

Implementasi VGG 16 dan Augmentasi Zoom Untuk Klasifikasi Kematangan Sawit

T Kaisyarendika Mazdavilaya, Febi Yanto*, Elvia Budianita, Suwanto Sanjaya, Fadhilah Syafria

Fakultas Sains dan Teknologi, Prodi Teknik Informatika, Universitas Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia

Email: ¹12050112561@students.uin-suska.ac.id, ^{2,*}febiyanto@uin-suska.ac.id, ³elviabudianita@uin-suska.ac.id,

⁴suwantosanjaya@uin-suska.ac.id, ⁵fadhilah.syafria@uin-suska.ac.id

Email Penulis Korespondensi: febiyanto@uinsuska.ac.id

Abstrak—Indonesia merupakan negara yang memiliki hasil perkebunan kelapa sawit yang sangat melimpah dan menjadikan kelapa sawit sebagai salah satu komoditas export terbesar di Indonesia. Kematangan buah pada kelapa sawit memiliki pengaruh yang signifikan terhadap produksi minyak sawit dan inti. Tingkat kematangan pada buah kelapa sawit dapat mempengaruhi beberapa kandungan yang ada didalamnya seperti kandungan tokoferol, Klasifikasi akan dibagi menjadi 2 kelas yaitu antara buah yang matang dan tidak matang dengan data buah matang sebanyak 500 gambar dan tidak matang 500 gambar yang mana data diambil dari situs kaggle dan kebun pribadi yang diambil menggunakan kamera hp. Data yang sudah didapatkan dilakukan augmentasi yang berguna untuk memperkaya data agar lebih banyak. Augmentasi data menggunakan augmentasi zoom dan membuat data yang semula 1000 bertambah menjadi 2000 data. Model yang digunakan adalah VGG 16 yang merupakan bagian dari deep learning. Dataset yang sudah ada selanjutnya dilakukan preprocessing *resize* dan *rescale*, selanjutnya melakukan pembagian data menjadi 3 yaitu data *train*, *test* dan *valid*. Setelah melakukan pembagian data selanjutnya melakukan proses klasifikasi dengan VGG 16 dan mengatur hyperparameternya setelah itu model akan belajar dengan 20 epoch. Model akan melakukan pelajaran dengan 57 skema untuk dibandingkan dan mencari akurasi tertingginya. Setelah model selesai belajar dilakukan evaluasi dengan confusion matrix. Didapatkan hasil bahwa Pembagian data 90:10 menggunakan augmentasi data dengan learning rate 0,01 dan dropout 0,001 mendapatkan akurasi terbaik yaitumencapai 93,8%.

Kata Kunci: VGG16; Augmentasi; Hyperparameter; Deep learning; Random Zoom

Abstract—Indonesia is a country that has very abundant palm oil plantations and makes palm oil one of the largest export commodities in Indonesia. Fruit maturity on oil palms has a significant influence on palm oil and kernel production. The level of ripeness in palm oil fruit can affect several contents in it, such as tocopherol content, yield and FFA. The classification will be divided into 2 classes, namely between ripe and immature fruit with data on 500 images of ripe fruit and 500 images of immature fruit, data taken from the Kaggle site and private gardens taken using a cellphone camera. The data that has been obtained is augmented which is useful for enriching the data to make it more abundant. Data augmentation uses zoom augmentation and makes the original 1000 data increase to 2000 data. The model used is VGG 16 which is part of deep learning. The existing dataset is then preprocessed, resized and rescaled, then divides the data into 3, namely train, test and valid data. After dividing the data, then carry out the classification process with VGG 16 and set the hyperparameters after that the model will learn with 20 epochs. The model will learn with 57 schemes to compare and find highest accuracy. After the model has finished learning, it is evaluated using a confusion matrix. The results obtained were that the 90:10 data division using data augmentation with a learning rate of 0.01 and a dropout of 0.001 obtained the best accuracy, reaching 93.8%.

Keywords: VGG16; Augmentation; Hyperparameter; Deep learning; Random Zoom

1. PENDAHULUAN

Indonesia merupakan negara yang memiliki hasil perkebunan kelapa sawit yang sangat melimpah dan menjadikan kelapa sawit sebagai salah satu komoditas export terbesar di Indonesia dan juga menjadi kontributor terbesar di dunia sebagai penyedia produk olahan kelapa sawit yang mana pada tahun 2016 Indonesia telah mengeksport sebanyak 31,1 juta ton dan juga per tahunnya menyumbang devisa rata-rata sebesar 15 miliar dollar AS. Kematangan buah pada kelapa sawit memiliki pengaruh yang signifikan terhadap produksi minyak sawit dan inti [1]. Buah pada kelapa sawit yang matang memiliki warna cerah yang didominasi oleh warna kuning kemerahan sedangkan buah yang belum matang berwarna ungu tua. Pada umumnya buah kelapa sawit sudah matang jika sudah 6 bulan sejak proses penyerbukan yang memiliki tanda tanda warna sudah mulai menguning kemerahan [2]. Tingkat kematangan pada buah kelapa sawit dapat mempengaruhi kadar tokoferol di dalamnya [3]. Yang mana kandungan tokoferol tinggi pada buah yang matang dapat membuat minyak tidak mudah tengik [4]. Kematangan pada buah kelapa sawit juga mempengaruhi tingkat kandungan rendemen dan komponen asam lemak. Semakin tinggi tingkat kematangan buah maka semakin tinggi rendemen yang dihasilkan dan komponen asam lemak yang dikandungnya. FFA atau free fatty acid juga terpengaruh oleh tingkat kematangan dari buah. Kadar FFA akan semakin meningkat berdasarkan tingkat kematangan buah. Rendemen dan FFA memiliki pengaruh terhadap CPO (Crude Palm Oil) karena FFA yang terlalu banyak dan rendemen yang terlalu rendah yang terkandung dalam buah sawit dapat menurunkan kualitas CPO [5].

Machine learning merupakan bagian dari AI yang memungkinkan computer untuk belajar dengan data yang ada tanpa harus diprogram secara eksplisit. *Machine learning* memiliki sub kategori yaitu *deep learning* yang menggunakan jaringan *deep neural networks* dengan menggunakan lebih banyak lapisan untuk memodelkan fungsi dan melakukan penangkapan fitur tingkat tinggi di data. Keunggulan dari *deep learning* dibandingkan *machine learning* ada pada kemampuannya yang dapat secara otomatis untuk melakukan ekstrak fitur-fitur representative mendalam dari data yang mentah, memungkinkan menangkap informasi semantic yang berhubungan dengan tugas yang laten. *Deep learning* juga dapat menyelesaikan sebuah masalah yang tidak terstruktur dalam skala besar dan heterogen dan dapat belajar dari data

seperti gambar, audio dan teks [6]. Dengan keunggulan tersebut sekarang banyak yang menggunakan *deep learning* sebagai metode penelitian yang berkaitan dengan hal klasifikasi [7]. Pada penelitian yang dilakukan untuk mengklasifikasi makanan yang menggunakan 2 metode yaitu SqueezeNet dan VGG 16 mendapatkan hasil yang baik dari VGG 16 dengan akurasi hingga 85,07% [8]. Penelitian menggunakan VGG 16 juga dilakukan oleh [9] untuk mengklasifikasi penyakit pada daun tanaman yang mana VGG 16 mendapatkan hasil dengan akurasi mencapai 98,40% yang mana akurasi ini paling tinggi dibandingkan dengan *deep learning* lainnya yang sudah dicoba.

Metode augmentasi data merupakan teknik yang berfungsi pada tingkat data bukan pada tingkat arsitektur model. Augmentasi dapat membantu performa dari model *deep learning* menjadi lebih baik dengan menciptakan sampel yang berbeda dan beragam secara artifisial dengan kelas yang seimbang untuk melatih data [10]. Penggunaan augmentasi data ini dapat meningkatkan akurasi hingga 1,5% [11]. Untuk membuktikan ini [12] melakukan perbandingan terhadap model yang menggunakan augmentasi data dan yang tidak yang mana model yang menggunakan augmentasi data memiliki akurasi yang lebih tinggi dibandingkan model yang tidak menggunakan augmentasi data. Akurasi model yang menggunakan data augmentasi adalah 97,12% sedangkan yang tidak menggunakan augmentasi mendapatkan hasil 92,87%. Kemudian [13] juga melakukan penelitian dengan membandingkan sebelum dan sesudah melakukan augmentasi data ke 3 arsitekturnya. Hasilnya semua arsitektur mengalami peningkatan akurasi setelah melakukan augmentasi data. ResNet yang awalnya memiliki akurasi 73,6% naik menjadi 87,8%, Vgg-16 yang akurasi awalnya 75,9% naik menjadi 93,5% dan fine tuned Vgg-19 awalnya memiliki akurasi sebesar 81,3% dan naik menjadi 96,1%. Zoom merupakan bagian dari data augmentasi tradisional geometric. Augmentasi zoom melakukan duplikasi data dengan melakukan zoom in atau zoom out. Zoom dapat meningkatkan akurasi lebih baik dibandingkan augmentasi data lainnya [14].

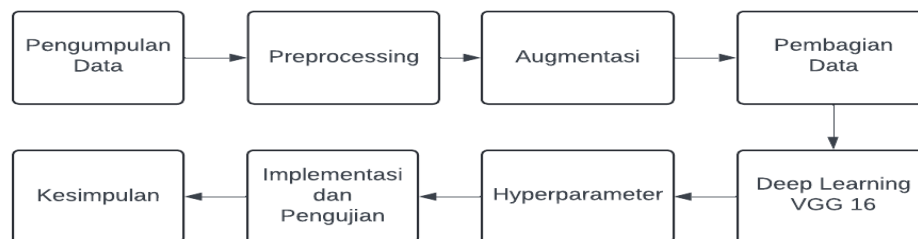
Selain augmentasi data yang digunakan pada deep learning untuk meningkatkan keragaman dataset, ada juga algoritma yang dibutuhkan untuk meminimalisasi *error* pada operasinya yaitu optimizer [15]. Memilih optimizer yang bagus merupakan hal yang penting dan sedikit sulit karena ada berbagai macam pilihan optimizer. RAdam merupakan optimizer yang bagus dalam mengolah data dalam skala besar [16]. Dengan menggunakan RAdam dapat dengan efektif memperbaiki akurasi sistem dan mengurangi waktu yang diperlukan [17]. RAdam juga tidak mudah jatuh ke dalam solusi optimal lokal dan dapat menyatu dengan cepat [18]. Untuk membuktikan ini [19] menggunakan optimasi RAdam dalam 3 arsitekturnya yaitu VGG16, ResNet dan InceptionV3 mendapatkan hasil bahwa akurasi mengalami peningkatan sebesar 3-4%. Dalam penelitian yang dilakukan oleh [20] menggunakan 4 arsitektur yaitu ResNet34, Resnet50, Vgg19 dan DenseNet121 menggunakan optimasi RAdam untuk mengklasifikasi kanker. Dengan menggunakan optimasi RAdam keempat arsitektur itu mendapatkan akurasi di atas 90% dan DenseNet 121 dengan optimasi RAdam yang memiliki akurasi paling tinggi yaitu 98%.

Berdasarkan latar belakang di atas yang sudah dijabarkan maka ditetapkan lah bahwa penelitian ini akan melakukan pembuatan sistem klasifikasi kematangan buah kelapa sawit dengan menggunakan pendekatan *deep learning* dengan menggunakan salah satu arsitektur dari CNN yaitu VGG 16 dan menggunakan augmentasi data *zoom* agar mendapatkan dataset yang beragam dan menggunakan optimasi RAdam untuk meminimalisasi *error* supaya mendapatkan hasil akurasi yang maksimal.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Tahapan penelitian dilakukan seperti pada **Gambar 1**. Pertama melakukan pengumpulan data citra sawit selanjutnya melakukan preprocessing pada data citra yang kemudian dilakukan augmentasi data agar data citra menjadi lebih banyak. Setelah dilakukan augmentasi berikutnya data dibagi menjadi 3 yaitu data *train*, *valid* dan *test* kemudian dilakukan proses klasifikasi dengan menggunakan VGG 16 dan dilanjut dengan mengatur hyperparameternya. Setelah itu dilakukan implementasi dan pengujian yang mana penelitian ini menguji antara data yang menggunakan augmentasi dan memiliki 57 skema dengan mencari komposisi learning rate dan dropout yang bisa mendapatkan akurasi terbaik.



Gambar 1. Tahapan Penelitian

2.2 Pengumpulan Data

Pada tahap ini data yang digunakan memiliki 2 kelas yaitu gambar sawit yang sudah matang dan yang belum matang. Data citra ini didapatkan dari hasil foto sendiri dengan kamera HP di kebun pribadi sebagai data primer. Pada data primer ini buah sawit yang difoto adalah buah yang belum matang dan masih berada pohon. Data sekunder diambil dari situs

kaggle dengan alamat url nya <https://www.kaggle.com/datasets/nabylalfahrezra/sawitrangers>. Jumlah data ada 1000 gambar dengan pembagian 500 sawit matang dan 500 sawit belum matang. Gambar sawit pada data ini ada yang diambil dengan posisi buah masih ada dipohon jika buah itu belum matang dan ada foto buah sawit yang di tanah baik yang sudah matang maupun belum matang.

2.3 Preprocessing

Pada tahap preprocessing gambar akan diresize yaitu ukuran gambar diubah dari yang asli menjadi gambar ukuran 224x224 sesuai dengan ukuran gambar inputan VGG 16. Selain *resize* dilakukan juga *rescale* untuk menormalisasi citra dari dataset dengan membagi piksel yang terkecil dengan piksel yang terbesar. *Rescale* dilakukan untuk mengubah rentang intensitas piksel dari 0-255 dalam skala RGB menjadi lebih kecil seperti 0-1 atau dari -1 sampai 1 [21].

2.4 Augmentasi Data

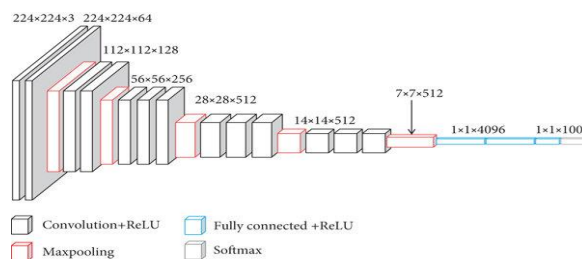
Dataset yang sudah dikumpulkan sebelumnya diperbanyak gambarnya dengan menggunakan metode augmentasi data. Augmentasi data yang digunakan adalah metode augmentasi *zoom* yang dapat dengan signifikan meningkatkan akurasi data dibandingkan metode augmentasi lainnya [14]. Augmentasi dilakukan dengan menggunakan *library* augmentor. Jumlah augmentasi data yang dilakukan sebanyak 1000 gambar.

2.5 Pembagian Data

Pada tahapan ini data dibagi menjadi 3 yaitu data *train*, *valid* dan *test*. Pembagian data ini dibagi dengan data *valid* yang diambil 20% terlebih dahulu setelah itu sisa datanya akan dibagi menjadi data *train* dan data *test* dengan beberapa skema yaitu 90:10, 80:20 dan 70:30. Skema ini akan dilakukan pada data yang sudah dilakukan augmentasi dan tidak dilakukan augmentasi.

2.6 VGG 16

VGG 16 telah terpilih menjadi arsitektur dengan performa terbaik di ILSVRC tahun 2014 dengan varian CNN yang lebih dalam tetapi lebih simple [8]. VGG 16 merupakan jaringan luas yang memiliki parameter sekitar 138 juta. Arsitektur dari VGG 16 yang ditunjukkan pada **gambar 2**. Pertama gambar diinput ke dalam *input layer* dengan gambar berukuran 224x224. Di **gambar 2** memiliki 5 blok konvolusi yang berurutan dengan diakhiri oleh *maxpooling* disetiap bloknya. Disetiap blok memiliki filter dengan masing-masing ukuran berurutan 64,128,256,512,512. Pada bagian akhirnya ada *fully connected layers*. Pada akhir lapisan dense menggunakan fungsi aktivasi softmax yang bisa mengeluarkan hasil pada 1000 kategori berbeda [22].



Gambar 2. Arsitektur VGG 16

2.7 Hyperparameter

Hyperparameter adalah parameter yang sudah ditetapkan nilainya sebelum dilakukannya proses *training*. Tujuan dari hyperparameter adalah untuk menemukan nilai parameter yang sesuai agar mendapatkan hasil model yang maksimal. Salah satu bagian dari hyperparameter adalah optimizer. Optimizer adalah algoritma atau metode yang digunakan untuk mencari bobot terbaik, meminimalisasi *error*, memaksimalkan akurasi dan mengurangi nilai kerugian atau *loss* pada model yang dibuat agar hasil prediksi yang dikeluarkan akurat. Optimizer memiliki beberapa variasi dan salah satunya yang paling populer digunakan adalah Adam [23]. Radam (Rectified Adaptive Moment Estimation) adalah salah satu optimasi terbaru yang tercipta dari modifikasi optimasi Adam. Radam memiliki efek yang sangat bagus dengan karakteristiknya yang cepat konvergen dan akurasi yang tinggi [17]. Radam memiliki ketahanan terhadap perubahan kecepatan pembelajaran dan dapat beradaptasi dengan kecepatan pembelajaran yang berbeda dengan rentang yang luas [16]. Pada penelitian ini menggunakan 3 *learning rate* yaitu 0,1 0,01 dan 0,001. *Dropout* yang digunakan adalah 0,1 0,01 dan 0,001 dengan epoch sebanyak 20.

2.8 Confusion matrix

Confusion matrix digunakan dalam penelitian ini untuk mengukur performa dari model ini. Confusion matrix diperlukan untuk melakukan proses pengukuran pada model karena nilai kinerja klasifikasi pada dataset tidak semua hasilnya menyentuh akurasi 100% [24]. Pada **tabel 1** confusion matrix memiliki 4 kategori hasil prediksi. True Positive (TP) adalah Jumlah observasi yang secara benar diklasifikasikan sebagai positif oleh model, True Negative (TN) adalah Jumlah observasi yang secara benar diklasifikasikan sebagai negatif oleh model, False Positive (FP) yaitu Jumlah observasi yang

seharusnya diklasifikasikan sebagai negatif, tetapi salah diklasifikasikan sebagai positif oleh model juga dikenal sebagai kesalahan Type I, False Negative (FN) Jumlah observasi yang seharusnya diklasifikasikan sebagai positif, tetapi salah diklasifikasikan sebagai negatif oleh model. Juga dikenal sebagai kesalahan Type II.

Tabel 1. Confusion Matrix

Kelas Benar	Sawit Matang	Sawit Belum Matang
Sawit matang	True Positive	False Positive
Sawit Belum Matang	False Positive	False Negative

Dari confusion matrix lah didapatkan nilai *accuracy*, *precision*, *recall* dan *f1 score*. *Accuracy* merupakan salah satu dari matrix untuk mengevaluasi model klasifikasi. *Accuracy* adalah rasio dari gambar yang diberi label dengan benar dari total seluruh sampel yang ada.

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

precision adalah rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Dari semua kelas positif yang telah diprediksi dengan benar, berapa banyak data yang benar-benar positif.

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

Recall atau *sensivity* adalah akurasi yang diprediksi secara positif yang menjelaskan berapa banyak yang diberi label dengan benar

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

F1 Score adalah perbandingan rata-rata dari *precision* dan *recall* yang dibobotkan.

$$f1\ score = 2x \frac{Precision \times Recall}{(Precision+Recall)} \tag{4}$$

Hasil dari evaluasi ini akan membandingkan model yang menggunakan augmentasi data dan tidak. Masing masing dari model juga akan menggunakan rasio pembagain data yang berbeda yaitu (90:10), (80:20), dan (70:30).

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Gambar buah sawit diambil dengan menggunakan kamera pada kebun pribadi dan data juga diambil dari situs kaggle. Data yang diperoleh berjumlah 1000 gambar dengan membaginya menjadi 2 kelas berbeda yaitu kelas sawit matang dan sawit belum matang. Pada kelas sawit matang berisikan 500 gambar dan begitu juga pada sawit tidak matang diberikan 500 gambar. Kemudian untuk memperbanyak dataset dengan menggunakan metode augmentasi data *zoom*. Gambar asli diambil kemudian dilakukan teknik *random zoom* sehingga total gambar bertambah 1000. Jumlah dataset setelah menggunakan augmentasi menjadi 2000 gambar.

3.2 Preprocessing

Preprocessing menggunakan *rescaling* dan *resize*. Pada **Gambar 3** menunjukan hasil dari gambar yang sudah dilakukan *preprocessing*. *resize* digunakan untuk mengubah ukuran citra menjadi 224x224 yang sesuai dengan ukuran inputan dari algoritma VGG16. *Rescaling* dilakukan untuk mengubah skala citra warna yang sebelumnya rentang warna 0-255 menjadi 0-1.



Gambar 3. Hasil preprocessing

3.3 Pembagian data

Pembagian data dilakukan dengan menggunakan 3 skema percobaan seperti di **tabel 2**. Data dibagi menjadi data *train*, *data valid* dan *data test*. Data validasi diambil terlebih dahulu sebanyak 20% dari total citra gambar sawit pada dua kelas sawit matang dan belum matang. Kemudian sisa data dibagi menjadi data *train* dan data *test* dibagi menjadi (90:10), (80:20) dan (70:30). Pada data yang tidak diaugmentasi memiliki data valid berjumlah 200. Untuk pembagian data 90:10 data *train* mendapatkan 720 gambar dan data *test* memiliki 80. Pada pembagian 80:20 data *train* berjumlah 640 dan *testnya* 160. Pada pembagian 70:30 data *train* memiliki 560 gambar dan *testnya* 240 gambar. Untuk data yang sudah dilakukan augmentasi memiliki lebih banyak data validnya karena gambar sudah ditambah. Gambar yang ada pada data valid yaitu 400 gambar. Untuk pembagian data 90:10 data *train* memiliki 1440 gambar dan data *test* memiliki 160. Pada pembagian 80:20 data *train* berjumlah 1280 dan *testnya* 320. Pada pembagian 70:30 data *train* memiliki 1120 gambar dan *testnya* 480 gambar.

Tabel 2. Pembagian Data

Pembagian Data	Data Train	Data Test	Data Valid
Tidak Augmentasi			
(90:10):20	720	80	200
(80:20):20	640	160	200
(70:30):20	560	240	200
Augmentasi			
(90:10):20	1440	160	400
(80:20):20	1280	320	400
(70:30):20	1120	480	400

3.4 Implementasi VGG 16

Pengujian menggunakan *Library* Tensorflow yang ada pada Keras dengan menggunakan bahasa pemrograman Python. Pengujian dilakukan pada Google Colab dengan memanggil *library* VGG16. Pengujian dilakukan dengan 57 Skema dengan mengubah parameter uji yaitu *Learning rate* dan nilai *dropout*. *Learning rate* yang digunakan 0,1 0,001 dan 0,001. Untuk *dropout* yang digunakan 0,1 0,01 dan 0,001. *Library* algoritma VGG16 yang digunakan dari tensorflow ada pada **gambar 4**.

```

1. from keras.applications import VGG16
2. base_model=tf.keras.applications.VGG16(include_top=False,
weights="imagenet",input_tensor=Input(shape=(224,224,3)))
weights="imagenet",input_tensor=Input(shape=(299,299,3)))
    
```

Gambar 4. Implementasi VGG 16

3.5 Pembahasan

Berikut tabel 3 merupakan skema pengujian.

Tabel 3. Skema Pengujian

No	Pembagian Data	Learning rate	Dropout	Augmentasi	Epoch
1	90:10	tidak	tidak	tidak	20
2	90:10	0,001	0,001	iya	20
3	90:10	0,001	0,01	iya	20
4	90:10	0,001	0,1	iya	20
5	90:10	0,01	0,001	iya	20
6	90:10	0,01	0,01	iya	20
7	90:10	0,01	0,1	iya	20
8	90:10	0,1	0,001	iya	20
9	90:10	0,1	0,01	iya	20
10	90:10	0,1	0,1	iya	20
11	90:10	0,001	0,001	tidak	20
12	90:10	0,001	0,01	tidak	20
13	90:10	0,001	0,1	tidak	20
14	90:10	0,01	0,001	tidak	20
15	90:10	0,01	0,01	tidak	20
16	90:10	0,01	0,1	tidak	20
17	90:10	0,1	0,001	tidak	20
18	90:10	0,1	0,01	tidak	20

19	90:10	0,1	0,1	tidak	20
20	80:20	tidak	tidak	tidak	20
21	80:20	0,001	0,001	iya	20
22	80:20	0,001	0,01	iya	20
23	80:20	0,001	0,1	iya	20
24	80:20	0,01	0,001	iya	20
25	80:20	0,01	0,01	iya	20
26	80:20	0,01	0,1	iya	20
27	80:20	0,1	0,001	iya	20
28	80:20	0,1	0,01	iya	20
29	80:20	0,1	0,1	iya	20
30	80:20	0,001	0,001	tidak	20
31	80:20	0,001	0,01	tidak	20
32	80:20	0,001	0,1	tidak	20
33	80:20	0,01	0,001	tidak	20
34	80:20	0,01	0,01	tidak	20
35	80:20	0,01	0,1	tidak	20
36	80:20	0,1	0,001	tidak	20
37	80:20	0,1	0,01	tidak	20
38	80:20	0,1	0,1	tidak	20
39	70:30	tidak	tidak	tidak	20
40	70:30	0,001	0,001	iya	20
41	70:30	0,001	0,01	iya	20
42	70:30	0,001	0,1	iya	20
43	70:30	0,01	0,001	iya	20
44	70:30	0,01	0,01	iya	20
45	70:30	0,01	0,1	iya	20
46	70:30	0,1	0,001	iya	20
47	70:30	0,1	0,01	iya	20
48	70:30	0,1	0,1	iya	20
49	70:30	0,001	0,001	tidak	20
50	70:30	0,001	0,01	tidak	20
51	70:30	0,001	0,1	tidak	20
52	70:30	0,01	0,001	tidak	20
53	70:30	0,01	0,01	tidak	20
54	70:30	0,01	0,1	tidak	20
55	70:30	0,1	0,001	tidak	20
56	70:30	0,1	0,01	tidak	20
57	70:30	0,1	0,1	tidak	20

Pada Tabel 3 dapat dilihat jika pengujian akan dilakukan dengan 57 skema dengan pembagian data 90:10, 80:20, dan 70:30 dengan menggunakan optimizer RAdam dan dropout dan tanpa menggunakan optimizer dan *dropout* yang setiap pengujian menggunakan 20 *epoch*. Nilai parameter pada optimizer RAdam ditentukan pada *learning rate*. *Learning rate* dan *dropout* yang digunakan berbeda-beda yaitu 0,1 0,01 dan 0,001. Pengujian juga dilakukan dengan 2 kondisi berbeda yaitu dengan menggunakan data augmentasi dan tidak menggunakan data augmentasi. Hasil dari 57 skema pengujian yang telah dilakukan dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian

No	Pembagian Data	Learning rate	Dropout	Augmentasi	Tanpa Augmentasi
1	90:10	-	-	-	86,3%
2		0,001	0,001	93,12%	85,0%
3		0,001	0,01	93,12%	80,0%
4		0,001	0,1	91,9%	81,3%
5		0,01	0,001	93,8%	85,0%
6		0,01	0,01	90,6%	83,8%
7		0,01	0,1	92,5%	81%
8		0,1	0,001	85,6%	86,3%
9		0,1	0,01	82,5%	82,5%
10	80:20	0,1	0,1	83,8%	77,5%
11		-	-	-	85,6%
12		0,001	0,001	91,9%	88,1%
13		0,001	0,01	92,2%	86,9%

14		0,001	0,1	91,6%	85,0%
15		0,01	0,001	91,87%	88,8%
16		0,01	0,01	90,0%	83,1%
17		0,01	0,1	90,6%	89,4%
18		0,1	0,001	80,62%	85,0%
19		0,1	0,01	85,62%	86,3%
20		0,1	0,1	81,7%	86,9%
21	70:30	-	-	-	85,42%
22		0,001	0,001	90,83%	87,92%
23		0,001	0,01	92,80%	87,08%
24		0,001	0,1	90,8%	86,25%
25		0,01	0,001	92,3%	86,67%
26		0,01	0,01	92,92%	86,25%
27		0,01	0,1	93,12%	86,67%
28		0,1	0,001	85,6%	64,6%
29		0,1	0,01	86,5%	82,9%
30		0,1	0,1	84,6%	85,4%

Pada pembagian data 90:10 dengan tidak menggunakan *learning rate*, *dropout* dan augmentasi mendapatkan akurasi sebesar 86,3%. Kemudian pada pembagian data 90:10 ditambahkan lah hyperparameter yaitu *learning rate* dan *dropout*. Hasil yang didapatkan setelah menambahkan *learning rate* dan *dropout* mengalami perubahan yang berbeda beda tergantung nilai parameter pada *learning rate* dan *dropout* yang digunakan. Akurasi yang didapatkan ada yang naik dan turun dengan akurasi tertinggi adalah 86,3% dengan nilai *learning rate* 0,1 dan *dropout* 0,001 yang mana akurasi ini sama dengan akurasi saat tidak belum menambahkan *learning rate* dan *dropout*. Sedangkan akurasi terendahnya ada pada angka 77,5% dengan menggunakan *learning rate* 0,1 dan *dropout* 0,1.

Setelah menambahkan *learning rate* dan *dropout* pada model dengan pembagian 90:10, dilanjut dengan menambahkan augmentasi data. Setelah menambahkan augmentasi data akurasi yang didapatkan kebanyakan memiliki hasil yang lebih baik dibandingkan dengan akurasi model yang tidak menggunakan augmentasi data dengan *learning rate* dan *dropout* yang sama dan hanya 1 model saja yang menggunakan augmentasi data memiliki akurasi yang lebih rendah seperti yang terlihat pada Tabel 4. Akurasi terbaik setelah menggunakan augmentasi data memiliki nilai 93,8% dengan menggunakan *learning rate* 0,01 dan *dropout* 0,001 sedangkan akurasi terburuknya adalah 82,5% yang menggunakan *learning rate* 0,1 dan *dropout* 0,01.

Pada pembagian data 80:20 dilakukan hal yang sama yaitu awalnya dilakukan pengujian dengan membagi data menjadi 80% data *train* dan 20% data *test* tanpa menggunakan augmentasi, optimizer RAdam dan *dropout* yang mendapatkan akurasi sebesar 85,6%. Kemudian dilakukan pengujian selanjutnya dengan menggunakan optimizer dan *dropout* yang mana hasil yang didapatkan banyak yang mengalami peningkatan akurasi dibandingkan tidak menggunakan optimizer dan *dropout*. Akurasi tertingginya ada pada angka 89,4% yang menggunakan *learning rate* 0,01 dan *dropout* 0,1 sedangkan akurasi terendahnya ada di angka 83,1%. Selanjutnya ditambahkan augmentasi data ke dalamnya yang mendapatkan hasil yang lebih baik lagi. Akurasi tertinggi setelah menggunakan augmentasi data ada di *learning rate* 0,001 dan *dropout* 0,01 dengan akurasi sebesar 92,2% yang mana akurasinya lebih tinggi dibandingkan dengan akurasi tertinggi yang dapat dicapai jika tidak menggunakan augmentasi. Sedangkan akurasi terendahnya ada pada angka 80,62% yang menggunakan *learning rate* 0,1 dan *dropout* 0,001.

Selanjutnya terakhir dilakukan pada pembagian data 70:30 dengan kondisi yang sama tanpa optimizer, *dropout* dan augmentasi dan mendapatkan akurasi sebesar 85,42%. Untuk akurasi tertinggi yang menggunakan optimizer dan *dropout* tetapi tanpa augmentasi akurasi mencapai 87,92% yang menggunakan *learning rate* 0,001 dan *dropout* 0,001 yang mana akurasi ini mengalami kenaikan dibandingkan dengan akurasi yang tidak menggunakan optimizer dan *dropout*. Untuk akurasi terendahnya terjadi di *learning rate* 0,1 dan *dropout* 0,001 yaitu 64,6%. Setelah ditambahkan augmentasi data akurasi tertinggi dapat mencapai 93,12% dengan menggunakan *learning rate* 0,01 dan *dropout* 0,1. Akurasi ini juga lebih tinggi jika dibandingkan dengan akurasi tertinggi yang didapatkan oleh model yang tidak menggunakan augmentasi. Kemudian untuk akurasi terendahnya ada pada *learning rate* 0,1 dengan *dropout* 0,1 yaitu 84,6%.

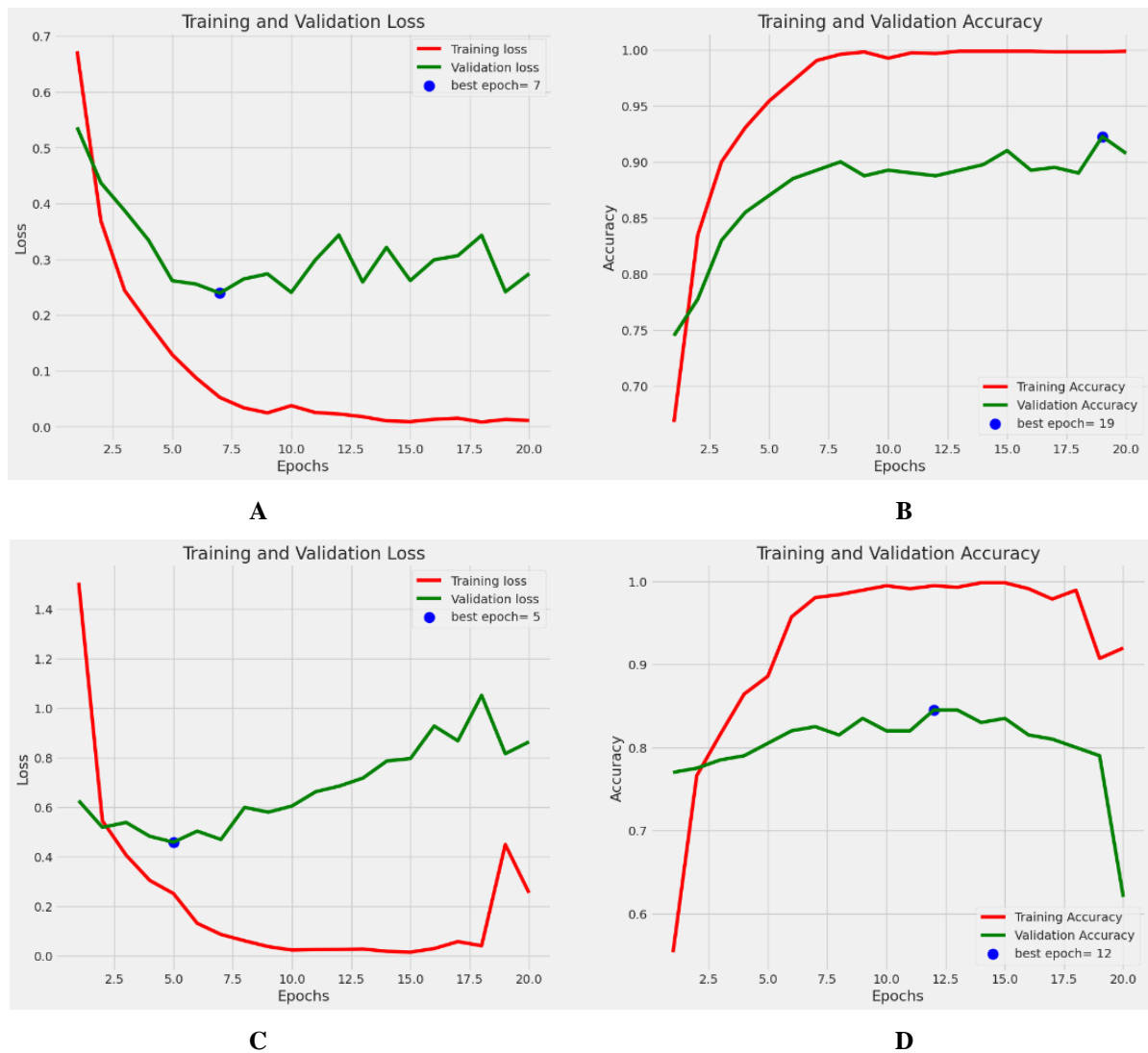
Model yang menggunakan augmentasi data lebih baik dibandingkan model yang tidak menggunakan augmentasi data. seperti yang terlihat di Tabel 4 bahwa model yang menggunakan augmentasi data banyak yang mendapatkan akurasi 90% ke atas sedangkan model yang tidak menggunakan augmentasi tidak ada yang mendapatkan akurasi diatas 90% dan hanya mentok di angka 87,92%.

Learning rate 0,01 merupakan *learning rate* yang paling banyak menghasilkan akurasi tertinggi sedangkan untuk *dropout* yang menghasilkan banyak akurasi tertinggi adalah *dropout* dengan nilai 0,001. Kebalikannya untuk *learning rate* yang memiliki hasil yang tidak maksimal adalah *learning rate* dengan nilai 0,1 tetapi ini sangat Nampak pada model yang menggunakan augmentasi data karena dapat dilihat pada pembagian data berapapun yang menggunakan *learning rate* 0,1 akurasi yang didapatkan tidak ada yang mencapai 90% sedangkan yang menggunakan selain *learning rate* 0,1 akurasinya bisa mencapai 90%. Tetapi hal ini tidak berlaku untuk yang tidak menggunakan augmentasi data karena pada

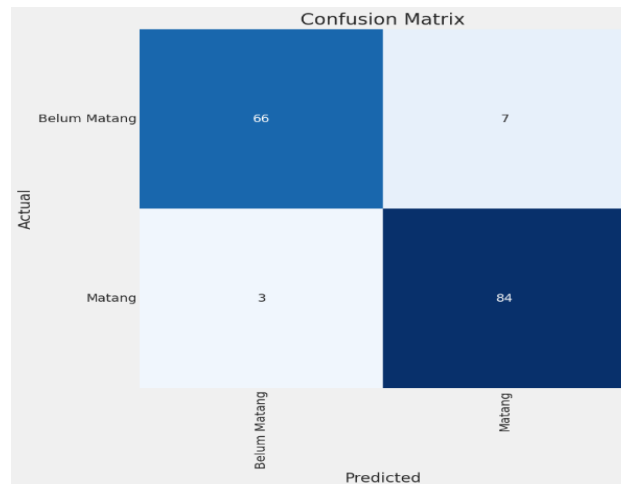
learning rate 0,1 ada yang bisa mendapatkan akurasi yang bagus walau model dengan akurasi terendah ada pada model yang menggunakan *learning rate* 0,1.

Model dengan pembagian data 90:10 menggunakan augmentasi data dengan *learning rate* 0,01 dan *dropout* 0,001 merupakan model terbaik karena mendapatkan akurasi yang paling tinggi dibandingkan model lainnya dengan akurasi sebesar 93,8%. Sedangkan model terburuk adalah model yang menggunakan pembagian data 70:30 yang tidak menggunakan augmentasi data dengan *learning rate* 0,1 dan *dropout* 0,001 dengan akurasi 64,6%.

Pada Gambar 5 menunjukkan grafik Training dan validasi loss augmentasi data pembagian data 90:10 *learning rate* 0,01 dan *dropout* 0,001 (Gambar 5(A)), Training dan validasi akurasi augmentasi data pembagian data 90:10 *learning rate* 0,01 (Gambar 5(B)), Training dan validasi loss tanpa augmentasi data pembagian data 70:30 *learning rate* 0,1 dan *dropout* 0,001 (Gambar 5(C)) dan Training dan validasi akurasi tanpa augmentasi data pembagian data 70:30 *learning rate* 0,1 dan *dropout* 0,001 (Gambar 5(D)). Pada (Gambar 5(A)) grafik antara training dan validasi loss memiliki jarak yang jauh walaupun training loss nya sudah sangat kecil tetapi validasi loss belum dengan konsisten dalam mengurangi lossnya dan terpaut sangat jauh walau lossnya sudah cukup kecil. Pada (Gambar 5(B)) memperlihatkan training dan akurasi dari model terbaik yang mana akurasi validasinya sudah tinggi tetapi masih mengalami naik turun dan tidak konsisten naik dan juga jarak antara akurasi training dan akurasi validasi masih jauh. Sedangkan (Gambar 5(C)) merupakan training dan validasi loss dari model yang paling buruk dan dapat terlihat pada grafiknya walau training lossnya sudah cukup rendah tetapi validasi lossnya sangat tinggi menjulang naik bukan semakin menurun. Pada (Gambar 5 (D)) merupakan akurasi training dan validasi yang mana akurasi validasinya semakin lama belajar semakin menurun dan sangat jauh melakukan menurun tajam kebawah sehingga model ini mendapatkan akurasi yang paling buruk.



Gambar 5. Hasil evaluasi : (A) Training dan validasi loss augmentasi data pembagian data 90:10 *learning rate* 0,01 dan *dropout* 0,001, (B) Training dan validasi akurasi augmentasi data pembagian data 90:10 *learning rate* 0,01 dan *dropout* 0,001, (C) Training dan validasi loss tanpa augmentasi data pembagian data 70:30 *learning rate* 0,1 dan *dropout* 0,001, (D) Training dan validasi akurasi tanpa augmentasi data pembagian data 70:30 *learning rate* 0,1 dan *dropout* 0,001



Gambar 6. Confusion matrix pembagian data 90:10 learning rate 0,1 dan dropout 0,001 dengan augmentasi

```

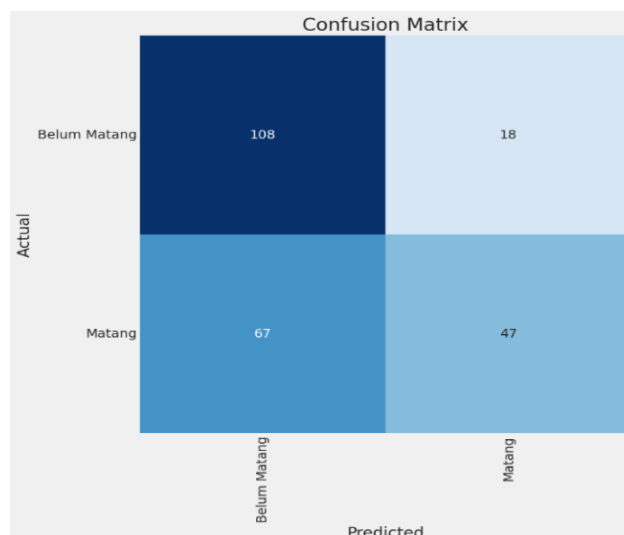
-----
              precision    recall  f1-score   support

Belum Matang    0.96      0.90      0.93      73
Matang          0.92      0.97      0.94      87

 accuracy              0.94      160
 macro avg           0.94      0.93      0.94      160
 weighted avg        0.94      0.94      0.94      160
    
```

Gambar 7. Precision, recall, f1 score dan akurasi pembagian data 90:10 learning rate 0,1 dan drop out 0,001 dengan augmentasi

Pada gambar 6 dan Gambar 7 menunjukkan confusion matrix yang mana Gambar 6 menunjukkan hasil prediksi dan Gambar 7 menunjukkan hasil perhitungan dari confusion matrix dari model dengan pembagian data 90:10 *learning rate* 0,01 dan dropout 0,001 dengan augmentasi data. Pada Gambar 6 menunjukkan bahwa dari 160 data test model dapat mengklasifikasi buah yang matang dengan benar sebanyak 84 gambar dan salah mengklasifikasi sebanyak 3 gambar. Sedangkan pada buah yang belum matang model dapat dengan mengklasifikasikan dengan benar sebanyak 66 gambar dan salah mengklasifikasi buah belum matang sebanyak 7 gambar. Dari hasil ini menunjukkan bahwa model lebih banyak melakukan klasifikasi dengan benar dibandingkan salah mengklasifikasikan tingkat kematangan sawit. Pada Gambar 7 adalah hasil dari perhitungan dari *accuracy*, *precision*, *recall*, dan *f1 score* dari model ini yang mana perhitungan ini mengambil data dari confusion matrix dan dihitung sesuai dengan rumusnya. Dari model ini didapatkan *accuracy* 94%, *precision* matang 92% dan belum matang 96%, *recall* matang 97% dan belum matang 90%, *f1 score* matang 94% dan belum matang 93%.



Gambar 8. Confusion matrix pembagian data 70:30 learning rate 0,1 dan dropout 0,001 tanpa augmentasi

	precision	recall	f1-score	support
Belum Matang	0.62	0.86	0.72	126
Matang	0.72	0.41	0.53	114
accuracy			0.65	240
macro avg	0.67	0.63	0.62	240
weighted avg	0.67	0.65	0.63	240

Gambar 9. Precision, recall, f1 score dan akurasi pembagian data 70:30 learning rate 0,1 dan dropout 0,001 tanpa augmentasi

Pada gambar 8 dan Gambar 9 menunjukkan confusion matrix yang mana Gambar 8 menunjukkan hasil prediksi dan Gambar 9 menunjukkan hasil perhitungan dari confusion matrix dari model dengan pembagian data 70:30 *learning rate* 0,1 dan dropout 0,001 tanpa menggunakan augmentasi data. Pada Gambar 8 menunjukkan bahwa dari 240 data test model dapat mengklasifikasi buah yang matang dengan benar hanya 47 gambar dan salah mengklasifikasi sebanyak 67 gambar. Sedangkan pada buah yang belum matang model dapat dengan mengklasifikasikan dengan benar sebanyak 108 gambar dan salah mengklasifikasi buah belum matang sebanyak 18 gambar. Dari hasil ini menunjukkan bahwa model bisa memiliki grafik yang jelek dalam akurasi validasi karena model ini sangat banyak melakukan kesalahan pada saat mengklasifikasi buah yang matang sampai 67 gambar dan hanya dapat melakukan klasifikasi dengan benar sebanyak 47 gambar. Sedangkan pada bagian mengklasifikasi buah belum matang model dapat cukup baik melakukannya dengan dapat mengklasifikasinya dengan benar sebanyak 108 gambar. Walau begitu model ini juga cukup banyak salah mengklasifikasi gambar tidak matang sebanyak 18 gambar. Pada Gambar 9 adalah hasil dari perhitungan dari *accuracy*, *precision*, *recall*, dan *f1 score* dari model ini yang mana perhitungan ini mengambil data dari confusion matrix dan dihitung sesuai dengan rumusnya. Dari model ini didapatkan *accuracy* 65%, *precision* matang 72% dan belum matang 62%, *recall* matang 86% dan belum matang 41%, *f1 score* matang 72% dan belum matang 53%. Hasil dari perhitungan model ini menjadi buruk karena banyaknya kesalahan dalam melakukan klasifikasi seperti yang terlihat di Gambar 8.

4. KESIMPULAN

Algoritma yang diusulkan yaitu VGG 16 dapat melakukan klasifikasi kematangan sawit dengan baik dan cepat. Dengan 57 skema yang telah dilakukan, didapatkan hasil bahwa augmentasi data dapat meningkatkan hasil akurasi menjadi lebih baik dibandingkan tidak menggunakan. Tidak hanya augmentasi saja tetapi pemilihan hyperparameter juga berpengaruh dan pembagian data juga begitu sama pengaruhnya. Rata rata model yang menggunakan augmentasi lebih tinggi akurasinya dibandingkan dengan yang tidak menggunakannya. Tetapi ada juga model yang sudah menggunakan augmentasi tetapi akurasinya lebih rendah dibandingkan yang tidak menggunakan augmentasi dan disinilah yang menjadikan bahwa hyperparameter itu juga berpengaruh pada naik turunnya akurasi. Dengan hyperparameter yang tepat membuat akurasi naik dan dengan menggunakan kombinasi hyperparameter yang tidak tepat membuat akurasi mengalami penurunan. Model terbaik adalah VGG 16 dengan pembagian data 90:10 menggunakan augmentasi data dengan *learning rate* 0,01 dan drop out 0,001. Dari confusion matrix yang didapat menunjukkan bahwa dari 160 data test, model dapat mengklasifikasi buah yang matang dengan benar sebanyak 84 gambar dan hanya salah mengklasifikasi sebanyak 3 gambar. Sedangkan pada buah yang belum matang model dapat dengan mengklasifikasikan dengan benar sebanyak 66 gambar dan salah mengklasifikasi buah belum matang sebanyak 7 gambar. Dari hasil ini menunjukkan bahwa model lebih banyak melakukan klasifikasi dengan benar dibandingkan salah mengklasifikasikan tingkat kematangan sawit. Dan setelah dilakukan perhitungan didapatkan *accuracy* 94%, *precision* matang 92% dan belum matang 96%, *recall* matang 97% dan belum matang 90%, *f1 score* matang 94% dan belum matang 93%. Untuk penelitian kedepannya bisa dicoba untuk melakukan perubahan variasi pada learning rate dan dropout yang lainnya dan juga bisa menggunakan lebih banyak metode augmentasi dan jumlah augmentasinya. Kemudian pada penelitian ini hanya menggunakan 20 *epoch* saja, untuk penelitian selanjutnya bisa menggunakan *epoch* yang lebih tinggi lagi.

REFERENCES

- [1] M. Lambok, F. Sitorus, E. N. Akoeb, R. Sembiring, and M. A. Siregar, "Peningkatan Produksi Crude Palm Oil Melalui Kriteria Matang Panen Tandan Buah Segar untuk Optimalisasi Pendapatan Perusahaan Improving Crude Palm Oil Production Through Fresh Fruit Harvest Criteria for Optimization of Company Income," *J. Ilm. Magister Agribisnis*, vol. 2, no. 1, pp. 26–32, 2020, [Online].
- [2] J. Supriadi, G. Tabrani, and Isnaini, "As the Reap Indicator Observed From Morphological," *Jom Faperta*, vol. 6, pp. 1–9, 2019.
- [3] D. Bayu, Priyambada, and G. Supriyanto, "Analisis Rendemen Minyak Kelapa Sawit (CPO) berdasarkan Tingkat Kematangan Buah di PT . Bumitama Gunajaya Agro," vol. 1, no. September, pp. 2051–2060, 2023.
- [4] I. U. P. Rangkuti, "Rendemen dan Komponen Minor Minyak Sawit Mentah Berdasarkan Tingkat Kematangan Buah pada Elevasi Tinggi," *Agrotekma J. Agroteknologi dan Ilmu Pertan.*, vol. 3, no. 1, p. 9, 2018.
- [5] R. A. Sirait and G. Supriyanto, "Pengaruh Kematangan Buah Terhadap FFA dan Besarnya Kandungan Minyak di Dalamnya di Pabrik Kelapa Sawit," *Agroforetech*, vol. 1, no. Gapksi 2022, p. 677, 2023.

- [6] M. Subramanian, V. Easwaramoorthy Sathiskumar, G. Deepalakshmi, J. Cho, and G. Manikandan, "A survey on hate speech detection and sentiment analysis using machine learning and deep learning models," *Alexandria Eng. J.*, vol. 80, no. August, pp. 110–121, 2023.
- [7] F. A. Junior and Suharjo, "Video based oil palm ripeness detection model using deep learning," *Heliyon*, vol. 9, no. 1, p. e13036, 2023.
- [8] S. Yadav, Alpana, and S. Chand, "Automated Food image Classification using Deep Learning approach," *2021 7th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2021*, pp. 542–545, 2021.
- [9] A. S. Paymode and V. B. Malode, "Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG," *Artif. Intell. Agric.*, vol. 6, pp. 23–33, 2022.
- [10] K. Alomar, H. I. Aysel, and X. Cai, "Data Augmentation in Classification and Segmentation: A Survey and New Strategies," *J. Imaging*, vol. 9, no. 2, 2023.
- [11] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *Pattern Recognit.*, vol. 137, p. 109347, 2023.
- [12] M. Z. Islam, M. S. Hossain, R. Ul Islam, and K. Andersson, "Static hand gesture recognition using convolutional neural network with data augmentation," *2019 Jt. 8th Int. Conf. Informatics, Electron. Vision, ICIEV 2019 3rd Int. Conf. Imaging, Vis. Pattern Recognition, icIVPR 2019 with Int. Conf. Act. Behav. Comput. ABC 2019*, pp. 324–329, 2019.
- [13] M. Alkhaleefah, P. Kumar Chittam, V. P. Achhannagari, S. C. Ma, and Y. L. Chang, "The Influence of Image Augmentation on Breast Lesion Classification Using Transfer Learning," *2020 Int. Conf. Artif. Intell. Signal Process. AISP 2020*, pp. 0–4, 2020.
- [14] T. B. Sasongko, H. Haryoko, and A. Amrullah, "Analisis Efek Augmentasi Dataset dan Fine Tune pada Algoritma Pre-Trained Convolutional Neural Network (CNN)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 4, pp. 763–768, 2023.
- [15] P. Kamsing, P. Torteeka, and S. Yooyen, "Deep Convolutional Neural Networks for plane identification on Satellite imagery by exploiting transfer learning with a different optimizer," *Int. Geosci. Remote Sens. Symp.*, pp. 9788–9791, 2019.
- [16] I. Valova, C. Harris, T. Mai, and N. Gueorguieva, "Optimization of convolutional neural networks for imbalanced set classification," *Procedia Comput. Sci.*, vol. 176, pp. 660–669, 2020.
- [17] K. Cui, Z. Zhan, and C. Pan, "Applying Radam Method to improve treatment of Convolutional Neural Network on Banknote Identification," *Proc. - 2020 Int. Conf. Comput. Eng. Appl. ICCEA 2020*, pp. 468–476, 2020.
- [18] G. W. Qiu, X. Yu, B. Sun, Y. Wang, and L. Zhang, "Metastatic Cancer Image Classification Based on Deep Learning Method," *2021 IEEE Int. Conf. Consum. Electron. Comput. Eng. ICCECE 2021*, no. Iccece, pp. 658–661, 2021.
- [19] D. O. Melinte and L. Vladareanu, "Facial Expressions Recognition for Human – Robot Interaction Using Deep Convolutional Neural," 2020, [Online].
- [20] T. Chauhan, H. Palivela, and S. Tiwari, "International Journal of Information Management Data Insights Optimization and fine-tuning of DenseNet model for classification of COVID-19 cases in medical imaging," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100020, 2021.
- [21] F. G. Febrinanto, C. Dewi, and A. T. Wiratno, "Implementasi Algoritme K-Means Sebagai Metode Segmentasi Citra Dalam Identifikasi Penyakit Daun Jeruk," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 11, pp. 5375–5383, 2018.
- [22] K. L. Pham, K. M. Dang, L. P. Tang, and T. N. Nguyen, "GAN Generated Portraits Detection Using Modified VGG-16 and EfficientNet," *Proc. - 2020 7th NAFOSTED Conf. Inf. Comput. Sci. NICS 2020*, pp. 344–349, 2020.
- [23] N. HIDAYATI, "Modifikasi Arsitektur Densenet121 Dengan Transfer Learning Untuk Deteksi Penyakit Tanaman Jagung Berdasarkan Citra Daun," *Angew. Chemie Int. Ed.* 6(11), 951–952., pp. 2013–2015, 2021.
- [24] M. A. Muslim *et al.*, *Data Mining Algoritma C4.5 Disertai contoh kasus dan penerapannya dengan program computer.* 2019.