv		
	28 × 28	2 × 2 average pool, stride 2		
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv		
	14 × 14	2 × 2 average pool, stride 2		
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Transition Layer (3)	14 × 14	1 × 1 conv		
	7 × 7	2 × 2 average pool, stride 2		
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Classification Layer	1 × 1	7 × 7 global average pool		
		1000D fully-connected, softmax		

D. Transfer Learning

Transfer learning adalah metode pelatihan model CNN yang bekerja dengan memanfaatkan model yang sudah dilatih dengan suatu dataset. Model yang sudah dilatih ini kemudian dilatih ulang dengan dataset baru yang berbeda dengan dataset awal [8]. Model akan lebih mudah untuk mengenali pola yang ada pada dataset baru dengan pengetahuan yang didapat dari dataset lama, sehingga pelatihan model akan lebih singkat. Selain itu, dengan transfer learning, kinerja model yang baik dapat dicapai dengan lebih sedikit data latih [9].

III. METODE

A. Desain Sistem

Pada penelitian ini, dilakukan perbandingan kinerja model berbasis arsitektur DenseNet dengan hyperparameter bervariasi dalam melakukan klasifikasi penyakit pada singkong. Masing-masing model CNN dengan hyperparameter yang bervariasi dilatih menggunakan dataset penyakit singkong berupa gambar yang telah melalui tahap preprocessing. Kemudian, model dengan akurasi validation terbaik dievaluasi dengan parameter pengujian berupa presisi, recall, dan F1-score.

B. Dataset

Dataset yang digunakan adalah *dataset* berjudul *Cassava Disease Classification* yang berasal dari Tensorflow Datasets yang diperkenalkan pada konferensi CVPR 2019 [15]. *Dataset* terdiri atas 9430 gambar berwarna dengan label berupa kelas gambar tersebut. 80% dari *dataset* (7544) dijadikan data latih dan sisanya sebagai data validasi dan uji. Terdapat lima kelas pada *dataset* yang digunakan, yakni daun yang terpapar CBSD (2406 gambar), CMD (4431 gambar), CBB (777 gambar), CGM (1289 gambar), dan daun yang sehat (527 gambar).

C. *Dataset Preprocessing*

Sebelum digunakan sebagai data latih oleh model CNN, dilakukan *preprocessing* pada *dataset* yang digunakan. Dilakukan normalisasi, yakni mengubah skala intensitas warna pada gambar yang awalnya bernilai 0–255 menjadi bernilai 0–1, dan perubahan ukuran gambar menjadi 224×224 *pixel*.

D. Parameter Pengukur Kinerja

Parameter pengukur kinerja model CNN yang digunakan adalah akurasi, presisi, *recall*, dan *F1-score*.

1. Akurasi

$$= \frac{+}{+ + +} \tag{1}$$

Akurasi menunjukkan perbandingan antara prediksi yang akurat dan semua prediksi yang dibuat [16].

2. Presisi

$$= \frac{+}{+} \tag{2}$$

Presisi menunjukkan seberapa banyak nilai yang benar-benar positif dari seluruh data yang diprediksi positif [16].

3. *Recall*

$$= \frac{+}{+} \tag{3}$$

Recall menunjukkan seberapa banyak nilai yang benar-benar positif dari seluruh data yang sebenarnya positif [16].

4. *F1-score*

$$1 - = 2 \times \frac{\times}{+} \tag{4}$$

F1-score adalah *harmonic mean* dari presisi dan *recall* [16].

E. *Cross-Validation*

Untuk mendapatkan konfigurasi *hyperparameter* terbaik, dilakukan *cross-validation* pada pelatihan model CNN dasar. Dalam metode ini, pelatihan model dilakukan sebanyak beberapa kali. Banyaknya pelatihan model dihitung dengan membagi jumlah data dengan data validasi. Pada pengujian pada penelitian ini, masing-masing model CNN dasar dilatih sebanyak lima kali sebab data validasi yang digunakan sebesar 20%. Pada pelatihan pertama, model menggunakan 20% data pertama sebagai data validasi. Selanjutnya pada pelatihan kedua, model menggunakan 20% data kedua sebagai data validasi. Pelatihan dilakukan hingga semua data pernah digunakan sebagai data latih dan data uji.

IV. HASIL DAN PEMBAHASAN

A. Skenario Pengujian

Dilakukan pelatihan model-model dasar, yakni model-model CNN berbasis DenseNet-121 dengan *optimizer* dan *learning rate* yang berbeda untuk masing-masing model. Model-model ini dilatih menggunakan *learning rate* 0,01; 0,001; dan 0,0001. *Optimizer* yang digunakan untuk masing-masing *learning rate* adalah Adam, RMSprop, dan SGD. Pelatihan model CNN menggunakan *batch size* sebesar 64 dan dengan *epoch* sebanyak 25. Pada akhir tahap pengujian ini, didapatkan model dengan akurasi *validation* terbaik untuk diuji kinerjanya dari segi presisi, *recall*, dan F1-score. Selanjutnya, dilakukan penyempurnaan pada model dasar terbaik dengan mengubah arsitektur dasar model dan memanfaatkan augmentasi pada *dataset* yang digunakan.

B. Hasil dan Analisis Pengujian

Berikut hasil dan analisis pada pengujian dengan model dasar. Nilai akurasi dan *loss* yang ditampilkan adalah rerata dari hasil *cross-validation* model. Model terbaik pada masing-masing skenario ditandai dengan warna biru muda.

1. Pengujian Dengan *Learning Rate* 0,01

Hasil pengujian skenario ini menunjukkan bahwa model dengan *optimizer* Adam dan RMSprop mengalami *underfitting*, yakni keadaan di mana model tidak dapat melakukan generalisasi pada data dengan baik, ditunjukkan oleh akurasi *training* dan *validation* yang rendah. Model dengan *optimizer* SGD memiliki nilai akurasi *training* yang baik, namun nilai akurasi *validation* yang terpaut jauh. Hal ini dinamakan *overfitting*, di mana model melakukan generalisasi berlebihan pada data latih sehingga tidak bekerja baik pada data baru [9].

TABEL 2
NILAI RERATA AKURASI DAN *LOSS* PADA *LEARNING RATE* 0,01

<i>Optimizer</i>	Akurasi <i>Training</i>	Akurasi <i>Validation</i>	<i>Loss Training</i>	<i>Loss Validation</i>
Adam	67,50%	67,19%	0,79792	1,3487
RMSprop	53,59%	54,69%	1,19996	1,31956
SGD	99,12%	77,19%	0,07894	0,68748

2. Pengujian Dengan *Learning Rate* 0,001

Pada skenario ini, performa masing-masing model relatif sama ditinjau dari segi akurasi *validation*, dengan nilai berkisar 70%. Pada akurasi *training*, model dengan *optimizer* SGD memiliki akurasi terendah. Walaupun begitu, model ini tidak memiliki masalah *overfitting* seperti model dengan *optimizer* Adam dan RMSprop, ditandai dengan nilai akurasi *training* dan *validation* yang tidak terpaut jauh. Model ini pun memiliki nilai akurasi *validation* tertinggi.

TABEL 3
NILAI RERATA AKURASI DAN *LOSS* PADA *LEARNING RATE* 0,001

<i>Optimizer</i>	Akurasi <i>Training</i>	Akurasi <i>Validation</i>	<i>Loss Training</i>	<i>Loss Validation</i>
Adam	95,80%	73,12%	0,1164	1,20058
RMSprop	96,06%	72,81%	0,16328	2,40256
SGD	83,57%	77,19%	0,53908	0,70824

3. Pengujian Dengan *Learning Rate* 0,0001

Dari hasil pengujian yang didapatkan, model dengan *optimizer* Adam dan RMSprop memiliki peningkatan performa yang sedikit lebih baik dari model pada skenario sebelumnya. Tetapi, model dengan *optimizer* SGD, *optimizer* model terbaik pada dua skenario sebelumnya, mengalami *underfitting*.

TABEL 4
NILAI RERATA AKURASI DAN *LOSS* PADA *LEARNING RATE* 0,0001

<i>Optimizer</i>	Akurasi <i>Training</i>	Akurasi <i>Validation</i>	<i>Loss Training</i>	<i>Loss Validation</i>
Adam	99,47%	75,31%	0,02618	1,0398
RMSprop	98,12%	78,44%	0,07968	1,04172
SGD	60,60%	59,69%	1,0433	1,06508

Dari ketiga skenario pengujian yang dilakukan, model dengan *learning rate* 0,0001 dan *optimizer* RMSprop memiliki performa terbaik ditinjau dari akurasi *validation*. Model ini kemudian dievaluasi dengan parameter presisi, *recall*, dan F1-Score dan selanjutnya digunakan pada tahap selanjutnya, yakni penyempurnaan model.

4. Pengujian Parameter Presisi, Recall, dan F1-Score pada Model Terbaik

Dilakukan evaluasi lebih lanjut pada model terbaik pada pengujian yang sudah dilakukan, yakni model dengan *learning rate* 0,0001 dan *optimizer* RMSprop.

TABEL 5
NILAI PRESISI, RECALL, DAN F1-SCORE MODEL TERBAIK

Kelas	Presisi	Recall	F1-Score
CMD	0,84	0,92	0,88
CBSD	0,74	0,82	0,78
CGM	0,71	0,54	0,62
CBB	0,62	0,43	0,51
Daun Sehat	0,71	0,59	0,65

Dari hasil evaluasi, diketahui bahwa model memiliki performa yang tidak seimbang untuk tiap kelasnya. Untuk meningkatkan performa, dilakukan penyempurnaan pada model terbaik.

C. Penyempurnaan Model

Model dasar terbaik pada tahap sebelumnya dijadikan acuan untuk tahap ini. Dilakukan perubahan pada pelatihan model untuk mendapatkan hasil yang lebih baik. Perubahan mencakup dua hal, yakni pengubahan varian DenseNet sebagai arsitektur dasar model dan penggunaan augmentasi

pada *dataset*. Proporsi data yang digunakan pada tahap ini adalah 80% data latih, 10% data validasi, dan 10% data uji.

1. Pengubahan Arsitektur Dasar Model

Digunakan dua varian DenseNet tambahan yakni DenseNet-169 dan DenseNet-201 sebagai arsitektur dasar model.

TABEL 6
NILAI AKURASI TIGA MODEL BERBASIS VARIAN DENSENET BERBEDA

Arsitektur Dasar	Akurasi		
	Training	Validation	Testing
DenseNet-121	94,53%	79,87%	79%
DenseNet-169	98,44%	81,04%	82%
DenseNet-201	98,52%	79,56%	79%

Pada tahap ini, model dengan arsitektur DenseNet-169 memiliki performa terbaik, sehingga model ini yang akan digunakan untuk tahap selanjutnya.

2. Augmentasi Dataset

Sebagai upaya peningkatan performa model, dilakukan augmentasi pada data latih. Digunakan empat jenis augmentasi yakni *flip*, *rotation*, *zoom*, dan *random contrast*.

TABEL 7
NILAI AKURASI DENGAN AUGMENTASI DATASET

Model	Augmentasi				Akurasi		
	Flip	Rotation	Zoom	Contrast	Training	Validation	Testing
DenseNet-169	✓	-	-	-	98,70%	81,57%	81%
DenseNet-169	-	✓	-	-	98,30%	78,71%	78%
DenseNet-169	-	-	✓	-	98,36%	78,28%	79%
DenseNet-169	-	-	-	✓	98,65%	80,51%	82%
DenseNet-169	✓	-	-	✓	98,73%	81,67%	82%
DenseNet-169	✓	✓	✓	✓	98,12%	74,47%	76%

Data latih dengan augmentasi *flip* dan *random contrast* menghasilkan model dengan performa terbaik, namun hasil yang didapatkan tidak berbeda signifikan dengan performa model dengan data latih tanpa augmentasi. Model ini kemudian dianggap sebagai model terbaik dari keseluruhan pengujian.

D. Perbandingan Kinerja dengan Model Lain

Pada bagian "Pendahuluan" disebutkan bahwa terdapat model CNN buatan Google yang dibuat

khusus untuk mengklasifikasikan penyakit singkong dengan nama CropNet. Dilakukan perbandingan performa antara model terbaik dari pengujian ini dan CropNet.

TABEL 8
PERBANDINGAN KINERJA MODEL TERBAIK DAN CROPNET

Kelas	Model Terbaik	CropNet
-------	---------------	---------

	Presisi	Recall	F1-Score	Presisi	Recall	F1-Score
CMD	0,88	0,91	0,90	0,94	0,97	0,95
CBSD	0,75	0,85	0,80	0,89	0,81	0,85
CGM	0,79	0,68	0,73	0,86	0,86	0,86
CBB	0,68	0,45	0,54	0,79	0,58	0,67
Daun Sehat	0,73	0,67	0,70	0,59	0,90	0,71
Akurasi Testing	82%			88%		

Dari hasil perbandingan diketahui bahwa CropNet mengungguli model terbaik pada pengujian ini dari segi akurasi *testing* dan nilai *F1-Score* setiap kelas.

V. KESIMPULAN

Implementasi klasifikasi gejala penyakit pada daun singkong berhasil dilakukan menggunakan model *convolutional neural network* (CNN) berbasis arsitektur DenseNet. Dari hasil pengujian, didapatkan bahwa model CNN dengan arsitektur dasar DenseNet-169, *optimizer* RMSprop, dan *learning rate* 0,0001 memiliki performa terbaik dibanding dengan variasi model lainnya. Dengan menggunakan metode augmentasi tertentu pada data latih, model ini memiliki akurasi *training* 98,73%, akurasi *validation* 81,67%, dan akurasi *testing* 82%. Meskipun memiliki akurasi yang cukup tinggi, performa model untuk tiap kelas tidak seimbang. Salah satu faktor utama dari masalah ini adalah ketidakseimbangan jumlah gambar untuk tiap kelas pada *dataset* yang digunakan.

REFERENSI

[1] "Produksi Ubi Kayu Menurut Provinsi (ton), 1993-2015." <https://www.bps.go.id/dynamictable/2015/09/09/880/produksi-ubi-kayu-menurut-provinsi-ton-1993-2015.html> (diakses Nov 11, 2021).

[2] N. Saleh, D. Harnowo, dan I. M. J. Mejaya, *Penyakit-Penyakit Penting pada Ubi Kayu Deskripsi, Bioekologi, dan Pengendaliannya*. Malang: Balai Penelitian Tanaman Aneka Kacang dan Umbi, 2016.

[3] N. O'Mahony *et al.*, "Deep Learning vs. Traditional Computer Vision," *Adv. Intell. Syst. Comput.*, vol. 943, no. Cv, hal. 128-144, 2020, doi: 10.1007/978-3-030-17795-9_10.

[4] J. Yi *et al.*, "Deep learning for non-invasive diagnosis of nutrient deficiencies in sugar beet using RGB images," *Sensors (Switzerland)*, vol. 20, no. 20, hal. 1-19, 2020, doi: 10.3390/s20205893.

[5] B. R. D. Prá, R. N. De Mesquita, M. O. De Menezes, dan D. A. De Andrade, "Nutritional evaluation of brachiaria brizantha cv. Marandu using convolutional neural networks," *Intel. Artif.*, vol. 23, no. 66, hal. 85-96, 2020, doi: 10.4114/intartif.vol23iss66pp85-96.

[6] "CropNet: Cassava Disease Detection | TensorFlow Hub." https://www.tensorflow.org/hub/tutorials/cropnet_cassava (diakses Agu 03, 2022).

[7] K. Gurney, *An introduction to neural networks*. Taylor & Francis e-Library, 1997.

[8] A. Burkov, *The Hundred-Page Machine Learning Book*. Quebec City: Andriy Burkov, 2019.

[9] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. Sebastopol: O'Reilly Media, 2019.

[10] J. Brownlee, "How Do Convolutional Layers Work in Deep Learning Neural Networks?" <https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/> (diakses Des 20, 2021).

[11] J. P. Mueller dan L. Massaron, *Machine Learning For Dummies*, 1st ed. New Jersey: John Wiley & Sons, 2016.

[12] S. Albawi, T. A. M. Mohammed, dan S. Alzawi, "Understanding of a Convolutional Neural Network," *Int. Conf. Educ. Technol.*, hal. 16, 2017.

[13] G. Huang, Z. Liu, L. Van Der Maaten, dan K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, hal. 2261-2269, 2017. doi: 10.1109/CVPR.2017.243.

[14] "CVPR2017." https://cvpr2017.thecvf.com/program/main_conference#cvpr2017_awards (diakses Des 05, 2021).

[15] E. Mwebaze, T. Gebru, A. Frome, S. Nsumba, dan J. Tusubira, "iCassava 2019 Fine-Grained Visual Categorization Challenge," Agu 2019, doi: 10.48550/arxiv.1908.02900.

[16] M. Vakili, M. Ghamsari, dan M. Rezaei, "Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification," 2020, [Daring]. Tersedia pada: <http://arxiv.org/abs/2001.09636>.