

KLASIFIKASI UNTUK MENDETEKSI PENYAKIT IKAN MENGUNAKAN MODEL VGG-16 DAN MOBILENETV2

(Fish Disease Classification Using VGG-16 and MobileNetV2 Models)

Maftuh Ahnan Al-Kautsar^[1], Fitri Bimantoro^[2], Gede Pasek Suta Wijaya^[3]

Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: f1d022135@student.unram.ac.id, [bimo, gpsutawijaya]@unram.ac.id

Abstract

Fish disease is a major challenge in the aquaculture sector, causing global economic losses of up to US\$6 billion annually. Conventional disease identification relies on visual observation and laboratory testing, which is time-consuming, costly, and requires specialized expertise. This study investigates freshwater fish disease classification using two Convolutional Neural Network (CNN) architectures based on transfer learning, namely VGG-16 and MobileNetV2, combined with sequential hyperparameter tuning. The dataset consists of 2,800 images across eight classes, comprising 2,000 training images and 800 images used for validation and final evaluation. Sequential tuning was performed across five hyperparameters: optimizer, learning rate, dropout, batch size, and number of epochs. The best configuration of VGG-16 (Adam optimizer, learning rate $1e-4$, dropout 0.2, batch size 16, and 50 epochs) achieved an accuracy of 99.62%, while MobileNetV2 (Adam, learning rate $1e-3$, dropout 0.2, batch size 16, and 50 epochs) achieved an accuracy of 99.37% on the evaluation data. VGG-16 produced a slightly higher accuracy, with a difference of 0.25 percentage points, whereas MobileNetV2 had substantially fewer base-model parameters. These results indicate that both architectures achieved high classification performance under the experimental setting used in this study, while their differences in model complexity may be considered when selecting a model for further deployment. However, the use of the same data for validation and final evaluation represents a limitation that may result in an optimistic estimation of model performance.

Keywords: Transfer Learning, VGG-16, MobileNetV2, Hyperparameter Tuning, Fish Disease Classification

1. PENDAHULUAN

Indonesia merupakan negara kepulauan dengan luas wilayah perairan sekitar 6,4 juta km², sehingga memiliki potensi sumber daya kelautan dan perikanan yang sangat besar [1]. Potensi tersebut menjadikan sektor perikanan, khususnya akuakultur, sebagai salah satu komponen penting dalam mendukung ketahanan pangan nasional maupun global. Seiring dengan meningkatnya kebutuhan pangan akibat pertumbuhan populasi dunia, produksi perikanan budidaya terus mengalami peningkatan [2]. Berdasarkan data Badan Pusat Statistik, produksi perikanan budidaya Indonesia mencapai 15,54 juta ton pada tahun 2019 [1].

Di balik potensi tersebut, sektor akuakultur masih menghadapi berbagai tantangan, salah satunya adalah tingginya prevalensi penyakit pada ikan budidaya. Wabah penyakit ikan diperkirakan menyebabkan kerugian ekonomi global hingga mencapai US\$6 miliar setiap tahun [3]. Berbagai penyakit, seperti Aeromoniasis, Bacterial Gill Disease, Saprolegniasis, penyakit parasit, dan White Tail Disease, merupakan

penyakit yang umum menyerang ikan air tawar maupun ikan laut, sehingga deteksi dini menjadi faktor penting untuk mengendalikan penyebaran penyakit serta mengurangi tingkat mortalitas [4]. Namun, proses identifikasi penyakit yang dilakukan secara konvensional melalui observasi visual oleh pakar maupun pengujian laboratorium masih memerlukan waktu, biaya, dan tenaga ahli, sehingga kurang efisien untuk diterapkan secara luas di lapangan.

Perkembangan Artificial Intelligence (AI), khususnya pada bidang *deep learning* dan *computer vision*, memberikan peluang yang signifikan dalam pengembangan sistem identifikasi penyakit ikan berbasis citra secara otomatis [5]. Salah satu pendekatan yang banyak digunakan adalah *Convolutional Neural Network* (CNN) dengan teknik *transfer learning*, yang mampu memanfaatkan pengetahuan dari model pralatih (*pre-trained model*) sehingga menghasilkan performa klasifikasi yang tinggi meskipun menggunakan dataset berukuran relatif terbatas. Berbagai penelitian melaporkan bahwa arsitektur seperti VGG-16, MobileNetV2, dan

EfficientNet mampu mencapai tingkat akurasi klasifikasi di atas 98% pada kasus identifikasi penyakit ikan [6].

Meskipun demikian, hasil tinjauan terhadap sepuluh penelitian terdahulu menunjukkan bahwa sebagian besar penelitian masih berfokus pada evaluasi satu arsitektur CNN tanpa melakukan analisis komparatif terhadap beberapa arsitektur dalam kondisi eksperimen yang sama. Selain itu, optimasi *hyperparameter* pada penelitian terdahulu belum banyak dilakukan secara sistematis dalam konteks klasifikasi penyakit ikan, sehingga konfigurasi yang digunakan masih didominasi oleh parameter bawaan (*default*) atau pengujian parameter secara terbatas. Beberapa strategi optimasi seperti *Grid Search*, *Random Search*, dan *Bayesian Optimization* menawarkan pendekatan yang berbeda dalam pencarian konfigurasi *hyperparameter*, tetapi masing-masing memiliki karakteristik dalam kebutuhan jumlah eksperimen dan strategi pencarian. Dalam konteks penelitian ini, *sequential hyperparameter tuning* digunakan dengan mengoptimalkan satu *hyperparameter* pada setiap tahap dan mempertahankan konfigurasi terbaik sebelum melanjutkan ke tahap berikutnya. Pendekatan tersebut memberikan jumlah eksperimen yang lebih sedikit dibandingkan pencarian kombinasi secara menyeluruh, tetapi tidak menjamin diperolehnya kombinasi *hyperparameter* terbaik secara global karena hasil optimasi dipengaruhi oleh urutan parameter yang diuji. Dengan demikian, *research gap* penelitian ini terletak pada penerapan dan analisis *sequential hyperparameter tuning* secara terstruktur pada dua arsitektur CNN berbasis *transfer learning* dalam kondisi eksperimen yang sama untuk klasifikasi penyakit ikan air tawar.

Berdasarkan kesenjangan penelitian tersebut, penelitian ini mengkaji klasifikasi penyakit ikan air tawar menggunakan dua arsitektur *Convolutional Neural Network* (CNN) berbasis *transfer learning*, yaitu VGG-16 dan MobileNetV2. Kedua arsitektur dipilih karena memiliki karakteristik yang berbeda dari sisi kompleksitas model dan efisiensi komputasi. VGG-16 memiliki struktur model yang lebih besar, sedangkan MobileNetV2 dirancang dengan arsitektur yang lebih ringan dan memiliki jumlah parameter yang lebih sedikit. Perbedaan karakteristik tersebut memungkinkan dilakukan analisis terhadap trade-off antara performa klasifikasi dan kompleksitas model. Optimasi masing-masing model dilakukan menggunakan pendekatan *sequential hyperparameter tuning* terhadap lima parameter utama, yaitu

optimizer, *learning rate*, *dropout*, *batch size*, dan jumlah *epoch*. Proses optimasi dilakukan secara bertahap dengan mempertahankan konfigurasi terbaik pada setiap tahap sebelum melanjutkan ke tahap optimasi berikutnya.

Penelitian ini memberikan tiga kontribusi utama, yaitu: (1) penerapan metode *sequential hyperparameter tuning* secara sistematis pada dua arsitektur CNN berbasis *transfer learning* dengan jumlah eksperimen yang lebih efisien; (2) identifikasi konfigurasi *hyperparameter* terbaik yang diperoleh melalui tahapan *sequential tuning* untuk masing-masing arsitektur; serta (3) analisis komparatif mengenai *trade-off* antara performa klasifikasi dan kompleksitas model VGG-16 dan MobileNetV2 pada kondisi eksperimen yang sama.

2. TINJAUAN PUSTAKA

2.1 Klasifikasi Penyakit Ikan Berbasis Deep Learning

Berbagai penelitian telah menunjukkan bahwa *Convolutional Neural Network* (CNN) memiliki kinerja yang sangat baik dalam tugas klasifikasi penyakit ikan berbasis citra. Biswas et al. [6] mengevaluasi beberapa arsitektur CNN, yaitu VGG-16, MobileNetV2, InceptionV3, ResNet, EfficientNet, dan ConvNeXt, yang dikombinasikan dengan metode *ensemble learning* berbasis *Support Vector Machine* (SVM) pada dataset *Freshwater Fish Disease Aquaculture in South Asia* yang terdiri atas sekitar 2.450 citra dengan tujuh kelas penyakit. Pendekatan tersebut berhasil mencapai akurasi sebesar 99,59% menggunakan *ensemble learning* berbasis SVM yang mengombinasikan fitur dari beberapa arsitektur. Hasan et al. [7] melaporkan akurasi sebesar 94,44% menggunakan CNN pada dataset yang lebih kecil, yaitu 90 citra yang mencakup dua kelas penyakit. Sementara itu, Tamut et al. [8] memperoleh akurasi sebesar 99,71% pada dataset yang terdiri atas 2.444 citra dengan tujuh kelas.

Selain menggunakan arsitektur CNN konvensional, beberapa penelitian juga mengembangkan pendekatan berbasis model ringan maupun teknik prapemrosesan citra. Penelitian mengenai deteksi *Epizootic Ulcerative Syndrome* menggunakan MobileNetV2 yang dikombinasikan dengan segmentasi warna HSV memperoleh akurasi sekitar 84% [9]. Di sisi lain, Afridiansyah et al. [10] menerapkan arsitektur EfficientNetB1 pada dua dataset yang berbeda dan memperoleh akurasi masing-masing sebesar 98,14% dan 99,18%. Hasil-hasil tersebut menunjukkan bahwa berbagai arsitektur CNN telah digunakan pada tugas

klasifikasi penyakit ikan dengan karakteristik model, dataset, dan pendekatan yang berbeda.

Berdasarkan penelitian-penelitian tersebut, penggunaan CNN dan *transfer learning* telah menunjukkan performa yang tinggi dalam klasifikasi penyakit ikan. Namun, pendekatan yang digunakan bervariasi dari sisi arsitektur, jumlah kelas, karakteristik dataset, dan strategi optimasi, sehingga hasil akurasi antarpelitian tidak dapat dibandingkan secara langsung. Selain itu, kajian yang secara khusus mengevaluasi VGG-16 dan MobileNetV2 dalam kondisi eksperimen yang sama disertai optimasi beberapa *hyperparameter* secara bertahap masih terbatas. Oleh karena itu, penelitian ini berfokus pada evaluasi kedua arsitektur tersebut menggunakan kondisi dataset dan tahapan optimasi yang sama sehingga perbedaan karakteristik performa dan kompleksitas model dapat dianalisis secara lebih terkontrol.

2.2 *Hyperparameter Tuning* pada Model CNN

Penelitian mengenai optimasi *hyperparameter* pada *Convolutional Neural Network* (CNN) telah banyak dilakukan pada berbagai domain klasifikasi citra dan menunjukkan bahwa konfigurasi *hyperparameter* dapat memengaruhi performa model. Zahro et al. [11] melakukan optimasi *learning rate*, *batch size*, *optimizer*, dan jumlah *epoch* pada arsitektur MobileNetV3-Small dan memperoleh akurasi hingga 92% pada tugas klasifikasi citra bunga. Sementara itu, Julianto et al. [12] melakukan optimasi *hyperparameter* MobileNetV2 untuk klasifikasi penyakit tanaman padi dan memperoleh nilai *accuracy*, *precision*, *recall*, serta *F1-score* di atas 97%.

Penelitian lain menunjukkan bahwa strategi pencarian *hyperparameter* dapat dilakukan melalui berbagai pendekatan. Kette [13] membandingkan *Bayesian Optimization*, *Grid Search*, dan *Random Search* pada klasifikasi data astronomi. Secara umum, *Grid Search* mengevaluasi kombinasi nilai pada ruang pencarian yang telah ditentukan, sedangkan *Random Search* melakukan pencarian berdasarkan kombinasi yang dipilih secara acak dan *Bayesian Optimization* menggunakan informasi dari evaluasi sebelumnya untuk menentukan konfigurasi berikutnya. Fitrianto dan Sartono [14] menganalisis berbagai konfigurasi *hyperparameter* pada CNN untuk klasifikasi citra daging sapi dan babi dan memperoleh akurasi sebesar 98,7% tanpa menggunakan pendekatan *transfer learning*. Selain itu, Rusman et al. [15] menerapkan *Grid Search* pada algoritma *Support Vector Machine* (SVM) untuk klasifikasi tingkat kematangan buah kopi dan memperoleh akurasi hingga 98%. Gea et al. [16] juga mengkaji konfigurasi *hyperparameter* pada CNN

untuk klasifikasi citra MRI tumor otak dan menunjukkan adanya perbedaan performa pada penggunaan *learning rate*, *batch size*, dan jumlah *epoch* yang berbeda.

Berbeda dengan metode pencarian yang mengevaluasi berbagai kombinasi *hyperparameter*, penelitian ini menggunakan pendekatan *sequential hyperparameter tuning*, yaitu optimasi dilakukan terhadap satu *hyperparameter* pada setiap tahap, kemudian nilai terbaik yang diperoleh dipertahankan sebagai konfigurasi tetap pada tahap berikutnya. Pendekatan ini mengurangi jumlah eksperimen karena tidak mengevaluasi seluruh kombinasi *hyperparameter* secara menyeluruh. Namun, hasil yang diperoleh bergantung pada urutan *hyperparameter* yang dioptimalkan dan tidak menjamin diperolehnya kombinasi terbaik secara global. Oleh karena itu, pendekatan *sequential tuning* dalam penelitian ini digunakan sebagai strategi optimasi yang menekankan efisiensi jumlah eksperimen, bukan sebagai pengganti yang secara umum lebih baik daripada *Grid Search*, *Random Search*, maupun *Bayesian Optimization*.

Secara keseluruhan, penelitian-penelitian terdahulu menunjukkan bahwa konfigurasi *hyperparameter* merupakan salah satu faktor penting yang memengaruhi performa model CNN. Namun, pendekatan optimasi yang digunakan, arsitektur model, serta domain penerapannya berbeda-beda. Pada domain klasifikasi penyakit ikan, penelitian yang mengevaluasi dua arsitektur CNN berbasis *transfer learning* dalam kondisi eksperimen yang sama dengan optimasi beberapa *hyperparameter* secara bertahap masih terbatas. Oleh karena itu, penelitian ini menerapkan *sequential hyperparameter tuning* pada VGG-16 dan MobileNetV2 untuk memperoleh konfigurasi *hyperparameter* terbaik dari kandidat nilai yang diuji serta menganalisis karakteristik performa kedua arsitektur pada kondisi eksperimen yang sama.

3. METODE PENELITIAN

3.1 Dataset

Dataset yang digunakan dalam penelitian ini merupakan dataset penyakit ikan air tawar yang diperoleh dari dua sumber, yaitu Kaggle dan Roboflow. Sebanyak tujuh kelas penyakit ikan diperoleh dari dataset *Freshwater Fish Disease Aquaculture in South Asia* pada Kaggle, sedangkan kelas White Spot Disease diperoleh dari Roboflow. Penggunaan dua sumber data dilakukan untuk melengkapi jumlah kelas penyakit yang digunakan dalam penelitian sehingga diperoleh total delapan kelas.

Pada dataset Kaggle, setiap kelas telah tersedia dalam pembagian 250 citra data pelatihan (*training*) dan 100 citra data pengujian (*test*). Selanjutnya, data kelas *White Spot Disease* yang diperoleh dari Roboflow disesuaikan dengan distribusi yang sama, yaitu 250 citra untuk data pelatihan dan 100 citra untuk data pengujian. Dengan demikian, seluruh kelas memiliki jumlah data yang seimbang. Setelah seluruh kelas digabungkan, diperoleh sebanyak 2.800 citra, yang terdiri atas 2.000 citra pelatihan dan 800 citra pengujian, dengan masing-masing kelas memiliki 250 citra pelatihan dan 100 citra pengujian.

Delapan kelas yang digunakan terdiri atas Bacterial Diseases – Aeromoniasis, Bacterial Gill Disease, Bacterial Red Disease, Fungal Diseases – Saprolegniasis, Parasitic Diseases, Viral Diseases – White Tail Disease, Healthy Fish, dan White Spot Disease. Distribusi dataset ditunjukkan pada Tabel I.

TABEL I. DISTRIBUSI DATASET PER KELAS

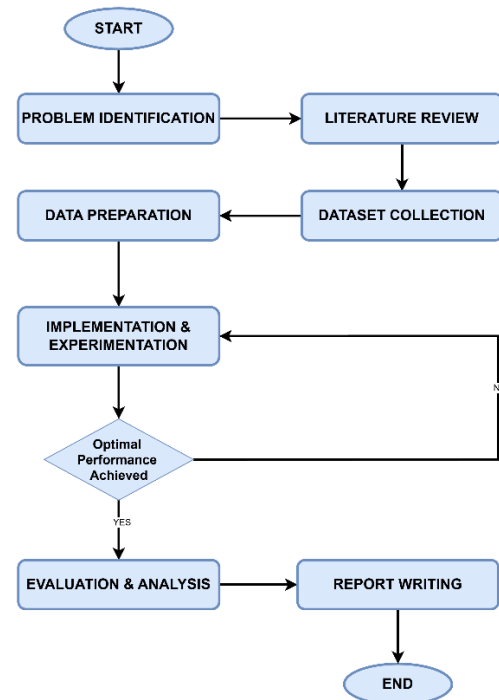
No.	Kelas	Citra Train	Citra Test	Total
1	Bacterial Diseases – Aeromoniasis	250	100	350
2	Bacterial Gill Disease	250	100	350
3	Bacterial Red Disease	250	100	350
4	Fungal Diseases – Saprolegniasis	250	100	350
5	Parasitic Diseases	250	100	350
6	Viral Diseases – White Tail Disease	250	100	350
7	Healthy Fish	250	100	350
8	White Spot	250	100	350

Sebelum digunakan dalam proses pelatihan, seluruh citra melalui tahap *preprocessing* berupa perubahan ukuran (*resize*) menjadi 224×224 piksel dan normalisasi nilai piksel dengan membagi nilai piksel dengan 255, sehingga nilai piksel berada pada rentang 0 hingga 1. Pada data pelatihan juga diterapkan *data augmentation* berupa rotasi hingga 15° , *horizontal flip*, *width shift* sebesar 0,1, *height shift* sebesar 0,1, dan *zoom* sebesar 0,1. Augmentasi digunakan untuk memberikan variasi citra pada proses pelatihan sehingga model dapat mempelajari karakteristik objek ikan pada kondisi posisi dan skala yang berbeda. Data yang digunakan untuk pengujian tidak diberikan augmentasi sehingga tetap merepresentasikan data pada kondisi aslinya.

Karena penelitian ini menggunakan data dari dua sumber yang berbeda, yaitu Kaggle dan Roboflow,

terdapat kemungkinan adanya perbedaan karakteristik citra, seperti latar belakang, pencahayaan, dan kualitas gambar. Perbedaan karakteristik sumber data tersebut menjadi salah satu aspek yang perlu diperhatikan dalam interpretasi hasil klasifikasi.

3.2 Alur Penelitian



Gambar 1. Alur Penelitian

Penelitian ini dilaksanakan melalui tujuh tahapan yang disusun secara sistematis, yaitu *Problem Identification*, *Literature Review*, *Dataset Collection*, *Data Preparation*, *Implementation and Experimentation*, *Evaluation and Analysis*, serta *Report Writing*. Pada tahap *Implementation and Experimentation*, model dikembangkan menggunakan pendekatan *transfer learning* dan dioptimalkan melalui metode *sequential hyperparameter tuning*. Proses optimasi dilakukan secara bertahap terhadap setiap *hyperparameter*, dengan mempertahankan konfigurasi terbaik yang diperoleh pada setiap tahap sebagai dasar untuk proses optimasi pada tahap berikutnya. Setelah seluruh proses optimasi selesai, model dilatih kembali menggunakan kombinasi *hyperparameter* terbaik melalui tahap *final training* untuk menghasilkan model dengan performa yang optimal. Alur penelitian secara keseluruhan ditunjukkan pada Gambar 1.

3.3 Arsitektur Model

Kedua arsitektur *Convolutional Neural Network* (CNN) yang digunakan dalam penelitian ini, yaitu VGG-

16 dan MobileNetV2, memiliki karakteristik yang berbeda dari sisi struktur jaringan dan jumlah parameter. VGG-16 menggunakan susunan lapisan konvolusi berukuran 3×3 secara berurutan, sedangkan MobileNetV2 menggunakan mekanisme *depthwise separable convolution* dan *inverted residual block* untuk membangun arsitektur yang lebih ringan. Pada penelitian ini, kedua arsitektur diimplementasikan menggunakan pendekatan *transfer learning* dengan bobot awal (*pre-trained weights*) dari ImageNet.

Pada kedua model, seluruh lapisan pada *base model* dibekukan (*frozen*) sehingga bobot hasil *pre-training* tidak diperbarui selama proses pelatihan. Lapisan klasifikasi (*classifier head*) kemudian ditambahkan pada *base model* dan dilatih untuk menyesuaikan model dengan tugas klasifikasi delapan kelas penyakit dan kondisi ikan. Pada VGG-16, fitur hasil ekstraksi dari *base model* diproses menggunakan lapisan *Flatten*, kemudian diteruskan ke lapisan *Dense* dengan 256 neuron dan fungsi aktivasi ReLU, *Dropout*, dan lapisan keluaran dengan 8 neuron menggunakan fungsi aktivasi Softmax. Sementara itu, MobileNetV2 menggunakan *GlobalAveragePooling2D* untuk mereduksi fitur hasil ekstraksi, kemudian diteruskan ke lapisan *Dense* dengan 128 neuron dan fungsi aktivasi ReLU, *Dropout*, serta lapisan keluaran dengan 8 neuron menggunakan fungsi aktivasi Softmax.

Nilai *dropout* pada *classifier head* termasuk dalam *hyperparameter* yang dioptimasi melalui *sequential hyperparameter tuning*. Kandidat nilai *dropout* yang diuji adalah 0,2, 0,3, dan 0,5. Dengan demikian, struktur dasar kedua model dipertahankan, sedangkan nilai *dropout* ditentukan melalui tahapan optimasi yang dijelaskan pada Subbab 3.4.

TABEL II. RINGKASAN ARSITEKTUR MODEL

Parameter	VGG-16	MobileNetV2
Pretrained Weights	<i>ImageNet</i>	<i>ImageNet</i>
Base Model	<i>Frozen</i>	<i>Frozen</i>
Feature Processing	<i>Flatten</i>	<i>GlobalAveragePooling2D</i>
Hidden Layer	Dense 256, ReLU	Dense 128, ReLU
Regularization	<i>Dropout</i>	<i>Dropout</i>
Output Layer	Dense 8, Softmax	Dense 8, Softmax
Jumlah Parameter Base	21.139.528 (±80,64 MB)	2.587.976 (±9,87 MB)

Mekanisme Model	Konvolusi 3×3 berurutan	<i>Depthwise separable convolution, inverted residual</i>
-----------------	-------------------------	---

3.4 Sequential Hyperparameter Tuning

Optimasi model dilakukan menggunakan pendekatan *sequential hyperparameter tuning*, yaitu dengan menguji satu *hyperparameter* pada setiap tahap secara berurutan, meliputi *optimizer*, *learning rate*, *dropout*, *batch size*, dan jumlah *epoch*. Pada setiap tahap, konfigurasi *hyperparameter* yang telah ditentukan pada tahap sebelumnya dipertahankan sebagai parameter tetap (*fixed parameter*), sedangkan hanya *hyperparameter* pada tahap yang sedang dioptimasi yang diubah. Konfigurasi dengan performa *validation* terbaik pada setiap tahap kemudian digunakan sebagai dasar untuk tahap optimasi berikutnya.

Urutan *hyperparameter* ditetapkan mulai dari *optimizer* dan *learning rate* karena kedua parameter tersebut secara langsung memengaruhi proses pembaruan bobot dan konvergensi model. Selanjutnya, *dropout* digunakan untuk mengatur regularisasi, kemudian *batch size* untuk menentukan jumlah sampel yang diproses dalam setiap iterasi pembaruan bobot, dan terakhir jumlah *epoch* untuk menentukan durasi proses pelatihan berdasarkan konfigurasi yang telah diperoleh dari tahap sebelumnya. Urutan tersebut digunakan secara konsisten pada kedua arsitektur agar perbandingan dilakukan dalam prosedur eksperimen yang sama.

Pendekatan *sequential hyperparameter tuning* dipilih untuk mengurangi jumlah kombinasi eksperimen dibandingkan *grid search* secara menyeluruh. Dengan lima *hyperparameter* yang masing-masing memiliki tiga kandidat nilai, *grid search* memerlukan $3^5 = 243$ kombinasi eksperimen untuk setiap model atau 486 eksperimen untuk dua model. Sebaliknya, *sequential hyperparameter tuning* hanya menguji tiga kandidat pada masing-masing dari lima tahap, sehingga memerlukan 15 eksperimen untuk setiap model atau 30 eksperimen untuk dua model. Dengan demikian, jumlah eksperimen berkurang sekitar 93,83%, atau dibulatkan menjadi 94%, dibandingkan *full grid search*.

Meskipun jumlah eksperimen lebih sedikit, pendekatan *sequential hyperparameter tuning* memiliki keterbatasan karena hasil optimasi bergantung pada urutan *hyperparameter* yang ditentukan. Konfigurasi yang diperoleh merupakan konfigurasi terbaik di antara kandidat yang diuji pada

urutan eksperimen tersebut dan tidak dimaksudkan untuk menyatakan bahwa kombinasi yang diperoleh merupakan *global optimum*. Oleh karena itu, hasil *sequential tuning* dalam penelitian ini digunakan sebagai strategi optimasi yang sistematis dengan jumlah eksperimen yang lebih terbatas, bukan sebagai jaminan bahwa seluruh ruang kombinasi *hyperparameter* telah dieksplorasi.

Nilai *hyperparameter* yang diuji pada setiap tahap ditunjukkan pada Tabel III.

TABEL III. NILAI PENGUJIAN HYPERPARAMETER

Parameter	Nilai	Percobaan
Optimizer	Adam, RMSprop, SGD	3
Learning Rate	1e-3, 1e-4, 1e-5	3
Dropout	0,2 / 0,3 / 0,5	3
Batch Size	16, 32, 64	3
Epoch	20, 30, 50	3

3.5 Strategi Evaluasi

Performa model dievaluasi menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, serta *confusion matrix*. *Precision*, *recall*, dan *F1-score* dihitung berdasarkan pendekatan *weighted average* dengan mempertimbangkan jumlah sampel pada masing-masing kelas. *Confusion matrix* digunakan untuk menganalisis distribusi prediksi benar dan kesalahan klasifikasi pada setiap kelas.

Pada proses *sequential hyperparameter tuning*, sebanyak 800 citra yang pada struktur awal dataset ditetapkan sebagai data uji digunakan sebagai *validation set* untuk menentukan konfigurasi *hyperparameter* terbaik pada setiap tahap. Setelah seluruh konfigurasi *hyperparameter* ditetapkan, model dilatih kembali pada tahap *final training* menggunakan konfigurasi terbaik dari kandidat yang diuji. Selanjutnya, 800 citra yang sama digunakan kembali untuk evaluasi akhir model. Dengan demikian, data yang digunakan sebagai *validation set* selama proses optimasi tidak sepenuhnya terpisah dari data yang digunakan untuk evaluasi akhir.

Pada tahap *final training*, diterapkan beberapa mekanisme *callback*, yaitu *EarlyStopping*, *ReduceLROnPlateau*, dan *ModelCheckpoint*. *EarlyStopping* digunakan untuk menghentikan proses pelatihan ketika performa validasi tidak menunjukkan peningkatan sesuai kriteria yang ditetapkan, *ReduceLROnPlateau* digunakan untuk menurunkan *learning rate* ketika performa validasi mengalami

kondisi yang tidak membaik, sedangkan *ModelCheckpoint* digunakan untuk menyimpan bobot model dengan performa validasi terbaik selama proses pelatihan.

Penggunaan data validasi dan evaluasi akhir yang sama merupakan keterbatasan metodologis dalam penelitian ini karena belum menerapkan pembagian data *training*, *validation*, dan *testing* yang sepenuhnya independen. Kondisi tersebut dapat menyebabkan hasil evaluasi akhir memberikan estimasi performa yang lebih optimistis dibandingkan apabila model dievaluasi menggunakan *test set* yang tidak pernah digunakan selama proses pemilihan *hyperparameter*. Oleh karena itu, hasil evaluasi dalam penelitian ini diinterpretasikan dengan mempertimbangkan keterbatasan tersebut, terutama dalam menarik kesimpulan mengenai kemampuan generalisasi model pada data baru.

4. HASIL DAN PEMBAHASAN

Hasil penelitian yang diperoleh melalui penerapan pendekatan *transfer learning* dan *sequential hyperparameter tuning* pada arsitektur VGG-16 dan MobileNetV2 untuk klasifikasi penyakit ikan air tawar disajikan sebagai berikut.

4.1 Hasil Sequential Hyperparameter Tuning

Konfigurasi *hyperparameter* terbaik dan *validation accuracy* pada setiap tahap *sequential hyperparameter tuning* ditunjukkan pada Tabel IV. Pada setiap tahap, hanya *hyperparameter* yang sedang diuji yang diubah, sedangkan parameter lainnya menggunakan konfigurasi terbaik dari tahap sebelumnya. Nilai *accuracy* yang digunakan merupakan *validation accuracy* karena 800 citra tersebut digunakan sebagai *validation set* selama proses *tuning*.

Hasil menunjukkan bahwa Adam memberikan *validation accuracy* tertinggi pada kedua model, yaitu 98,87% pada VGG-16 dan 93,75% pada MobileNetV2. Pada tahap *learning rate*, VGG-16 memperoleh hasil terbaik pada 1×10^{-4} (98,75%), sedangkan MobileNetV2 pada 1×10^{-3} (98,75%). Selanjutnya, *dropout* 0,2 dan *batch size* 16 memberikan hasil tertinggi pada kedua model. Pada tahap *epoch*, 50 epoch menghasilkan *validation accuracy* tertinggi, yaitu 99,75% untuk VGG-16 dan 99,25% untuk MobileNetV2.

Berdasarkan hasil tersebut, konfigurasi yang dipilih dari kandidat yang diuji adalah Adam/ 1×10^{-4} /0,2/16/50 untuk VGG-16 dan Adam/ 1×10^{-3} /0,2/16/50 untuk MobileNetV2. Ringkasan hasil ditunjukkan pada Tabel IV.

TABEL IV. RINGKASAN PARAMETER TERBAIK

Parameter	Nilai Terbaik VGG-16	Accuracy VGG-16	Nilai Terbaik MobileNetV2	Accuracy MobileNetV2
Optimizer	Adam	98,87%	Adam	93,75%
Learning Rate	1e-4	98,75%	1e-3	98,75%
Dropout	0,2	99,37%	0,2	98,62%
Batch Size	16	99,37%	16	99,00%
Epoch	50	99,75%	50	99,25%

4.2 Hasil Final Training dan Evaluasi

Model VGG-16 dengan konfigurasi *hyperparameter* terbaik menghasilkan *accuracy* sebesar 99,62%, dengan *weighted average precision*, *recall*, dan *F1-score* masing-masing sebesar 1,00. Berdasarkan *confusion matrix*, terdapat tiga kesalahan klasifikasi, yaitu masing-masing satu citra pada kelas Bacterial Red Disease, Bacterial Gill Disease, dan White Spot Disease. Dengan demikian, model mengklasifikasikan 797 dari 800 citra dengan benar.

Sementara itu, MobileNetV2 memperoleh *accuracy* sebesar 99,37%, dengan *weighted average precision*, *recall*, dan *F1-score* masing-masing sebesar 0,99. Kesalahan klasifikasi terjadi pada kelas Bacterial Red Disease, Bacterial Gill Disease, dan White Spot Disease, masing-masing satu citra, serta Parasitic Diseases sebanyak dua citra. Dengan demikian, model mengklasifikasikan 795 dari 800 citra dengan benar.

Secara keseluruhan, kedua model menghasilkan nilai evaluasi yang tinggi. Selisih *accuracy* sebesar 0,25 poin persentase menunjukkan bahwa perbedaan hasil klasifikasi kedua model relatif kecil pada data evaluasi yang digunakan. Namun, karena 800 citra tersebut sebelumnya digunakan sebagai *validation set* selama proses *hyperparameter tuning*, hasil ini belum dapat digunakan sebagai estimasi generalisasi pada *test set* yang sepenuhnya independen.

TABEL V. RINGKASAN HASIL FINAL TRAINING

Metrik	VGG-16	MobileNetV2
Accuracy	99,62%	99,37%
Precision (W.Avg.)	1,00	0,99
Recall (W.Avg.)	1,00	0,99
F1-Score (W.Avg.)	1,00	0,99

4.3 Perbandingan Performa VGG-16 dan MobileNetV2

Selisih *accuracy* antara VGG-16 dan MobileNetV2 sebesar 0,25 poin persentase, yaitu masing-masing 99,62% dan 99,37%. Perbedaan tersebut menunjukkan bahwa hasil klasifikasi kedua model pada data evaluasi yang digunakan relatif berdekatan. MobileNetV2 memiliki *validation loss* sebesar 0,0353, sedangkan VGG-16 sebesar 0,0593. Selain itu, jumlah parameter *base model* MobileNetV2 sebanyak 2.587.976, lebih sedikit dibandingkan VGG-16 yang memiliki 21.139.528 parameter, atau sekitar 8,2 kali lebih sedikit. Perbedaan tersebut berkaitan dengan struktur MobileNetV2 yang menggunakan *depthwise separable convolution* dan *inverted residual block*, sedangkan VGG-16 menggunakan susunan konvolusi 3x3 secara berurutan.

Berdasarkan hasil tersebut, MobileNetV2 memiliki jumlah parameter *base model* yang lebih rendah, sedangkan VGG-16 memperoleh *accuracy* yang sedikit lebih tinggi pada data evaluasi yang digunakan. Perbedaan ini menunjukkan adanya perbedaan karakteristik antara performa klasifikasi dan kompleksitas model. Namun, karena penelitian ini tidak melakukan pengukuran langsung terhadap waktu inferensi, penggunaan memori, atau FLOPs/MACs pada perangkat target, kesimpulan mengenai efisiensi *deployment* pada perangkat dengan sumber daya terbatas masih perlu dibatasi pada perbandingan jumlah parameter model.

TABEL VI. PERBANDINGAN PERFORMA MODEL

Metrik	VGG-16	MobileNetV2
Accuracy	99,62%	99,37%
Validation Loss	0,0593	0,0353
Parameter Base	21.139.528 (80,64 MB)	2.587.976 (9,87 MB)
Konfigurasi Terbaik (Optimizer/LR/Dropout/Batch Size/Epoch)	Adam / 1e-4 / 0,2 / 16 / 50	Adam / 1e-3 / 0,2 / 16 / 50

4.4 Analisis Pengaruh Hyperparameter

Hasil *sequential hyperparameter tuning* menunjukkan bahwa setiap *hyperparameter* memberikan pengaruh yang berbeda terhadap performa kedua model. Pada tahap *optimizer*, Adam menghasilkan *validation accuracy* tertinggi, yaitu 98,87% pada VGG-16 dan 93,75% pada MobileNetV2, sedangkan SGD menghasilkan nilai terendah, masing-masing 70,50% dan 67,37%. Berdasarkan hasil

pengujian tersebut, Adam dipilih sebagai *optimizer* untuk tahap berikutnya.

Pada tahap *learning rate*, VGG-16 memperoleh *validation accuracy* tertinggi sebesar 98,75% pada 1×10^{-4} , sedangkan MobileNetV2 memperoleh 98,75% pada 1×10^{-3} . Ketika menggunakan 1×10^{-5} , *validation accuracy* turun menjadi 88,38% pada VGG-16 dan 69,50% pada MobileNetV2. Hasil ini menunjukkan bahwa kandidat *learning rate* terbaik berbeda antara kedua arsitektur.

Pada tahap *dropout*, nilai 0,2 menghasilkan *validation accuracy* tertinggi, yaitu 99,37% pada VGG-16 dan 98,62% pada MobileNetV2. Sementara itu, *batch size* 16 menghasilkan nilai tertinggi dibandingkan *batch size* 64, yaitu 99,37% pada VGG-16 dan 99,00% pada MobileNetV2. Hasil tersebut menunjukkan bahwa konfigurasi *dropout* dan *batch size* yang diuji menghasilkan perbedaan performa yang relatif kecil.

Pada tahap *epoch*, penggunaan 50 epoch menghasilkan *validation accuracy* tertinggi, yaitu 99,75% pada VGG-16 dan 99,25% pada MobileNetV2, dibandingkan 20 *epoch* yang masing-masing menghasilkan 98,00%. Berdasarkan konfigurasi yang diuji, penambahan jumlah *epoch* hingga 50 memberikan peningkatan performa pada kedua model. Namun, hasil ini tidak dapat digunakan untuk menyimpulkan bahwa 50 *epoch* merupakan nilai optimal secara global karena jumlah *epoch* yang diuji terbatas pada tiga kandidat.

4.5 Perbandingan dengan Penelitian Terdahulu

Performa model pada penelitian ini dibandingkan dengan beberapa penelitian terdahulu sebagai referensi. VGG-16 dan MobileNetV2 pada penelitian ini masing-masing memperoleh *accuracy* sebesar 99,62% dan 99,37%, sedangkan Biswas *et al.* [6] melaporkan akurasi 88,82% untuk VGG-16 dan 85,20% untuk MobileNetV2. Penelitian Biswas *et al.* [6] juga melaporkan akurasi 99,59% menggunakan pendekatan *ensemble CNN + SVM*. Selain itu, penelitian ini menggunakan delapan kelas, sedangkan penelitian Biswas *et al.* menggunakan tujuh kelas.

Perbedaan hasil tersebut menunjukkan adanya perbedaan performa pada masing-masing penelitian, tetapi tidak dapat digunakan sebagai perbandingan langsung untuk menyimpulkan keunggulan suatu metode, karena terdapat perbedaan dataset, jumlah kelas, preprocessing, augmentasi, serta konfigurasi eksperimen. Hasil penelitian ini lebih tepat digunakan sebagai referensi untuk melihat posisi performa model dalam konteks eksperimen yang dilakukan.

Perbandingan dengan penelitian terdahulu ditunjukkan pada Tabel VII.

TABEL VII. PERBANDINGAN PENELITIAN TERDAHULU

Penelitian	Model	Kelas	Akurasi
Biswas et al. (2024) [6]	VGG-16 tunggal	7	88,82%
Biswas et al. (2024) [6]	MobileNetV2 tunggal	7	85,20%
Biswas et al. (2024) [6]	Ensemble CNN + SVM	7	99,59%
Aryanti & Wibowo (2025) [17]	CNN	4	95,00%
Penelitian ini	VGG-16 + Sequential Tuning	8	99,62%
Penelitian ini	MobileNetV2 + Sequential Tuning	8	99,37%

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian penerapan *sequential hyperparameter tuning* pada arsitektur VGG-16 dan MobileNetV2 berbasis *transfer learning* untuk klasifikasi penyakit ikan air tawar, diperoleh beberapa kesimpulan sebagai berikut.

- Sequential hyperparameter tuning* menghasilkan konfigurasi terbaik dari kandidat yang diuji, yaitu Adam, learning rate 1×10^{-4} , dropout 0,2, batch size 16, dan 50 epoch untuk VGG-16, serta Adam, learning rate 1×10^{-3} , dropout 0,2, batch size 16, dan 50 epoch untuk MobileNetV2.
- Pada evaluasi akhir menggunakan 800 citra, VGG-16 memperoleh *accuracy* 99,62% dengan *weighted average precision*, *recall*, dan *F1-score* sebesar 1,00, sedangkan MobileNetV2 memperoleh *accuracy* 99,37% dengan masing-masing metrik sebesar 0,99. Selisih *accuracy* sebesar 0,25 poin persentase menunjukkan bahwa hasil kedua model relatif berdekatan pada data evaluasi yang digunakan.
- MobileNetV2 memiliki jumlah parameter *base model* sebesar 2.587.976, sedangkan VGG-16 memiliki 21.139.528 parameter, atau sekitar 8,2 kali lebih sedikit. Hasil tersebut menunjukkan perbedaan kompleksitas model berdasarkan jumlah parameter, sedangkan penelitian ini belum

melakukan pengukuran langsung terhadap waktu inferensi, penggunaan memori, atau FLOPs/MACs.

- d. Penelitian ini memiliki keterbatasan karena 800 citra yang digunakan sebagai *validation set* selama proses *hyperparameter tuning* juga digunakan kembali untuk evaluasi akhir. Dengan demikian, penelitian belum menerapkan pemisahan *training*, *validation*, dan *testing* yang sepenuhnya independen. Kondisi tersebut perlu dipertimbangkan dalam menginterpretasikan hasil evaluasi dan kemampuan generalisasi model.

5.2 Saran

Berdasarkan hasil penelitian serta keterbatasan yang masih terdapat dalam penelitian ini, beberapa saran yang dapat dipertimbangkan untuk pengembangan penelitian selanjutnya adalah sebagai berikut.

- Memperluas ukuran dan keragaman dataset dengan mempertimbangkan variasi spesies ikan, tingkat keparahan penyakit, pencahayaan, sudut pengambilan citra, serta kondisi lingkungan budidaya.
- Menggunakan skema pembagian data *training*, *validation*, dan *testing* yang sepenuhnya independen, sehingga evaluasi akhir dapat dilakukan pada data yang tidak digunakan selama proses pemilihan *hyperparameter*.
- Melakukan validasi menggunakan dataset eksternal atau citra yang diperoleh dari kondisi lapangan untuk menguji kemampuan model pada kondisi yang lebih beragam.
- Membandingkan *sequential hyperparameter tuning* dengan metode optimasi lain, seperti *Grid Search*, *Random Search*, atau *Bayesian Optimization*, menggunakan protokol eksperimen yang sama untuk memperoleh perbandingan yang lebih objektif.
- Melakukan pengukuran waktu inferensi, penggunaan memori, FLOPs/MACs, dan ukuran model atau pengujian langsung pada perangkat target untuk memberikan evaluasi efisiensi *deployment* yang lebih lengkap.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, masukan, dan kontribusi selama proses pelaksanaan penelitian ini sehingga penelitian dapat diselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] Badan Pusat Statistik, Statistik Sumber Daya Laut Dan Pesisir 2021. Jakarta, Indonesia: Badan Pusat Statistik, 2021.
- [2] *The State of World Fisheries and Aquaculture 2024*. Fao, 2024. Doi: 10.4060/Cd0683en.
- [3] Y. P. Huang and S. P. Khabusi, "A CNN-OSELM Multi-Layer Fusion Network with Attention Mechanism for Fish Disease Recognition in Aquaculture," *IEEE Access*, vol. 11, pp. 58729–58744, 2023, doi: 10.1109/ACCESS.2023.3280540.
- [4] M. S. Ahmed, T. T. Aurpa, and M. A. K. Azad, "Fish Disease Detection Using Image Based Machine Learning Technique in Aquaculture," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 5170–5182, Sep. 2022, doi: 10.1016/j.jksuci.2021.05.003.
- [5] S. N. Pauzi, M. G. Hassan, N. Yusoff, N. H. Harun, A. H. Abu Bakar, and B. C. Kua, "A review on image processing for fish disease detection," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Aug. 2021. doi: 10.1088/1742-6596/1997/1/012042.
- [6] S. Biswas et al., "Empirical Evaluation of Deep Learning Techniques for Fish Disease Detection in Aquaculture Systems: A Transfer Learning and Fusion-Based Approach," *IEEE Access*, vol. 12, pp. 176136–176154, 2024, doi: 10.1109/ACCESS.2024.3504283.
- [7] N. Hasan, S. Ibrahim, and A. Azlan, "Fish diseases detection using convolutional neural network (CNN)," *Int. J. Nonlinear Anal. Appl.*, vol. 13, pp. 2008–6822, 2022, doi: 10.22075/ijnaa.2022.5839.
- [8] H. Tamut, R. Ghosh, K. Gosh, and M. A. S. Siddique, "Enhancing Disease Detection in the Aquaculture Sector Using Convolutional Neural Networks Analysis," *Aquaculture Journal*, vol. 5, no. 1, Mar. 2025, doi: 10.3390/aquacj5010006.
- [9] A. Mns, P. E. eFishery, and M. Teknologi Nusantara, "Fish Disease Detection Of Epizootic Ulcerative Syndrome Using Deep Learning Image Processing Technique," vol. 8, no. 1, pp. 23–34, 2023, doi: 10.17501/23861282.2023.8102.
- [10] R. Afridiansyah, D. Rosal, and I. M. Setiadi, "Comparison of EfficientNetB1 Model Effectiveness in Identifying Fish Diseases in South Asian Fish Diseases and Salmon Fish Diseases," 2024. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [11] L. Zahro, H. Syaputra, S. Oktarina, M. T. Roseno, and E. Supratman, "Optimisasi Hyperparameter pada Model Klasifikasi Jenis Bunga dengan arsitektur MobileNetV3-Small," 2024.

- [12] A. Julianto, A. Sunyoto, D. Ferry, and W. Wibowo, "Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi (Optimization Of Convolutional Neural Network Hyperparameters For Classification Of Rice Plant Diseases)." [Online]. Available: <https://www.kaggle.com/tedisetiady/leaf-rice-dis->
- [13] E. K. D. Kette, "Hyperparameter Optimization In Machine Learning Models On Sky Survey Data Classification," *Jurnal Komputer dan Informatika*, vol. 13, no. 2, pp. 77–84, Sep. 2025, doi: 10.35508/jicon.v13i2.18493.
- [14] S. -, Anwar Fitrianto, and Bagus Sartono, "Hyperparameter Optimization of an Image Classification Model for Beef and Pork Using Convolutional Neural Networks," *EKSAKTA: Journal of Sciences and Data Analysis*, Oct. 2025, doi: 10.20885/eksakta.vol6.iss2.art6.
- [15] J. Rusman, B. Z. Haryati, and A. Michael, "Optimisasi Hiperparameter Tuning pada Metode Support Vector Machine untuk Klasifikasi Tingkat Kematangan Buah Kopi," *Jurnal Komputer dan Informatika*, vol. 11, no. 2, pp. 195–202, Oct. 2023, doi: 10.35508/jicon.v11i2.12571.
- [16] M. N. Gea, W. Wanayumini, and R. Rosnelly, "Impact of Hyperparameter Tuning on CNN-Based Algorithm for MRI Brain Tumor Classification," *JURNAL TEKNIK INFORMATIKA*, vol. 18, no. 1, pp. 87–100, Apr. 2025, doi: 10.15408/jti.v18i1.44147.
- [17] R. Aryanti and A. P. W. Wibowo, "Building a fish disease detection application based on Smart Fishery with Image Processing and Deep Learning," *Brilliance: Research of Artificial Intelligence*, vol. 5, no. 1, pp. 66–73, Apr. 2025, doi: 10.47709/brilliance.v5i1.5719.