

Implementasi Transfer Learning Dalam Mendeteksi Penyakit Pada Daun Gandum

Faisal Mashuri*¹, Ultach Enri²

^{1,2,3}Universitas Singaperbangsa Karawang

E-mail: *¹faisalmashuri16@gmail.com, ²ultach@staff.unsika.ac.id

Abstrak

Gandum merupakan salah satu komoditas yang paling sering di konsumsi oleh masyarakat Indonesia. Tanaman ini sering sekali dikonsumsi sebagai tambahan karbohidrat maupun pengganti beras. Kebanyakan masyarakat Indonesia mengonsumsi gandum untuk dijadikan bahan olahan seperti tepung, roti, sereal dan bahan olahan lainnya. Banyaknya permintaan gandum ini tidak diimbangi dengan tingkat produksi. Salah satu faktor yang menghambat produksi gandum adalah kegagalan panen akibat penyakit atau hama. Penyakit yang sering dijumpai pada tanaman gandum adalah Septoria dan Stripe Rust. Penyakit tersebut dapat diidentifikasi melalui warna dan bercak daun, tetapi sukar untuk membedakan kedua penyakit tersebut. Dengan perkembangan teknologi yang pesat masalah ini dapat diselesaikan menggunakan salah satu teknik deep learning yang dikenal dengan transfer learning. Tujuan pada penelitian ini adalah menguji lima pretrained model untuk mendeteksi penyakit pada daun gandum, beberapa model yang diujikan adalah Inception V3, MobileNetV2, VGG16, ResNet101V2, DenseNet 201. Hasil dari pengujian dan membandingkan lima pretrained model, Inception V3 memberikan hasil yang lebih baik dibandingkan model yang lain dengan waktu komputasi yang rendah yaitu hanya 976 detik atau setara dengan 16 menit serta memiliki akurasi yang sangat tinggi.

Kata Kunci—deep learning, deteksi penyakit pada gandum, convolutional neural network, transfer learning

Abstract

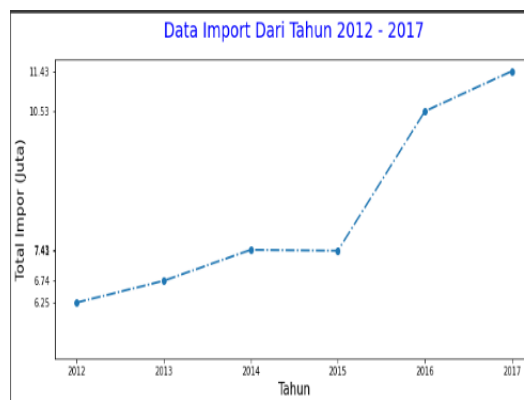
Wheat is one of the most frequently consumed commodities of Indonesian people. This plant is often consumed as an carbohydrate addition or rice substitution. Most Indonesians process the wheat for ingredients such as flour, bread, instant noodles, cereals and other processed ingredients. Unfortunately, the demand for wheat is not suitable with level of production. One of the factors that hinder wheat production is crop failure due to disease or pests. Diseases that are often found in wheat are Septoria and Stripe Rust. The disease can be identified by color and leaf spot, but it is difficult to distinguish between the two diseases. With the rapid development of technology, this problem can be solved using one of the deep learning techniques known as transfer learning. The purpose of this study was to test five pretrained models to diagnose disease in wheat leaf, the models tested were InceptionV3, MobileNetV2, VGG16, ResNet101V2, DenseNet201. The results of testing and comparing five pretrained models, InceptionV3 gives better results than other models with a low computation time of only 976 seconds or the equivalent of 16 minutes and has a very high accuracy.

Keywords—deep learning, wheat leaf diseases detection, convolutional neural network, transfer learning

masih tergolong tinggi.

1. PENDAHULUAN

Gandum merupakan salah satu komoditas yang paling sering di konsumsi oleh masyarakat Indonesia. Tanaman ini bukanlah makanan pokok bagi masyarakat Indonesia, namun perannya semakin penting dalam beberapa tahun terakhir. Tanaman yang termasuk kedalam genus *Triticum* ini memiliki jumlah karbohidrat yang lebih banyak dibandingkan dengan beras sehingga sering sekali dikonsumsi sebagai tambahan karbohidrat maupun pengganti beras. Kebanyakan masyarakat Indonesia mengonsumsi gandum untuk dijadikan bahan olahan seperti tepung, roti, mi instan, sereal dan bahan olahan lainnya. Tingkat konsumsi gandum di Indonesia dari tahun ke tahun terus mengalami peningkatan [1]. Dan pada Tahun 2019 Badan Pusat Statistik (BPS) Indonesia menyatakan bahwa sedikitnya gandum di konsumsi sebesar 30,5 kg/tahun oleh masyarakat Indonesia. Tapi permintaan yang tinggi ini tidak di imbangi dengan produksi dari petani gandum itu sendiri, sehingga pemerintah harus melakukan impor dari berbagai negara. Salah negara pengimpor gandum terbesar adalah Australia, hal ini terjadi dikarenakan Australia merupakan mitra dagang terdekat secara geografis dengan Indonesia [1]. Pada tahun 2019, Badan Pusat Statistik (BPS) Indonesia menyatakan bahwa sedikitnya gandum di konsumsi sebesar 30,5 kg/tahun oleh masyarakat Indonesia. Tapi permintaan yang tinggi ini tidak di imbangi dengan produksi dari petani gandum itu sendiri, sehingga pemerintah harus melakukan impor dari berbagai negara. Salah negara pengimpor gandum terbesar adalah Australia, hal ini terjadi dikarenakan Australia merupakan mitra dagang terdekat secara geografis dengan Indonesia Berdasarkan data BPS tingkat impor gandum di Indonesia dalam kurun waktu 2012 – 2017 mengalami kenaikan seperti yang ditunjukkan pada Gambar 1. Walaupun pada tahun 2020 terjadi penurunan impor menjadi 8 Juta ton, namun jumlah tersebut



Gambar 1 Kenaikan impor gandum 2012 – 2017

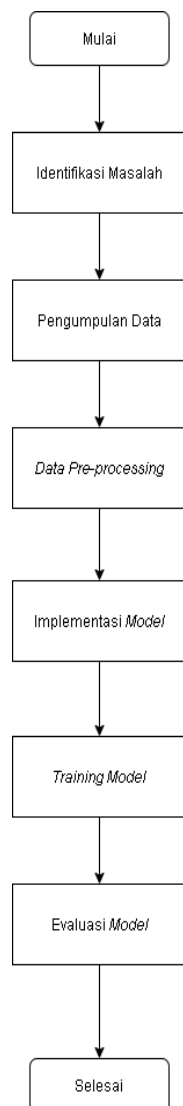
Rendahnya tingkat produksi gandum di negara kita disebabkan oleh faktor alam dan penyakit atau hama yang sering menyerang tanaman gandum. Penyakit yang paling umum terjadi pada gandum diantaranya *stripe rust* dan septoria. Septoria adalah sebuah penyakit pada daun jamur yang disebabkan oleh cendawan. Penyakit ini menyebabkan perubahan warna *glume* dan *nekrosis*, yang disebut penyakit bercak daun dan penyakit *glume* [2]. *Stripe rust* ditandai dengan bintil merah bata, yang dapat dibersihkan dengan tangan [2]. Karat batang paling umum pada gulungan daun. Kedua penyakit ini sekilas terlihat mirip, yang membedakannya adalah warna dan corak, sehingga petani sulit untuk membedakan penyakit ini yang dapat menyebabkan menurunnya hasil produksi dikarenakan gandum terserang penyakit.

Semakin pesatnya perkembangan teknologi saat ini, masalah petani tersebut dapat dibantu dengan bantuan mesin atau komputer, sehingga dapat membantu petani mendeteksi penyakit pada tanaman gandum dengan lebih mudah. Untuk melakukan deteksi penyakit pada tanaman gandum, mesin atau komputer dirancang untuk dapat melihat dan mendeteksi objek atau lebih dikenal dengan nama *computer vision* (visi komputer). Penelitian ini akan

memanfaatkan beberapa model yang sudah di-training sebelumnya (*pre-trained*) sehingga tidak perlu untuk melakukan *training data* dari awal untuk mendeteksi penyakit gandum, teknik ini dikenal dengan *transfer learning* [3].

2. METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah dengan *transfer learning* untuk dapat mendiagnosis penyakit pada daun gandum menggunakan bahasa pemrograman *python*. Adapun tahapan yang digunakan sebagaimana pada Gambar 2.



Gambar 2 Tahapan metodologi penelitian

2.1. Identifikasi Masalah

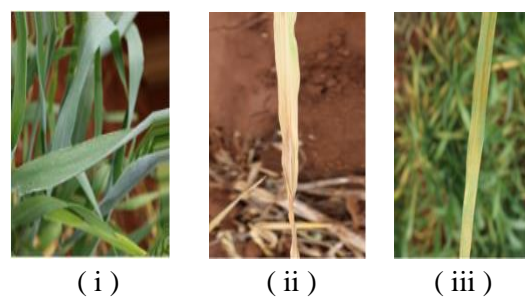
Identifikasi Masalah dilakukan untuk menguraikan masalah dengan jelas sehingga dapat membuat solusi yang optimal. Untuk masalah ini kita akan mendiagnosis penyakit daun gandum berdasarkan citra yang sudah dikumpulkan.

2.2. Pengumpulan Data

Data yang digunakan pada penelitian ini merupakan data *open source* berisi 407 data gambar yang diambil dari Kaggle (*wheat leaf desesea*). Data ini terbagi menjadi tiga kategori sehat, penyakit *stripe rust*, penyakit septoria, dimana persebaran dari tiga kategori tersebut tidak merata. Untuk kategori sehat berisi 102 gambar daun gandum dengan kondisi sehat, 97 gambar daun gandum dengan kondisi terkena penyakit septoria, 208 gambar daun gandum dengan kondisi terkena penyakit *stripe rust*.

Tabel 1 Jumlah data setiap kelas

Sample	Jumlah Data	Sumber Data
Sehat	102	Kaggle (Wheat leaf)
Septoria	97	
<i>Stripe rust</i>	208	
Total	497	



Gambar 3 Gambar (i) menunjukkan daun gandum yang sehat , (ii) menunjukkan daun yang terjangkit penyakit septoria, (iii) menunjukkan daun gandum yang terjangkit penyakit stripe rust

2.3. Data Pre-processing

Persebaran data yang tidak merata dapat memperbesar peluang kesalahan diagnosis penyakit daun gandum yang

dilakukan oleh sistem. Untuk itu diperlukan *pre-processing data*, dimana data yang tersebut dilakukan *oversampling*. *Synthetic Minority Oversampling Technique (SMOTE)* adalah salah satu turunan dari *oversampling*. *SMOTE* pertama kali diperkenalkan oleh Nithes V. Chawla. Pendekatan ini bekerja dengan membuat replikasi dari data minoritas [4]. Dengan teknik ini makan dataset yang kita punya menjadi *balanced* (seimbang) pada tiap kategori menjadi 200 data. Kemudian *dataset* akan dibagi lagi menjadi tiga *data train, data test, data validation*. *Data train* berjumlah 600 data citra, *data test* berjumlah 21 data citra, *data validation* berjumlah 20 data citra dengan masing - masing ukuran citra 300 x 300 piksel.

2.4. Implementasi Model

Model yang akan gunakan menggunakan *transfer learning* yang berbasis pada arsitektur *convolutional neural network*. Inti dari *Convolutional Neural Networks (CNN)* adalah lapisan *convolutional* yang dapat mengekstrak fitur lokal (misalnya tepi) melintasi gambar input melalui konvolusi. Setiap *node* dalam lapisan *convolutional* terhubung ke subset kecil dari neuron yang terhubung secara spasial [3].

Convolutional Neural Network (CNN) adalah salah satu jenis *neural network* yang biasa digunakan pada *data image*. *CNN* bisa digunakan untuk mendeteksi dan mengenali objek pada sebuah citra [5].

CNN dilatih dengan algoritma *backpropagation*, dimana dalam setiap iterasi, bobot yang terkait dengan *neuron* di lapisan *convolutional* diperbarui dengan cara yang meminimalkan sebuah fungsi biaya. Namun, dengan set data pelatihan yang kecil, hal ini dapat mengakibatkan dalam fungsi biaya terjebak dalam minimum lokal, yang dapat mengakibatkan *overfitting / underfitting*.

Alternatif yang lebih baik di kasus seperti itu adalah *transfer learning*, di mana bobot yang telah dilatih sebelumnya dari arsitektur yang sama, tetapi pada *dataset*

berbeda yang lebih besar dari domain yang sama/berbeda digunakan untuk menginisialisasi jaringan lapisan. Hanya lapisan terakhir yang terhubung penuh yang dilatih kembali dengan data pelatihnnya. Ini tidak hanya memberi kita seperangkat yang kuat bobot pra-latihan untuk dikerjakan, itu memberi kita kesempatan untuk menggunakan arsitektur jaringan yang telah terbukti dalam masalah kita. terdapat lima arsitektur populer yang akan kita gunakan dalam makalah ini adalah:

1) Inception V3

Arsitektur Inception adalah varian dari arsitektur *deep learning* yang dibangun oleh Google. Arsitektur ini memiliki parameter yang lebih sedikit dibanding arsitektur *VGG16*. Model ini diciptakan karena pada penelitian sebelumnya bahwa *auxiliary classifier* (pengklasifikasian tambahan) tidak berguna [6]. Untuk itu *Inception V3* diciptakan dengan tambahan *RMSprop optimizer, Factorized 7x7 convolutional, Label Smoothing* [7].

Tabel 2. Arsitektur Inception v3

Type	Filter size / stride	Input size
Conv	3 x 2 / 2	299 x 299 x 3
Conv	3 x 3 / 1	149 x 149 x 32
Conv padded	3 x 3 / 2	147 x 147 x 64
Pool	3 x 3 / 2	147 x 147 x 64
Conv	3 x 3 / 1	73 x 73 x 64
Conv	3 x 3 / 2	71 x 71 x 80
conv	3 x 3 / 1	35 x 35 x 192
Inception layer	3 x 3 / 2	35 x 35 x 288
Inception layer	5 x 5 / 1	17 x 17 x 768
Inception layer	7 x 7 / 2	8 x 8 x 1280
pool	8 x 8	8 x 8 x 2048
Linear	Logits	1 x 1 x 2048

2) MobileNet

MobileNet dikembangkan untuk mengatasi kelemahan Inception dengan menambahkan *deep wise convolutional* sehingga dapat berjalan di perangkat yang memiliki komputasi yang rendah, seperti perangkat *mobile* [8].

Tabel 3. Arsitektur MobileNet

Type / Stride	Filter Shape	Input Size
Conv / s2	3×3×3×32	224×224×3
Conv dw / s1	3×3×32 dw	112×112×32
Conv / s1	1×1×32×64	112×112×32
Conv dw / s2	3×3×64 dw	112×112×64
Conv / s1	1×1×64×128	56×56×64
Conv dw / s1	3×3×128 dw	56×56×128
Conv / s1	1×1×128×128	56×56×128
Conv dw / s2	3×3×128 dw	56×56×128
Conv / s1	1×1×128×256	28×28×128
Conv dw / s1	3×3×256 dw	28×28×256
Conv / s1	1×1×256×256	28×28×256
Conv dw / s2	3×3×256 dw	28×28×256
Conv / s1	1×1×256×512	14×14×256
Conv dw / s1	3×3×512 dw	14×14×512
Conv / s1	1×1×512×512	14×14×512
Conv dw / s2	3×3×512 dw	14×14×512
Conv / s1	1×1×512×1024	7×7×512
Conv dw / s2	3×3×1024 dw	7×7×1024
Conv / s1	1×1×1024×1024	7×7×1024
Avg Pool / s1	Pool 7×7	7×7×1024
FC / s1	1024×1000	1×1×1024
Softmax / s1	Classifier	1×1×1000

3) VGG16

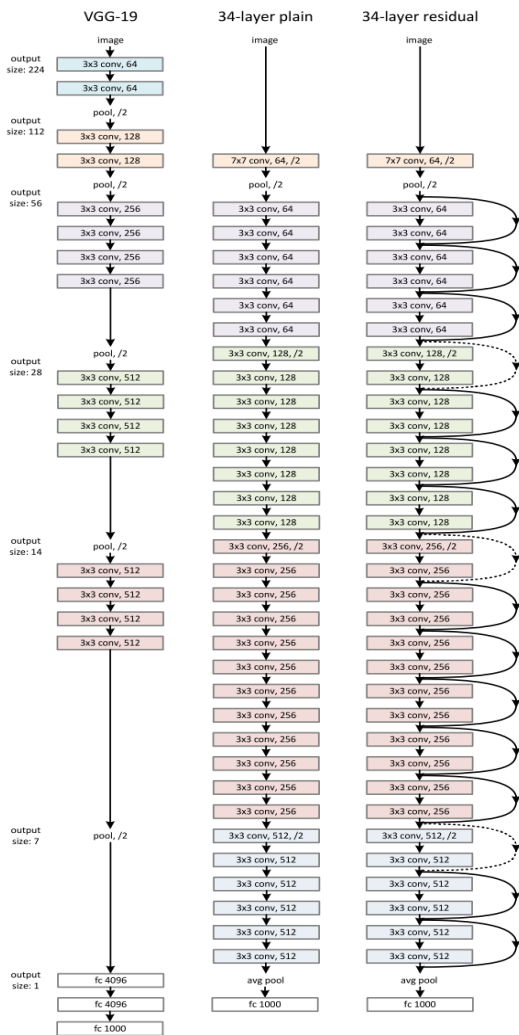
VGG16 adalah model jaringan saraf *convolutional* yang menggunakan konvolusi dengan ukuran yang kecil (3 x 3) serta menggunakan 16 layer *convolutional* [9].

Tabel 4. Arsitektur VGG16

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

4) ResNet101V2

Residual Neural Network atau biasa disingkat ResNet ini merupakan arsitektur yang dibuat oleh Kaiming He, Xiangyu Zhang, Shaoqing Ren dan Jian 5 Sun pada tahun 2015. Terdapat berbagai macam arsitektur ResNet, namun yang digunakan untuk penelitian ini adalah ResNet101V2.



Gambar 4. Arsitektur Resnet

5) DenseNet

Dengan DenseNet, setiap lapisan memperoleh masukan tambahan dari semua lapisan sebelumnya dan meneruskan peta fiturnya sendiri ke semua lapisan berikutnya dengan menggunakan penggabungan, maka setiap lapisan menerima "pengetahuan kolektif" dari semua lapisan sebelumnya [10].

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 x 112	7 x 7 conv, stride 2			
Pooling	56 x 56	3 x 3 max pool, stride 2			
Dense Block (1)	56 x 56	1 x 1 conv 3 x 3 conv	x 6	1 x 1 conv 3 x 3 conv	x 6
Transition Layer (1)	56 x 56	1 x 1 conv			
(1)	28 x 28	2 x 2 average pool, stride 2			
Dense Block (2)	28 x 28	1 x 1 conv 3 x 3 conv	x 12	1 x 1 conv 3 x 3 conv	x 12
Transition Layer (2)	28 x 28	1 x 1 conv			
(2)	14 x 14	2 x 2 average pool, stride 2			
Dense Block (3)	14 x 14	1 x 1 conv 3 x 3 conv	x 24	1 x 1 conv 3 x 3 conv	x 48
Transition Layer (3)	14 x 14	1 x 1 conv			
(3)	7 x 7	2 x 2 average pool, stride 2			
Dense Block (4)	7 x 7	1 x 1 conv 3 x 3 conv	x 16	1 x 1 conv 3 x 3 conv	x 32
Classification Layer	1 x 1	7 x 7 global average pool 1000D fully-connected, softmax			

Gambar 5. Arsitektur DenseNet

2.5. Pelatihan Model

Masing – masing *model* akan dilatih dengan tambahan *parameter* yang sama berikut uraian *parameter* yang digunakan :

- 1) *Learning rate* sebesar 0.001 [11]
- 2) *Jumlah epoch* adalah 10 [12]
- 3) Menggunakan *dropout* sebesar 0.45 [13]
- 4) Menggunakan dua macam *regularization*, yaitu *l1-regularization* dan *l2-regularization* [14,15]
- 5) Menambahkan layer *batch normalization* [13]
- 6) Fungsi aktivasi pada *layer convolutional* pada penelitian ini menggunakan *rectified linear unit (RELU)* [16]
- 7) *Optimizer* yang digunakan adalah Adam
- 8) *Loss function* yang digunakan *categorical crossentropy*
- 9) Fungsi aktivasi pada *layer* terakhir pada penelitian ini menggunakan *softmax* [17].

2.6. Evaluasi Model

Untuk mendapatkan *model* yang terbaik, maka perlu dilakukan evaluasi terhadap lima *model* tersebut. Evaluasi *model* dilakukan dengan mempertimbangkan *accuracy* yang didapat dari Persamaan (1) dan hasil dari *confusion matrix*. Selain itu *model* juga dievaluasi dengan *recall* yang didapat dari Persamaan

(2), dan *f1-score* yang didapat dari Persamaan (3).

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1-score = 2 \times \frac{precision \times accuracy}{precision+recall} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

3. HASIL DAN PEMBAHASAN

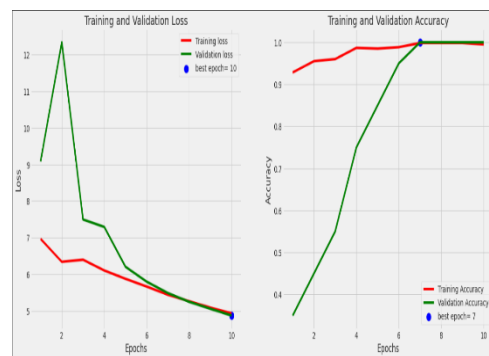
Langkah selanjutnya adalah melakukan pelatihan pada kelima *model* dengan data citra daun gandum. Tabel 5 menunjukkan hasil dari pelatihan yang dilakukan sebanyak 10 kali.

Tabel 5. Hasil evaluasi model

Model	Kelas	Performa				waktu pros
		Accuracy	Recall	f1-score	test accuracy (%)	
InceptionV3	Healthy	1	1	1		976 s
	septoria	1	1	1	100	
	stripe rust	1	1	1		
Mobilenet	Healthy	1	1	1		1406 s
	septoria	1	1	1	100	
	stripe rust	1	1	1		
VGG16	Healthy	0.71	1	0.83		1701 s
	septoria	0.62	1	0.77	76	
	stripe rust	1	0.55	0.71		
Resnet101V2	Healthy	1	1	1		1007 s
	septoria	1	1	1	100	
	stripe rust	1	1	1		
DenseNet	Healthy	1	1	1		1223 s
	septoria	1	1	1	95	
	stripe rust	1	1	1		

Pada Tabel 5 menunjukkan bahwa *model InceptionV3* mendapatkan nilai akurasi satu pada tebakan ketiga daun gandum, ini berarti model dapat mengenali dengan baik, dan saat di coba dengan *data test* juga menghasilkan akurasi yang sangat baik dengan mendapatkan 100%, dan pelatihan selama 10 iterasi dilakukan selama 976 detik. *Model MobileNet*, *Resnet101V2*, dan juga *DenseNet* juga menghasilkan akurasi

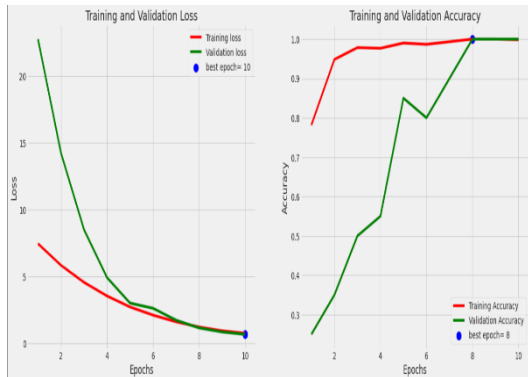
yang sama baiknya saat di tes dengan *data test* maupun *data train* dengan mendapatkan nilai akurasi, *recall*, presisi, dan *f1-score* sebesar satu, ini menandakan ketiga model itu berhasil menebak data citra dengan benar. Namun yang membedakan dari ketiga *model* tersebut adalah waktu komputasinya. *Model MobileNet* berhasil menyelesaikan pelatihan sebanyak 10 iterasi dengan waktu komputasi 1406 detik. *Model ResNet101V2* berhasil menyelesaikan pelatihan sebanyak 10 iterasi dengan waktu sedikit lebih cepat dari *MobileNet* yaitu selama 1007 detik. Dan *Model DenseNet* menyelesaikan pelatihan sebanyak 10 iterasi dengan waktu yang lebih lama dibandingkan dengan *ResNet101V2* dan lebih cepat dibanding dengan *MobileNet* yaitu selama 1223 detik. Hasil yang berbeda di hasilkan oleh Model *VGG16*, dimana model ini hanya menghasilkan akurasi, presisi, *recall*, dan *f1-score* sebesar 0.71, 1, 0.83 saat menebak data citra *healthy*, 0.62, 1, 0.77 saat menebak data citra daun gandum yang terkena septoria, 1, 0.55, 0.71 saat menebak data citra *stripe rust*. Dan juga saat di tes dengan data pelatihan *model VGG16* mendapatkan akurasi sebesar 76% dengan waktu komputasi yang lebih lama dibandingkan keempat model lainnya yaitu selama 1701 detik.



Gambar 6 Accuracy dan loss model InceptionV3

Gambar 6 terdapat dua buah grafik garis yang menunjukkan *loss* (grafik sebelah kiri) dan *accuracy* (grafik sebelah kanan) dari *model InceptionV3*. Terlihat pada grafik sebelah kiri *validation loss*-nya sempat naik

di iterasi kedua dan kembali turun lagi di iterasi ketiga lalu konvergen mendekati nol pada iterasi selanjutnya. Dan pada grafik sebelah kanan yang menunjukkan *accuracy*, *validation accuracy* dimulai dari sangat kecil lalu perlahan lahan berhasil mendekati *accuracy* dan puncak terbaiknya terdapat pada iterasi ke-7.



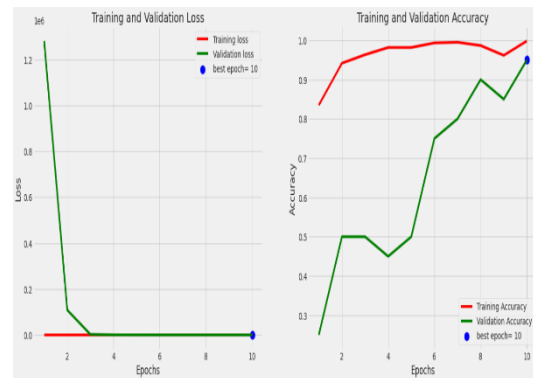
Gambar 7. Accuracy dan loss model MobileNet

Gambar 7 merupakan grafik *loss* (sebelah kiri) dan *accuracy* (sebelah kanan) dari model *MobileNet*. Dilihat dari grafik sebelah kanan *loss function* sudah konvergen mendekati nol sejak iterasi pertama dan pada grafik di sebelah kanan akurasi sempat turun di iterasi ke-6 lalu kembali naik lagi di iterasi selanjutnya, dan puncak terbaiknya terdapat pada iterasi ke-8.



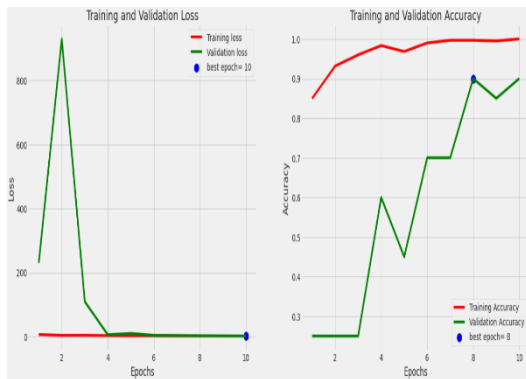
Gambar 8. Accuracy dan loss model VGG16

Gambar 8 menampilkan dua grafik yang berisi *loss* (sebelah kiri) dan *accuracy* (sebelah kanan) yang digambarkan dengan *line chart*. Terlihat bahwa *validation loss* yang digambarkan dengan garis berwarna hijau tidak bisa menyamai *loss* yang digambarkan dengan garis warna merah sampai iterasi terakhir. Dan juga *validation accuracy* yang digambarkan dengan garis hijau yang tidak juga bisa menyamai *accuracy* yang digambarkan dengan garis merah. Hal inilah yang membuat keadaan menjadi *overfit* dimana model tidak bisa mengenali data baru.



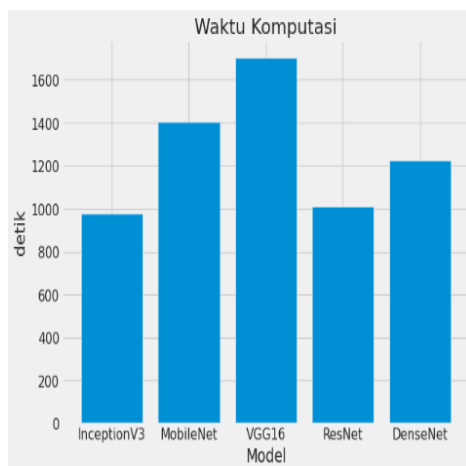
Gambar 9. Accuracy dan loss model ResNet101

Gambar 9 menampilkan dua grafik yang berisi *loss* (sebelah kiri) dan *accuracy* (sebelah kanan) yang digambarkan dengan *line chart*. Dilihat dari grafik sebelah kiri grafik *validation loss* konvergen mendekati nol dan pada iterasi ke-3 sudah tidak terjadi penurunan. Dan pada grafik sebelah kiri dapat dilihat bahwa walaupun *validation accuracy* mendekati satu tetapi di prosesnya terjadi naik turun dan mulai stabil naik pada iterasi ke-5.



Gambar 10. Accuracy dan loss model DenseNet

Gambar 10 menampilkan dua grafik yang berisi loss (sebelah kiri) dan accuracy (sebelah kanan) yang digambarkan dengan *line chart*. Terlihat pada grafik sebelah kiri *validation loss* sempat naik sangat tinggi pada iterasi kedua, walaupun akhirnya kembali konvergen mendekati 0. Dan pada grafik sebelah kanan *validation accuracy* juga mengalami fluktuasi dan mencapai puncak tertinggi pada iterasi ke-8.

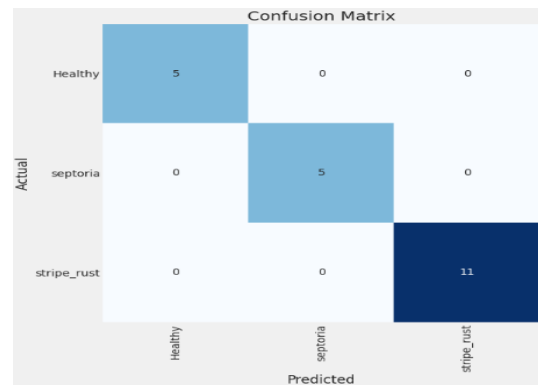


Gambar 11 Grafik perbandingan waktu komputasi lima model

Grafik batang yang ditunjukkan pada Gambar 11 memperlihatkan perbandingan waktu koputasi dari kelima *model*, terlihat *model VGG16* memerlukan waktu komputasi yang jauh lebih lama dibandingkan dengan *model* yang lain. Dan *model InceptionV3* memerlukan waktu

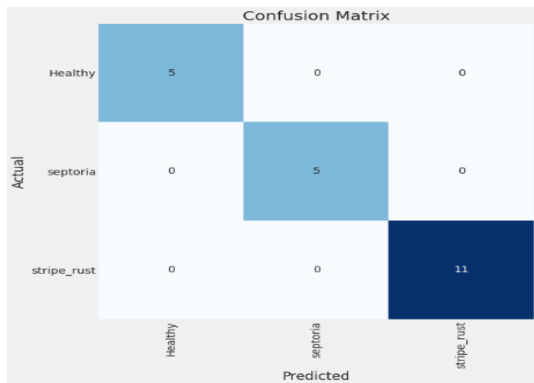
komputasi yang relatif lebih cepat dibanding *model* yang lain.

Untuk membuktikan akurasi *model* yang digambarkan grafik *line chart* pada gambar sebelumnya, kita bisa membandingkan hasil prediksi gambar citra yang dilakukan oleh model dengan gambar yang sebenarnya. Pada penelitian ini akan digunakan 21 data yang terdiri dari 11 gambar daun gandum yang terkena penyakit *stripe rust*, 5 gambar daun gandum yang terkena penyakit septoria, 5 daun gandum yang sehat untuk menguji keakuratan model.



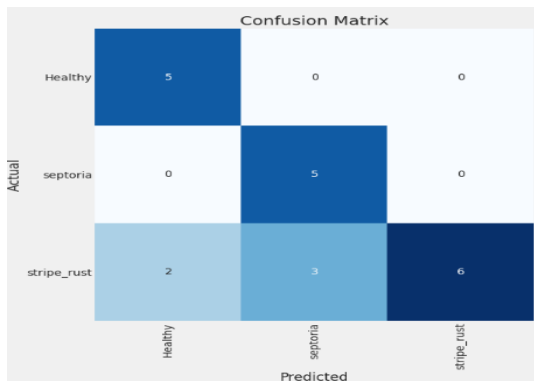
Gambar 12. Confusion matrix model InceptionV3

Model InceptionV3 berhasil memprediksi gambar dengan, 11 gambar daun gandum *stripe rust* berhasil di prediksi 11 kali. Daun gandum septoria berhasil diprediksi sebanyak 5 kali. Daun gandum yang sehat berhasil di prediksi sebanyak 5 kali seperti yang ditunjukkan pada Gambar 12.



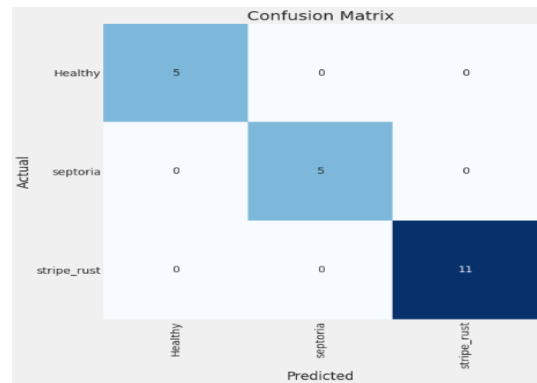
Gambar 13. Confusion matrix model MobileNet

Model MobileNet pada keseluruhan data yang diujikan juga berhasil menebak dengan baik, 11 data gambar stripe rust berhasil di prediksi semua, 5 gambar septoria berhasil diprediksi semua, dan 5 gambar daun yang sehat berhasil di prediksi semua.



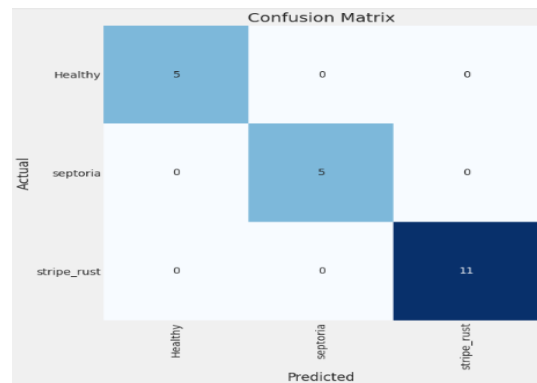
Gambar 14. Confusion matrix model VGG16

Pada Gambar 14 menunjukkan kalau model VGG16 kurang baik dalam memprediksi *data testing*. Pada gambar tersebut menunjukkan VGG16 kesulitan dalam memprediksi gambar yang seharusnya *stripe rust*. Dimana dari 11 gambar *stripe rust* model hanya berhasil memprediksi 6 gambar dengan benar, 3 gambar di prediksi sebagai septoria, dan 2 gambar diprediksi sebagai daun yang sehat.



Gambar 15. Confusion matrix ResNet101

Terlihat pada Gambar 15 model ResNet berhasil memrediksi semua data dengan benar. Citra daun stripe rust berhasil diprediksi 11 kali, septoria berhasil diprediksi 5 kali, dan daun yang sehat berhasil di prediksi 5 kali.



Gambar 16. Confusion matrix model DenseNet

Pada Gambar 16 model DenseNet juga berhasil menebak semua data dengan benar. Citra daun stripe rust berhasil diprediksi 11 kali, septoria berhasil diprediksi 5 kali, dan daun yang sehat berhasil di prediksi 5 kali.

4. KESIMPULAN

Penelitian ini diharapkan dapat membantu meningkatkan produksi gandum di Indonesia. Himpunan data yang digunakan adalah *wheat leaf datasets* yang diperoleh dari Kaggle, *dataset* tersebut berisi 3 buah kelas daun gandum yaitu daun sehat,

septoria, dan *stripe rust* dengan jumlah citra 208. Namun data tersebut imbalanced di setiap kelasnya, untuk itu dilakukan *oversampling* dengan metode *SMOTE*. Setelah data menjadi balance di setiap kelas lalu kita bagi menjadi 3 bagian yaitu *data train* sebanyak 600 data citra, *data test* sebanyak 21 citra, dan *data validation* sebanyak 20 citra. Kemudian *data train* digunakan untuk melatih lima *model pre-trained* dengan menambahkan paramater yang sudah disebutkan diatas selama 10 iterasi (*epochs*). Berdasarkan penelitian ini didapatkan *model Inception* mendapatkan *accuracy* sebesar 100% untuk data pelatihan dengan waktu komputasi sebesar 976 detik. *Model MobileNet* mendapatkan *accuracy* sebesar 100% untuk data pelatihan dengan waktu komputasi 1406 detik. *Model VGG16* mendapatkan *accuracy* 76% untuk data pelatihan dengan waktu komputasi 1701 detik. *Model ResNet101* mendapatkan *accuracy* sebesar 100% untuk data pelatihan dengan waktu komputasi 1007 detik, model *DenseNet* mendapatkan *accuracy* sebesar 95% untuk data pelatihan dengan waktu komputasi selama 1223 detik. Menimbang juga dengan *accuracy*, *recall*, dan *f1-score* dapat disimpulkan bahwa *model inceptionV3* merupakan *model* terbaik untuk memprediksi penyakit pada daun gandum. Dan dengan menimbang bentuk arsitekturnya *model mobilenet* lebih direkomendasikan jika ingin mengimplementasikan kedalam perangkat dengan komputasi yang rendah seperti perangkat *mobile*.

5. SARAN

Tentunya penelitian ini masih memiliki kekurangan, oleh karena itu peneliti ingin memberikan saran. Karena penelitian ini hanya sebatas mengembangkan dan membandingkan model deep learning untuk mendeteksi penyakit daun gandum, untuk itu pada penelitian selanjutnya dapat dikembangkan sebuah product yang bisa digunakan oleh end user dengan implementasi dari salah satu model diatas

DAFTAR PUSTAKA

- [1] N. Fitri, "Kepentingan Indonesia Melakukan Impor Gandum Dari Australia Tahun 2012-2016," *Jom Fisip*, vol. 6, pp. 1–8, 2019.
- [2] R. C. Downie *et al.*, "Septoria Nodorum Blotch of Wheat: Disease Management and Resistance Breeding in the Face of Shifting Disease Dynamics and a Changing Environment," *Phytopathology*®, p. PHYTO-07-20-028, 2021
- [3] M. Hon and N. M. Khan, "Towards Alzheimer's disease classification through transfer learning," *Proc. - 2017 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2017*, vol. 2017-Janua, pp. 1166–1169, 2017
- [4] R. Siringoringo, "Klasifikasi Data Tidak Seimbang Menggunakan Algoritma SMOTE dan k-Nearest Neighbor," *J. ISD*, vol. 3, no. 1, pp. 44–49, 2018.
- [5] B. Budiman, "Pendeteksian Penggunaan Masker Wajah Dengan Metode Convolutional Neural Network," *J. Ilmu Komput. dan Sist. Inf.*, vol. Vol.9 No.1, 2021.
- [6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.
- [7] N. Evans and A. Tedder, "Holographic model of hqadronization," *Phys. Rev. Lett.*, vol. 100, no. 16, 2008.
- [8] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017

- [9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.
- [10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017.
- [11] Darmatasia, "Analisis Perbandingan Performa Model Deep Learning untuk Mendeteksi Penggunaan Masker," *J. IT*, vol. 11, no. 2, pp. 101–107, 2020.
- [12] A. J. Rozaqi, A. Sunyoto, and M. rudyanto Arief, "Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network," *Creat. Inf. Technol. J.*, vol. 8, no. 1, p. 22, 2021.
- [13] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimed. Tools Appl.*, vol. 79, no. 19–20, pp. 12777–12815, 2020.
- [14] Y. Zhai *et al.*, "Robust SAR Automatic Target Recognition Based on Transferred MS-CNN with L2-Regularization," *Comput. Intell. Neurosci.*, vol. 2019.
- [15] C. Yang *et al.*, "Structured Pruning of Convolutional Neural Networks via L1 Regularization," *IEEE Access*, vol. 7, pp. 106385–106394, 2019.
- [16] T. R. Savera, W. H. Suryawan, and A. W. Setiawan, "Deteksi Dini Kanker Kulit Menggunakan K-Nn Dan Early Detection of Skin Cancer Using K-Nn and Convolutional," vol. 7, no. 2, pp. 373–378, 2020.
- [17] I. M. Dheir, A. Soliman, A. Mettleq, A. A. Elsharif, and S. S. Abu-Naser, "Classifying Nuts Types Using Convolutional Neural Network," *Int. J. Acad. Inf. Syst. Res.*, vol. 3, no. 12, pp. 12–18, 2020.