

**TESIS**  
**KLASIFIKASI KESEGERAN IKAN BERDASARKAN CITRA**  
**MATA MENGGUNAKAN EFFICIENTNET-B0 DENGAN**  
**OPTIMASI HYPERPARAMETER OPTUNA**



disusun oleh

**TRIO AGUNG PURWANTO**

**24.51.1563**

**Konsentrasi : Business Intelligence**

**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS AMIKOM YOGYAKARTA**  
**YOGYAKARTA**

**2026**

**TESIS**  
**KLASIFIKASI KESEGERAN IKAN BERDASARKAN CITRA**  
**MATA MENGGUNAKAN EFFICIENTNET-B0 DENGAN**  
**OPTIMASI HYPERPARAMETER OPTUNA**

**FISH FRESHNESS CLASSIFICATION BASED ON EYE IMAGES**  
**USING EFFICIENTNET-B0 WITH OPTUNA HYPERPARAMETER**  
**OPTIMIZATION**

Diajukan untuk memenuhi salah satu syarat mencapai derajat Pascasarjana  
Program Studi Informatika



disusun oleh

**TRIO AGUNG PURWANTO**

**24.51.1563**

**Konsentrasi : Business Intelligence**

**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS AMIKOM YOGYAKARTA**  
**YOGYAKARTA**

**2026**

HALAMAN PERSETUJUAN

KLASIFIKASI KESEGARAN IKAN BERDASARKAN CITRA MATA  
MENGUNAKAN EFFICIENTNET-B0 DENGAN OPTIMASI  
HYPERPARAMETER OPTUNA

FISH FRESHNESS CLASSIFICATION BASED ON EYE IMAGES USING  
EFFICIENTNET-B0 WITH OPTUNA HYPERPARAMETER  
OPTIMIZATION


yang disusun dan diajukan oleh

**TRIO AGUNG PURWANTO**

24.51.1563

telah disetujui oleh Dosen Pembimbing Tesis  
pada tanggal 5 Januari 2026

Dosen Pembimbing,



Dhan Ariatmanto, S.Kom., M.Kom., Ph.D.  
NIK. 190302197

HALAMAN PENGESAHAN

KLASIFIKASI KESEGERAN IKAN BERDASARKAN CITRA MATA  
MENGUNAKAN EFFICIENTNET-B0 DENGAN OPTIMASI  
HYPERPARAMETER OPTUNA

FISH FRESHNESS CLASSIFICATION BASED ON EYE IMAGES USING  
EFFICIENTNET-B0 WITH OPTUNA HYPERPARAMETER  
OPTIMIZATION

yang disusun dan diajukan oleh

**TRIO AGUNG PURWANTO**

**24.51.1563**

Telah dipertahankan di depan Dewan Penguji  
pada tanggal 5 Januari 2026

Susunan Dewan Penguji

Nama Penguji

Dr. Andi Sunyoto, M.Kom.  
NIK. 190302052

Dr. Kumara Ari Yuana, S.T., M.T.  
NIK. 190302575

Dhani Ariatmanto, S.Kom., M.Kom., Ph.D.  
NIK. 190302197

Tanda Tangan



Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer  
Tanggal 5 Januari 2026

DEKAN FAKULTAS ILMU KOMPUTER



Prof. Dr. Kusriani, M.Kom.  
NIK. 190302106

## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Trio Agung Purwanto  
NIM : 24.51.1563

Menyatakan bahwa Tesis dengan judul berikut:

**Klasifikasi Kesegaran Ikan Berdasarkan Citra Mata Menggunakan Efficientnet-B0 Dengan Optimasi Hyperparameter Optuna**

Dosen Pembimbing : Dhani Ariatmanto, S.Kom., M.Kom., Ph.D.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 5 Januari 2026

Yang Menyatakan,



Trio Agung Purwanto

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan tesis yang berjudul **“Klasifikasi Kesegaran Ikan Berdasarkan Citra Mata Menggunakan EfficientNet-B0 dengan Optimasi Hyperparameter Optuna”**. Tesis ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan pada jenjang magister.

Penulis menyadari bahwa penyusunan tesis ini tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Dr. Andi Sunyoto, M.Kom., selaku Dosen Penguji I, atas saran, masukan, dan penilaian yang telah diberikan demi penyempurnaan tesis ini.
2. Dr. Kumara Ari Yuana, S.T., M.T., selaku Dosen Penguji II, atas kritik dan masukan yang membangun selama proses pengujian tesis.
3. Dhani Ariatmanto, S.Kom., M.Kom., Ph.D., selaku Dosen Pembimbing sekaligus Dosen Penguji III, atas bimbingan, arahan, serta kesabaran yang diberikan kepada penulis selama proses penyusunan tesis ini.

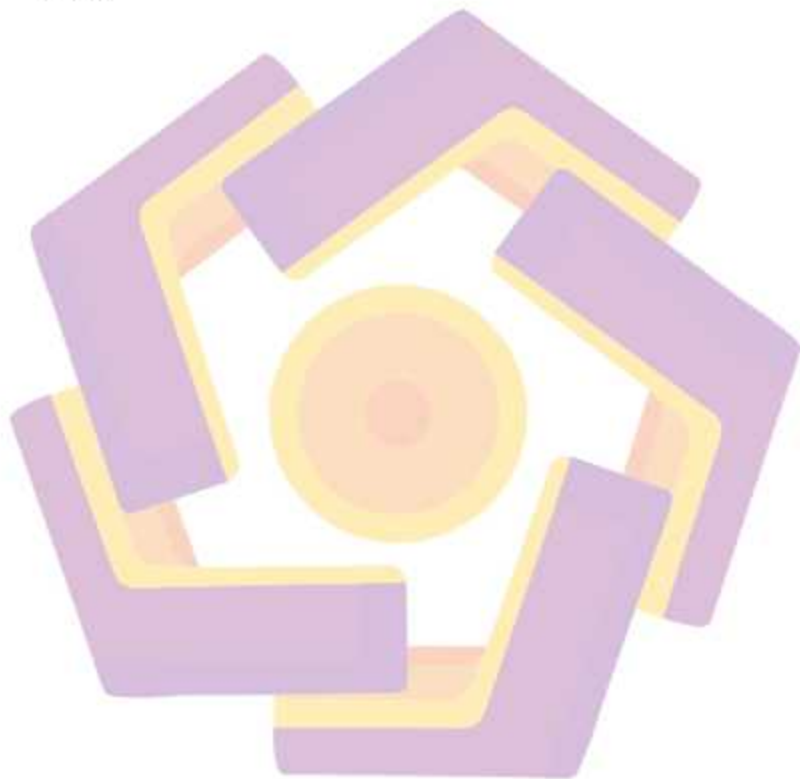
Penulis juga menyampaikan terima kasih kepada orang tua dan keluarga atas doa, dukungan moral, serta motivasi yang senantiasa diberikan. Ucapan terima kasih turut penulis sampaikan kepada semua pihak yang telah membantu secara langsung maupun tidak langsung dalam penyelesaian tesis ini.

Penulis menyadari bahwa tesis ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan demi

perbaikan di masa mendatang. Semoga tesis ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan dan pihak-pihak yang berkepentingan.

Yogyakarta, 20 Januari 2026

Penulis



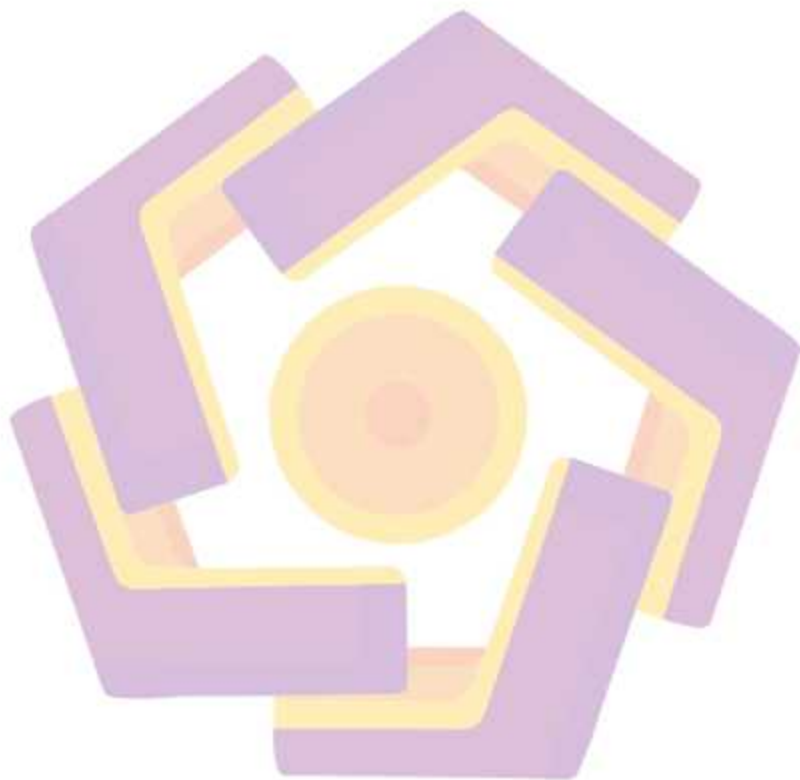
## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN .....	iii
HALAMAN PERNYATAAN KEASLIAN TESIS .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN.....	xiii
INTISARI .....	xiv
ABSTRACT.....	xv
<b>BAB 1 PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah .....	5
1.4 Tujuan Penelitian .....	5
1.5 Manfaat Penelitian .....	6
1.5.1 Manfaat Akademis .....	6
1.5.2 Manfaat Praktis .....	6
1.5.3 Manfaat Ekonomis .....	7
1.5.4 Manfaat Sosial .....	7
<b>BAB 2 TINJAUAN PUSTAKA</b> .....	<b>8</b>
2.1 Tjjuan Pustaka.....	8
2.2 Keaslian Penelitian .....	12
2.3 Landasan Teori .....	13
2.3.1 EfficientNet.....	13
2.3.2 Optuna dan Algoritma TPE untuk Optimasi Hiperparameter.....	16

2.3.3 Optimizer dalam Pembelajaran Jaringan Saraf.....	17
2.3.3.1. Stochastic Gradient Descent (SGD).....	17
2.3.3.2. Adaptive Moment Estimation (Adam).....	18
2.3.3.2. AdamW .....	19
2.3.4 Weight Decay dalam Pembelajaran Jaringan Saraf .....	20
2.3.5 Dropout dalam Jaringan Saraf Tiruan.....	22
2.3.6 Metrik Evaluasi Kinerja Model Klasifikasi .....	24
<b>BAB 3 METODE PENELITIAN .....</b>	<b>27</b>
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	27
3.2. Metode Pengumpulan Data.....	28
3.3. Metode Analisis Data.....	28
3.4. Alur Penelitian .....	29
3.4.1 Perumusan Masalah dan Studi Literatur .....	32
3.4.2 Pengumpulan Dataset.....	33
3.4.3 Pembagian Dataset.....	33
3.4.4 Pemodelan Deep Learning .....	34
3.4.5 Tuning Hyperparameter dengan Optuna.....	34
3.4.6 Evaluasi Hasil Tuning Hyperparameter.....	35
3.4.7 Penarikan Kesimpulan .....	35
<b>BAB 4 HASIL PENELITIAN DAN PEMBAHASAN .....</b>	<b>37</b>
4.1 Hasil Pengumpulan dan Persiapan Dataset.....	37
4.2 Pembagian Dataset.....	39
4.3 Pelatihan dan Evaluasi Model.....	42
4.3.1 Arsitektur Model .....	42
4.3.2 Pelatihan Model Baseline dengan Parameter Default.....	43
4.4 Pengaruh Optuna terhadap Generalisasi EfficientNet-B0 .....	45
4.4.1 Penjelasan Ruang Pencarian Hyperparameter .....	46

4.4.2	Gap Penalty Score	49
4.4.3	Alur Optimasi Hyperparameter Menggunakan Optuna	50
4.5	Jumlah Trial dalam Eksplorasi Hyperparameter dan Stabilitas Model	51
4.5.1	Evaluasi Konfigurasi 20 Trial	52
4.5.2	Evaluasi Konfigurasi 50 Trial	53
4.5.3	Evaluasi Konfigurasi 100 Trial	53
4.5.4	Penetapan Jumlah Trial Optimal	54
4.6	Konfigurasi Optimal EfficientNet-B0 Berdasarkan Generalisasi	55
4.6.1	Analisis Hasil Percobaan 1	55
4.6.2	Analisis Hasil Percobaan 2	57
4.6.3	Analisis Hasil Percobaan 3	58
4.6.4	Penetapan Konfigurasi Akhir	60
4.6.5	Perbandingan Performa Model Baseline dan Model Optuna	60
4.7	Kinerja dan Konsistensi Model Tanpa Augmentasi Data	62
4.8	Kinerja Klasifikasi Kurva Pelatihan dan Confusion Matrix	63
4.9	Analisis Robustness EfficientNet-B0 Tanpa Oversampling	68
4.9.1	Analisis Kasus Salah Klasifikasi (Incorrect Detection Analysis)	68
4.9.2	Analisis Grad-CAM pada Klasifikasi Benar	71
4.9.3	Analisis Grad-CAM pada Klasifikasi Salah	73
4.9.4	Robustness terhadap Geometric Attack	74
4.9.5	Robustness terhadap Noise Attack	76
4.9.6	Robustness terhadap Filter Attack	77
4.10	Evaluasi generalisasi model	78
4.11	Perbandingan dengan Penelitian Sebelumnya	81
<b>BAB 5</b>	<b>PENUTUP</b>	<b>84</b>
5.1	Kesimpulan	84
5.2	Saran	85

DAFTAR PUSTAKA .....	86
LAMPIRAN.....	89



## DAFTAR TABEL

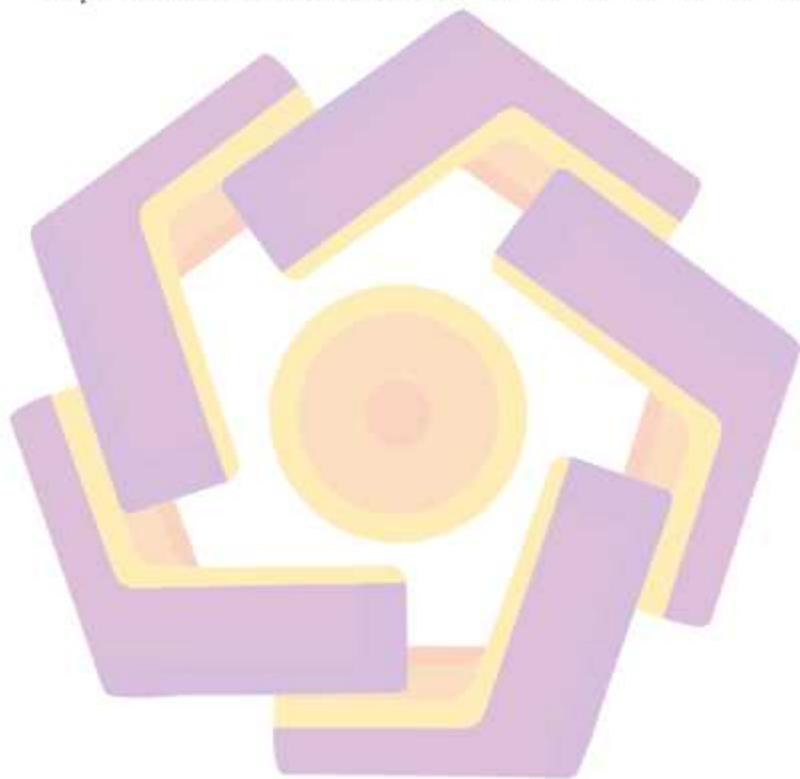
Tabel 2.1 Matriks literatur review dan posisi penelitian.....	12
Tabel 2.2 Perbandingan Varian EfficientNet.....	15
Tabel 4.1 Distribusi Dataset Berdasarkan Kelas.....	37
Tabel 4.2 Proporsi Pembagian Dataset.....	40
Tabel 4.3 Ruang Pencarian (Search Space) Hyperparameter.....	45
Tabel 4.4 Hasil Evaluasi Gap pada Konfigurasi 20 Trial.....	52
Tabel 4.5 Hasil Evaluasi Gap pada Konfigurasi 50 Trial.....	53
Tabel 4.6 Hasil Evaluasi Gap pada Konfigurasi 100 Trial.....	53
Tabel 4.7 Hasil Terbaik Percobaan 1 (Top-10 dari 100 Trial).....	55
Tabel 4.8 Hasil Terbaik Percobaan 2 (Top-10 dari 100 Trial).....	57
Tabel 4.9 Hasil Terbaik Percobaan 3 (Top-10 dari 100 Trial).....	58
Tabel 4.10 Hasil Pelatihan Ulang Model Terbaik (5 Kali Run).....	62
Tabel 4.11 Sampel incorrect classification pada data uji.....	69
Tabel 4.12 Ringkasan hasil evaluasi LSO disajikan.....	79
Tabel 4.13 Perbandingan Penelitian Sebelumnya dengan Penelitian Ini.....	81

## DAFTAR GAMBAR

Gambar 2.1	Arsitektur EfficientNet.....	14
Gambar 3.1	Alur Penelitian .....	30
Gambar 4.1	Distribusi Dataset Berdasarkan Kelas.....	38
Gambar 4.2	Distribusi Dataset Setelah Pembagian Data Latih, Validasi, dan Uji 41	
Gambar 4.3	Train histori, Train Acuaracy vs Val Acuaracy Baseline/Default .....	44
Gambar 4.4	Train histori, Train Acuaracy vs Val Acuaracy .....	64
Gambar 4.5	Confusion Matrix .....	66
Gambar 4.6	Grad-CAM pada Klasifikasi Benar Chanos chanos (Fresh) .....	71
Gambar 4.7	Grad-CAM pada Klasifikasi Salah Chanos chanos (Fresh).....	73
Gambar 4.8	Geometrical Attack pada Nibea albiflora – Highly Fresh.....	74
Gambar 4.9	Noise Attack pada Nibea albiflora – Highly Fresh.....	76
Gambar 4.10	Filter Attack pada Nibea albiflora – Highly Fresh.....	77

## DAFTAR LAMPIRAN

Lampiran 1 Data Percobaan Pertama 100 Trial .....	89
Lampiran 2 Data Percobaan Kedua 100 Trial.....	91
Lampiran 3 Data Percobaan Ketiga 100 Trial .....	94
Lampiran 4 Data Kesalahan Klasifikasi .....	98



## INTISARI

Penilaian kesegaran ikan merupakan aspek penting dalam menjaga mutu dan keamanan produk perikanan, namun metode konvensional berbasis pengamatan visual manusia masih bersifat subjektif dan tidak konsisten. Penelitian ini bertujuan untuk mengklasifikasikan tingkat kesegaran ikan berdasarkan citra mata menggunakan model deep learning EfficientNet-B0 dengan optimasi hiperparameter Optuna, dengan fokus pada stabilitas pelatihan dan kemampuan generalisasi lintas spesies. Dataset yang digunakan adalah The Freshness of the Fish Eyes yang terdiri dari citra mata ikan dari 8 spesies dengan 3 tingkat kesegaran, sehingga membentuk 24 kelas klasifikasi. Penelitian ini tidak menerapkan optimasi hiperparameter dilakukan menggunakan Optuna berbasis algoritma Tree-structured Parzen Estimator (TPE). Hasil eksperimen menunjukkan bahwa model mencapai akurasi pengujian sebesar 79,87%, dengan akurasi data latih sebesar 83,61% dan akurasi data validasi sebesar 81,17%, sehingga diperoleh generalization gap sebesar 2,45%. Evaluasi generalisasi lintas spesies dilakukan menggunakan skema leave-source-out (LSO) pada 8 skenario pengujian, yang menunjukkan bahwa model mampu mempertahankan performa klasifikasi pada spesies yang tidak dilibatkan dalam proses pelatihan. Selain itu, pelatihan ulang model terbaik sebanyak 5 kali menunjukkan performa yang konsisten, yang mengindikasikan stabilitas pembelajaran yang baik. Penelitian ini menyimpulkan bahwa EfficientNet-B0 yang dioptimasi menggunakan Optuna mampu memberikan klasifikasi kesegaran ikan yang stabil dan konsisten pada dataset multi-spesies tanpa penerapan augmentasi data, serta memiliki kemampuan generalisasi intrinsik terhadap distribusi asli dataset.

**Kata kunci:** kesegaran ikan, citra mata, EfficientNet-B0, optimasi hiperparameter, Optuna.

## ABSTRACT

*Fish freshness assessment is a critical factor in ensuring the quality and safety of fishery products; however, conventional visual inspection methods remain subjective and inconsistent. This study aims to classify fish freshness levels based on eye images using the EfficientNet-B0 deep learning model with Optuna-based hyperparameter optimization, with a focus on training stability and cross-species generalization capability. The dataset used in this study is The Freshness of the Fish Eyes, which consists of eye images from 8 fish species with 3 freshness levels, resulting in 24 classification classes. No data augmentation techniques were applied. Hyperparameter optimization was performed using Optuna with the Tree-structured Parzen Estimator (TPE) algorithm. Experimental results show that the optimized model achieved a test accuracy of 79.87%, with training and validation accuracies of 83.61% and 81.17%, respectively, resulting in a generalization gap of 2.45%. Cross-species generalization was evaluated using the leave-source-out (LSO) scheme across 8 testing scenarios, demonstrating that the model was able to maintain classification performance on fish species not included during training. Furthermore, retraining the best-performing model for 5 independent runs produced consistent results, indicating stable learning behavior. This study concludes that EfficientNet-B0 optimized with Optuna is capable of providing stable and consistent fish freshness classification on a multi-species dataset without data augmentation, while exhibiting intrinsic generalization capability toward the original data distribution.*

**Keyword:** fish freshness, eye images, EfficientNet-B0, hyperparameter optimization, Optuna, cross-species generalization

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Penilaian kesegaran ikan merupakan aspek fundamental dalam industri perikanan global, khususnya bagi negara pengekspor seperti Indonesia yang dituntut untuk memenuhi standar mutu dan keamanan pangan internasional [1], [2]. Tingkat kesegaran ikan tidak hanya berkaitan dengan aspek kesehatan dan keamanan konsumsi, tetapi juga berdampak langsung terhadap nilai ekonomi produk, daya saing ekspor, serta tingkat kepercayaan pasar internasional terhadap komoditas perikanan Indonesia. Dalam konteks perdagangan global yang semakin kompetitif dan diatur oleh regulasi food safety yang ketat, dibutuhkan sistem penilaian kesegaran ikan yang objektif, konsisten, dan efisien agar kualitas produk perikanan dapat dipertahankan secara berkelanjutan.

Hingga saat ini, penilaian kesegaran ikan masih banyak dilakukan menggunakan metode organoleptik, yaitu pengamatan visual manusia terhadap karakteristik fisik ikan seperti kejernihan mata, warna insang, dan tekstur daging [3], [4]. Meskipun metode ini relatif sederhana dan mudah diterapkan, pendekatan organoleptik memiliki keterbatasan mendasar karena sangat bergantung pada pengalaman dan persepsi individu. Penilaian yang bersifat subjektif tersebut rentan terhadap inkonsistensi antar penilai, serta dipengaruhi oleh faktor kelelahan, kondisi lingkungan, dan variasi pencahayaan, terutama ketika diterapkan pada skala industri yang menuntut kecepatan dan akurasi tinggi.

Perkembangan pesat di bidang computer vision dan deep learning membuka peluang besar untuk mengotomatisasi proses penilaian kesegaran ikan secara lebih objektif dan konsisten secara kuantitatif. Pendekatan berbasis citra memungkinkan sistem untuk mengekstraksi dan mempelajari pola visual secara konsisten dari data, sehingga dapat mengurangi ketergantungan pada penilaian subjektif manusia [5]. Sejumlah penelitian sebelumnya telah menerapkan model deep learning untuk klasifikasi kesegaran ikan berbasis citra mata, karena bagian mata ikan memiliki keterkaitan visual yang kuat dengan tingkat kesegaran selama proses penyimpanan.

Penelitian terdahulu menunjukkan bahwa pendekatan deep learning berbasis citra mata ikan memiliki potensi yang menjanjikan. Anas et al. (2021) menggunakan Tiny YOLOv2 untuk mengklasifikasikan tiga tingkat kesegaran ikan, namun performa yang diperoleh masih relatif rendah akibat keterbatasan dataset dan ketidakakuratan pelabelan organoleptic [3]. Prasetyo et al. (2022) mengembangkan arsitektur MobileNetV1 dengan Bottleneck dan Expansion (MB-BE) pada dataset Freshness of the Fish Eyes (FFE) yang bersifat multi-spesies dan multi-kelas, tetapi akurasi yang dicapai masih tergolong moderat karena perbedaan visual antar kelas yang sangat halus [6]. Guo et al. (2022) meningkatkan performa MobileNetV1 melalui penambahan attention mechanism, namun keberhasilan tersebut sangat bergantung pada penggunaan augmentasi data dan pengaturan parameter tertentu [7]. Sementara itu, Yildiz et al. (2024) memperoleh akurasi yang lebih tinggi dengan menggabungkan deep learning sebagai feature extractor dan machine learning sebagai classifier, meskipun pendekatan ini bukan merupakan model CNN end-to-end dan memiliki kompleksitas komputasi yang lebih tinggi [8].

Berdasarkan tinjauan tersebut, dapat disimpulkan bahwa meskipun pendekatan deep learning berbasis citra mata ikan telah menunjukkan kinerja yang cukup baik, sebagian besar penelitian sebelumnya masih sangat bergantung pada intervensi augmentasi data, modifikasi arsitektur, atau kombinasi dengan metode machine learning. Selain itu, aspek stabilitas pelatihan, kemampuan generalisasi model, serta optimasi hiperparameter yang dilakukan secara sistematis masih belum banyak dikaji secara eksplisit.

Berdasarkan kondisi tersebut, penelitian ini mengusulkan penggunaan EfficientNet-B0 sebagai model klasifikasi utama untuk mengidentifikasi tingkat kesegaran ikan berdasarkan citra mata pada dataset multi-spesies. EfficientNet-B0 merupakan arsitektur CNN yang dirancang untuk mencapai keseimbangan optimal antara akurasi dan efisiensi komputasi melalui prinsip compound scaling [9]. Untuk memperoleh konfigurasi pelatihan yang optimal dan stabil, model EfficientNet-B0 dipadukan dengan Optuna, sebuah kerangka kerja optimasi hiperparameter otomatis berbasis Tree-structured Parzen Estimator (TPE) yang memungkinkan eksplorasi ruang hiperparameter secara sistematis dan efisien [10].

Selain aspek arsitektur dan optimasi model, metode evaluasi yang digunakan juga memiliki peran penting dalam menilai kemampuan generalisasi model deep learning, khususnya pada dataset yang bersifat multi-sumber atau multi-domain. Penelitian terbaru menunjukkan bahwa skema evaluasi konvensional seperti K-fold cross-validation cenderung menghasilkan estimasi performa yang terlalu optimistis ketika model diuji pada sumber data yang tidak dilibatkan dalam proses pelatihan. Leinonen et al. (2024) menunjukkan bahwa pendekatan leave-

source-out cross-validation (LSO) mampu memberikan estimasi performa yang lebih realistis dan tidak bias untuk mengevaluasi generalisasi model terhadap sumber baru, meskipun dengan variansi yang lebih tinggi. Oleh karena itu, penggunaan skema evaluasi berbasis LSO menjadi relevan untuk menilai kemampuan generalisasi model klasifikasi kesegaran ikan pada spesies yang tidak dilibatkan selama proses pelatihan [11].

Berbeda dengan sebagian penelitian terdahulu, penelitian ini secara sengaja tidak menerapkan teknik augmentasi data maupun oversampling sintetis, sehingga performa model sepenuhnya merefleksikan kemampuan generalisasi terhadap distribusi asli dataset. Fokus penelitian tidak hanya diarahkan pada pencapaian akurasi, tetapi juga pada stabilitas pelatihan dan konsistensi performa model, sehingga sistem yang dihasilkan diharapkan mampu memberikan hasil klasifikasi yang lebih objektif, adaptif terhadap variasi visual alami, serta efisien dari sisi waktu dan sumber daya komputasi.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang penelitian yang menekankan perlunya sistem penilaian kesegaran ikan yang objektif dan konsisten berbasis citra mata ikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana kinerja model EfficientNet-B0 dalam mengklasifikasikan tingkat kesegaran ikan berbasis citra mata pada dataset multi-spesies tanpa penerapan augmentasi data?
2. Bagaimana pengaruh optimasi hiperparameter berbasis Optuna terhadap stabilitas pelatihan dan kemampuan generalisasi model EfficientNet-B0 yang dievaluasi menggunakan skema leave-source-out (LSO) pada data dari

spesies yang tidak dilibatkan dalam proses pelatihan?

### 1.3 Batasan Masalah

Untuk menjaga fokus dan ruang lingkup penelitian, maka ditetapkan batasan sebagai berikut:

1. Penelitian hanya menggunakan citra mata ikan sebagai indikator utama kesegaran.
2. Model deep learning yang digunakan adalah EfficientNet-B0 tanpa modifikasi struktur arsitektur.
3. Dataset yang digunakan adalah The Freshness of the Fish Eyes dari Mandeley Data yang terdiri dari 8 spesies ikan dengan 3 tingkat kesegaran (Highly Fresh, Fresh, dan Not Fresh).
4. Penelitian tidak menerapkan augmentasi data.
5. Optimasi hiperparameter dilakukan menggunakan Optuna dengan pendekatan Tree-structured Parzen Estimator (TPE).
6. Penelitian tidak mencakup implementasi sistem real-time maupun integrasi perangkat keras.

### 1.4 Tujuan Penelitian

1. Mengimplementasikan model EfficientNet-B0 untuk melakukan klasifikasi tingkat kesegaran ikan berbasis citra mata ikan pada dataset multi-spesies
2. Mengevaluasi kinerja model EfficientNet-B0 berdasarkan metrik performa pada data uji, stabilitas pelatihan, serta kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya.

3. Menganalisis pengaruh optimasi hiperparameter berbasis Optuna terhadap kestabilan pembelajaran dan selisih performa antara data latih dan data validasi sebagai indikator overfitting.
4. Mengevaluasi konsistensi performa model yang telah dioptimasi pada kondisi pelatihan tanpa penerapan augmentasi data.
5. Menyediakan dasar eksperimental bagi pengembangan sistem penilaian kesegaran ikan otomatis yang objektif dan konsisten berbasis citra, tanpa membahas aspek implementasi real-time.

## **1.5 Manfaat Penelitian**

Penelitian ini diharapkan dapat memberikan beberapa manfaat, baik dari segi akademis maupun praktis, antara lain:

### **1.5.1 Manfaat Akademis**

Memberikan kontribusi ilmiah dalam bidang computer vision dan deep learning melalui evaluasi sistematis terhadap stabilitas pelatihan dan kemampuan generalisasi lintas spesies pada model EfficientNet-B0 yang dioptimasi menggunakan Optuna, khususnya pada klasifikasi kesegaran ikan berbasis citra mata tanpa penerapan augmentasi data.

### **1.5.2 Manfaat Praktis**

Memberikan referensi metodologis bagi peneliti dan praktisi dalam mengembangkan sistem klasifikasi kesegaran ikan berbasis citra mata yang bersifat objektif dan konsisten, terutama dalam konteks dataset multi-spesies dengan keterbatasan data dan tanpa intervensi augmentasi buatan.

### **1.5.3 Manfaat Ekonomis**

Memberikan potensi efisiensi pada tahap evaluasi kualitas produk perikanan melalui pendekatan klasifikasi berbasis citra, dengan catatan bahwa penelitian ini masih berada pada tahap eksperimental dan belum mencakup implementasi sistem operasional di lingkungan industri.

### **1.5.4 Manfaat Sosial**

Memberikan dasar ilmiah bagi pengembangan teknologi penilaian kesegaran ikan yang lebih objektif dan konsisten, yang berpotensi mendukung peningkatan kepercayaan masyarakat terhadap mutu produk perikanan, tanpa mengklaim dampak langsung pada skala implementasi publik.



## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka

Penilaian tingkat kesegaran ikan merupakan isu penting dalam industri perikanan global, khususnya bagi negara-negara pengekspor seperti Indonesia. Hasan et al. (2024) menegaskan bahwa kualitas kesegaran ikan tidak hanya berpengaruh terhadap aspek keamanan pangan dan kesehatan konsumen, tetapi juga berkaitan langsung dengan daya saing produk di pasar internasional serta tingkat kepercayaan mitra dagang [1]. Oleh karena itu, dibutuhkan metode penilaian kesegaran ikan yang objektif, konsisten, dan dapat direplikasi.

Dalam praktik lapangan, metode yang masih banyak digunakan adalah uji organoleptik, yaitu penilaian visual manusia terhadap karakteristik fisik ikan seperti kejernihan mata, warna insang, dan tekstur daging [3]. Meskipun metode ini relatif sederhana dan murah, uji organoleptik memiliki keterbatasan mendasar berupa subjektivitas penilai, inkonsistensi antar penguji, serta rendahnya efisiensi ketika diterapkan pada skala industri [4]. Keterbatasan tersebut mendorong pengembangan pendekatan otomatis berbasis teknologi pengolahan citra.

Seiring dengan perkembangan computer vision dan deep learning, berbagai pendekatan berbasis citra mulai dikembangkan untuk menggantikan metode manual. Beberapa penelitian memanfaatkan citra mata ikan sebagai indikator utama kesegaran, karena perubahan fisiologis pada mata terjadi secara signifikan seiring penurunan kualitas ikan. Anas et al. (2021) menerapkan Tiny YOLOv2 untuk

mengklasifikasikan tiga tingkat kesegaran ikan, namun performa yang diperoleh masih relatif rendah akibat keterbatasan dataset, ketidakseimbangan kelas, serta ketergantungan pada pelabelan organoleptik [3].

Prasetyo et al. (2022) mengembangkan model ringan berbasis MobileNetV1 dengan pendekatan Depthwise Separable Convolution Bottleneck with Expansion (MB-BE) untuk klasifikasi kesegaran ikan pada dataset Freshness of the Fish Eyes (FFE) yang bersifat multi-spesies dan multi-kelas. Meskipun model tersebut mampu meningkatkan efisiensi komputasi dan menunjukkan kinerja yang lebih baik dibandingkan beberapa arsitektur CNN standar, akurasi yang dicapai masih tergolong moderat karena perbedaan visual antar kelas yang sangat halus. Penelitian ini berfokus pada pengembangan arsitektur jaringan tanpa mengevaluasi stabilitas pelatihan maupun optimasi hiperparameter secara sistematis [6].

Guo et al. (2022) mengusulkan peningkatan performa MobileNetV1 melalui penambahan berbagai attention mechanism seperti SENet dan ECA. Hasil penelitian menunjukkan bahwa mekanisme attention mampu meningkatkan akurasi dan mempercepat konvergensi model, dengan akurasi validasi mencapai sekitar 73%. Namun demikian, performa model sangat bergantung pada penerapan augmentasi data dan pengaturan parameter tertentu, serta tidak disertai analisis stabilitas pelatihan lintas eksperimen [7].

Sementara itu, Yildiz et al. (2024) mengombinasikan model deep learning sebagai feature extractor (VGG19 dan SqueezeNet) dengan berbagai metode machine learning sebagai classifier. Kombinasi VGG19 dan ANN menghasilkan akurasi tertinggi sebesar 77,3% pada dataset FFE. Meskipun memberikan performa

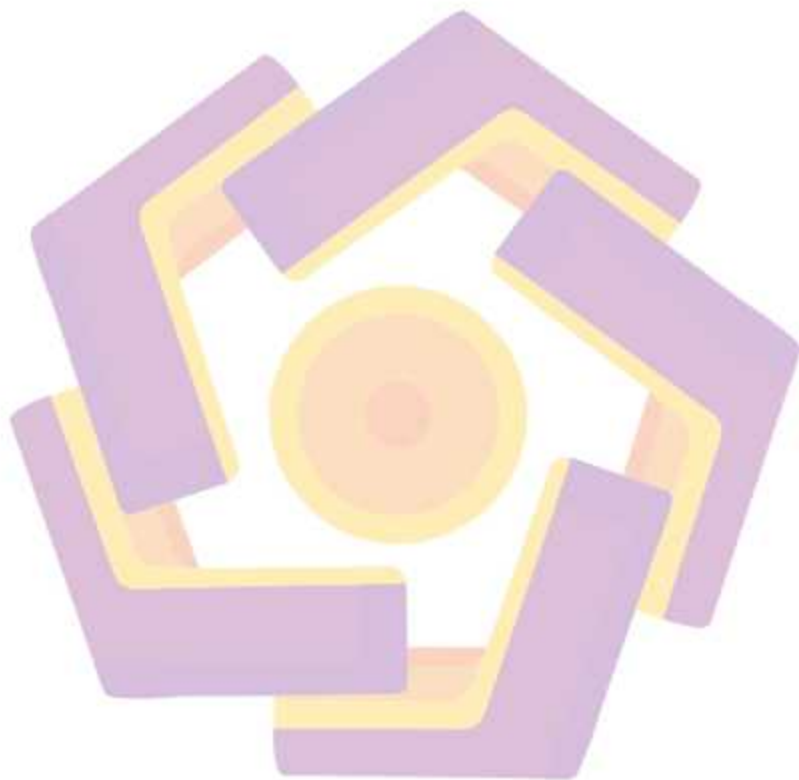
yang relatif tinggi, pendekatan ini bukan merupakan model CNN end-to-end dan memiliki kompleksitas komputasi yang lebih besar, sehingga kurang efisien untuk skenario sistem ringan atau analisis stabilitas model secara murni [8].

Berdasarkan tinjauan terhadap penelitian-penelitian tersebut, dapat disimpulkan bahwa model CNN konvensional maupun arsitektur ringan telah menunjukkan kemampuan dalam menangkap pola visual kesegaran ikan berbasis citra mata. Namun, sebagian besar penelitian masih memiliki keterbatasan utama, yaitu ketergantungan pada augmentasi data, tidak adanya optimasi hiperparameter yang sistematis, serta minimnya evaluasi stabilitas dan kemampuan generalisasi model terhadap distribusi data asli.

Oleh karena itu, penelitian ini mengusulkan penggunaan EfficientNet-B0 sebagai model klasifikasi utama untuk mengidentifikasi tingkat kesegaran ikan berbasis citra mata. EfficientNet-B0 dipilih karena kemampuannya mencapai keseimbangan optimal antara akurasi dan efisiensi komputasi melalui prinsip *compound scaling*. Berbeda dengan penelitian sebelumnya yang menggunakan konfigurasi pelatihan tetap, penelitian ini menekankan optimasi hiperparameter otomatis menggunakan Optuna dengan pendekatan *Tree-structured Parzen Estimator (TPE)* untuk mengevaluasi berbagai kombinasi parameter pelatihan secara sistematis.

Selain itu, penelitian ini secara sengaja tidak menerapkan augmentasi data maupun teknik penyeimbangan sintesis, sehingga performa model sepenuhnya merefleksikan kemampuan generalisasi terhadap distribusi asli dataset. Evaluasi performa tidak hanya difokuskan pada pencapaian akurasi, tetapi juga pada

stabilitas pelatihan dan konsistensi performa, yang diukur melalui selisih performa antara data latih dan data validasi. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi ilmiah berupa pendekatan klasifikasi kesegaran ikan berbasis citra mata yang lebih stabil, efisien, dan dapat direproduksi.



## 2.2 Keaslian Penelitian

Tabel 2.1 Matriks literatur review dan posisi penelitian

Klasifikasi Kesegaran Ikan Berdasarkan Citra Mata Menggunakan Efficientnet-B0 Dengan Optimasi Hyperparameter Optuna

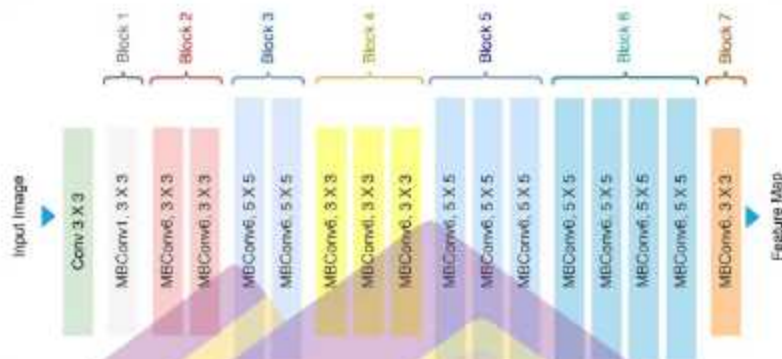
No	Judul Penelitian	Peneliti & Tahun	Metode	Hasil Utama	Keunggulan & Kelemahan	Perbandingan dengan Penelitian Ini
1	Design and Implementation of Fish Freshness Detection Algorithm Using Deep Learning	Anas et al., 2021[3]	Tiny YOLOv2 (object detection)	Akurasi 57,5%	(+) Non-destruktif, cepat. (-) Akurasi rendah, dataset terbatas, bukan klasifikasi murni.	Penelitian ini menggunakan klasifikasi murni berbasis EfficientNet-B0 dengan optimasi hiperparameter dan evaluasi stabilitas, sehingga lebih sesuai untuk analisis generalisasi model.
2	Combining MobileNetV1 and DSC Bottleneck with Expansion	Prasetyo et al., 2022[6]	MobileNetV1 MB-BE	Akurasi ±63% (FFE dataset)	(+) Arsitektur ringan. (-) Akurasi moderat, tidak mengevaluasi stabilitas dan tuning hiperparameter.	Berbeda dengan fokus arsitektur, penelitian ini mempertahankan arsitektur standar dan menitikberatkan pada optimasi hiperparameter serta stabilitas performa.
3	Fish Freshness Evaluation Based on MobileNetV1 and Attention Mechanism	Guo et al., 2022[7]	MobileNetV1 + Attention	Akurasi validasi ±73%	(+) Attention meningkatkan akurasi. (-) Bergantung augmentasi, stabilitas tidak dianalisis.	Penelitian ini mengevaluasi generalisasi model tanpa augmentasi dan menilai konsistensi performa antar pelatihan.
4	Fish-eye Freshness Detection Using DL and ML Methods	Yildiz et al., 2024[8]	VGG19 + ANN	Akurasi 77,3%	(+) Akurasi tinggi. (-) Bukan CNN end-to-end, kompleksitas tinggi.	Penelitian ini menggunakan pendekatan end-to-end EfficientNet-B0 yang lebih ringan dan mudah direproduksi.
5	Image-Based Fish Freshness Classification Using Two-Phase Transfer Learning with Fusion Model	Helmad et al., 2025[12]	EfficientNetB0 + InceptionV3 (Fusion)	Akurasi >93%	(+) Performa sangat tinggi. (-) Model berat, butuh komputasi besar.	Berbeda dari fusion kompleks, penelitian ini fokus pada arsitektur tunggal yang efisien dan stabil tanpa augmentasi.
6	Leveraging Feature Fusion of Image Features and Laser Reflectance	Bahm et al., 2025[13]	CNN + Laser Reflectance	Akurasi ±88%	(+) Multimodal, objektif. (-) Perlu sensor tambahan, sistem kompleks.	Penelitian ini hanya menggunakan citra mata ikan sehingga lebih sederhana dan praktis untuk implementasi.

## 2.3 Landasan Teori

### 2.3.1 EfficientNet

EfficientNet merupakan salah satu arsitektur Convolutional Neural Network (CNN) yang diperkenalkan oleh [9] dengan tujuan meningkatkan efisiensi dan kinerja model secara bersamaan. Berbeda dengan pendekatan konvensional yang hanya memperbesar kedalaman (depth) atau lebar (width) jaringan secara terpisah, EfficientNet mengusulkan sebuah strategi penskalaan terpadu yang dikenal sebagai *compound scaling*. Pendekatan ini memungkinkan peningkatan performa model secara lebih sistematis dengan tetap menjaga efisiensi komputasi.

Pada arsitektur CNN tradisional, peningkatan akurasi umumnya dicapai dengan memperdalam jaringan atau menambah jumlah filter pada setiap lapisan. Namun, strategi tersebut sering kali menghasilkan pertumbuhan parameter yang tidak seimbang serta kebutuhan sumber daya komputasi yang tinggi. EfficientNet mengatasi permasalahan ini dengan menskalakan tiga dimensi utama jaringan secara bersamaan, yaitu kedalaman jaringan, lebar jaringan, dan resolusi input, menggunakan satu koefisien skala yang terkontrol. Dengan demikian, model dapat belajar representasi fitur yang lebih kaya tanpa pemborosan parameter.



Gambar 2.1 Arsitektur EfficientNet

Seperti yang di tampilkan pada Gambar 2.1 arsitektur dasar EfficientNet dibangun menggunakan Mobile Inverted Bottleneck Convolution (MBCConv) yang dilengkapi dengan squeeze-and-excitation (SE) block. MBCConv memungkinkan ekstraksi fitur yang efisien melalui penggunaan depthwise separable convolution, sementara SE block berfungsi untuk meningkatkan sensitivitas model terhadap fitur-fitur penting dengan cara melakukan penyesuaian bobot kanal secara adaptif. Kombinasi kedua komponen ini menjadikan EfficientNet mampu mempertahankan performa tinggi dengan kompleksitas yang relatif rendah.

EfficientNet-B0 merupakan model dasar (baseline) dari keluarga EfficientNet yang diperoleh melalui proses neural architecture search (NAS). Model ini kemudian digunakan sebagai titik awal untuk menghasilkan varian EfficientNet-B1 hingga EfficientNet-B7 melalui penerapan compound scaling. Setiap varian memiliki tingkat kompleksitas dan performa yang berbeda, tergantung pada kebutuhan aplikasi dan keterbatasan sumber daya komputasi.

Tabel 2.2 Perbandingan Varian EfficientNet

Name	Params	Top-1 Acc. (%)	Pretrained
EfficientNet-B0	5.3M	76.3	Ya
EfficientNet-B1	7.8M	78.8	Ya
EfficientNet-B2	9.2M	79.8	Ya
EfficientNet-B3	12M	81.1	Ya
EfficientNet-B4	19M	82.6	Ya
EfficientNet-B5	30M	83.3	Ya
EfficientNet-B6	43M	84.0	Ya
EfficientNet-B7	66M	84.4	Ya

Berdasarkan Tabel 2.2, terlihat bahwa peningkatan indeks varian EfficientNet diikuti oleh penambahan jumlah parameter dan peningkatan akurasi Top-1. Namun, peningkatan akurasi tersebut bersifat gradual, sementara kebutuhan sumber daya komputasi meningkat secara signifikan. Hal ini menunjukkan bahwa varian dengan indeks lebih tinggi tidak selalu memberikan keuntungan yang sebanding terhadap kompleksitas tambahan yang dihasilkan.

EfficientNet-B0 memiliki jumlah parameter paling kecil dibandingkan varian lainnya, namun tetap mampu mencapai tingkat akurasi yang kompetitif. Dengan hanya sekitar 5,3 juta parameter, model ini menunjukkan efisiensi yang tinggi dalam mengekstraksi fitur visual, sehingga banyak digunakan sebagai model dasar dalam berbagai penelitian dan aplikasi yang memerlukan keseimbangan antara performa dan efisiensi komputasi. Selain itu, seluruh varian EfficientNet mendukung penggunaan bobot pretrained, yang memungkinkan pemanfaatan transfer learning untuk mempercepat proses pelatihan dan meningkatkan stabilitas model.

Secara teoretis, EfficientNet menandai pergeseran paradigma dalam perancangan arsitektur CNN, dari pendekatan penskalaan tunggal menuju penskalaan terpadu yang lebih rasional dan terkontrol. Pendekatan ini menjadikan

EfficientNet sebagai salah satu arsitektur CNN yang unggul dalam hal efisiensi parameter, fleksibilitas, dan kemampuan generalisasi, sehingga banyak dijadikan referensi dalam pengembangan sistem klasifikasi citra modern.

### 2.3.2 Optuna dan Algoritma TPE untuk Optimasi Hiperparameter

Optuna adalah kerangka kerja otomatis untuk optimasi hiperparameter yang menggunakan pendekatan Bayesian optimization dengan algoritma Tree-structured Parzen Estimator (TPE) sebagai intinya. Optuna secara adaptif menyesuaikan strategi pemilihan hiperparameter berdasarkan hasil percobaan sebelumnya, sehingga lebih efisien daripada pencarian manual atau grid search tradisional [14]. Dalam algoritma TPE, proses optimasi memodelkan fungsi objektif (misal akurasi validasi) sebagai distribusi probabilitas dan memisahkannya menjadi dua estimasi distribusi: satu untuk kumpulan hiperparameter “baik” (misalnya yang memberi nilai akurasi tinggi) dan satu untuk “buruk” (akurasi rendah) [15]. Dengan estimasi densitas menggunakan Parzen estimator, Optuna akan menghasilkan kandidat hiperparameter baru yang memiliki probabilitas tinggi berada dalam distribusi “baik” dan probabilitas rendah dalam distribusi “buruk”. Berkat pendekatan ini, Optuna dapat mengeksplorasi ruang hiperparameter secara cerdas dan fokus pada area yang menjanjikan tanpa menjelajahi kombinasi yang tidak produktif [15]. Optuna juga mendukung pengecekan interdependensi antar hiperparameter, yang berarti algoritma TPE yang diperbarui dapat mempertimbangkan pengaruh kombinasi parameter secara simultan, bukan secara terpisah [15]. Dalam konteks deep learning, Optuna telah berhasil digunakan untuk menyesuaikan parameter seperti learning rate, batch size, dan arsitektur jaringan demi meningkatkan

performa model[15]. Dengan Optuna, proses pencarian konfigurasi optimal menjadi lebih otomatis, sistematis, dan hemat sumber daya, menghindari trial-and-error manual yang memakan banyak waktu dan komputasi[16].

### **2.3.3 Optimizer dalam Pembelajaran Jaringan Saraf**

Optimizer merupakan komponen penting dalam proses pelatihan jaringan saraf tiruan yang berfungsi untuk memperbarui bobot model secara iteratif berdasarkan nilai gradien dari fungsi kerugian (loss function). Tujuan utama optimizer adalah meminimalkan fungsi kerugian dengan cara menentukan arah dan besar langkah pembaruan parameter yang paling efektif. Pemilihan optimizer yang tepat berpengaruh langsung terhadap kecepatan konvergensi, stabilitas pelatihan, serta kemampuan generalisasi model.

Dalam konteks deep learning, berbagai jenis optimizer telah dikembangkan untuk mengatasi keterbatasan metode optimasi konvensional, terutama yang berkaitan dengan gradien yang tidak stabil, kecepatan konvergensi yang lambat, serta sensitivitas terhadap nilai learning rate. Beberapa optimizer yang paling umum digunakan dan menjadi dasar dalam banyak penelitian adalah Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam), dan AdamW.

#### **2.3.3.1. Stochastic Gradient Descent (SGD)**

Stochastic Gradient Descent (SGD) merupakan salah satu metode optimasi paling dasar dan paling luas digunakan dalam pelatihan jaringan saraf. Berbeda dengan batch gradient descent yang menghitung gradien menggunakan seluruh data pelatihan, SGD memperbarui bobot model berdasarkan satu sampel atau satu mini-batch data pada setiap iterasi. Pendekatan ini membuat SGD lebih efisien secara

komputasi dan memungkinkan pembelajaran berlangsung pada dataset berukuran besar. Secara matematis, pembaruan bobot pada SGD dapat dirumuskan sebagai berikut:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) \quad (2.1)$$

Di mana  $\theta$  merepresentasikan parameter model,  $\eta$  adalah learning rate, dan  $\nabla L(\theta_t)$  merupakan gradien fungsi kerugian terhadap parameter pada iterasi ke- $t$

Keunggulan utama SGD terletak pada kesederhanaannya dan kemampuannya dalam mencapai solusi yang memiliki kemampuan generalisasi yang baik. Namun, SGD memiliki beberapa kelemahan, seperti sensitivitas tinggi terhadap pemilihan learning rate serta kecenderungan untuk mengalami osilasi pada permukaan loss yang tidak rata. Untuk mengatasi hal tersebut, SGD sering dikombinasikan dengan mekanisme tambahan seperti momentum, yang membantu mempercepat konvergensi dan mengurangi fluktuasi gradien.

### 2.3.3.2. Adaptive Moment Estimation (Adam)

Adaptive Moment Estimation (Adam) merupakan optimizer adaptif yang menggabungkan keunggulan dari dua metode optimasi sebelumnya, yaitu Momentum dan RMSProp. Adam menyimpan dua momen statistik dari gradien, yaitu momen pertama (rata-rata gradien) dan momen kedua (rata-rata kuadrat gradien), yang digunakan untuk menyesuaikan learning rate secara adaptif pada setiap parameter.

Pembaruan parameter pada Adam dirumuskan sebagai:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.2)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Di mana  $g_t$  adalah gradien pada iterasi ke- $t$   $\beta_1$  dan  $\beta_2$  merupakan koefisien peluruhan (decay rates), dan  $\epsilon$  adalah konstanta kecil untuk menjaga stabilitas numerik.

Keunggulan Adam terletak pada kemampuannya untuk beradaptasi secara otomatis terhadap skala gradien yang berbeda pada setiap parameter, sehingga sering kali menghasilkan konvergensi yang lebih cepat dibandingkan SGD. Adam juga relatif robust terhadap pemilihan learning rate awal. Namun, beberapa studi menunjukkan bahwa Adam dapat menghasilkan solusi yang kurang optimal dari sisi generalisasi jika dibandingkan dengan SGD, terutama pada tahap pelatihan jangka panjang.

### 2.3.3.2. AdamW

AdamW merupakan varian dari Adam yang diperkenalkan untuk mengatasi kelemahan utama Adam terkait penerapan weight decay. Pada Adam konvensional, weight decay secara implisit digabungkan dengan mekanisme pembaruan gradien, sehingga efek regularisasi menjadi tidak konsisten. AdamW memisahkan secara eksplisit antara pembaruan gradien dan weight decay, sehingga regularisasi dapat diterapkan secara lebih efektif dan terkontrol.

Secara konseptual, AdamW mempertahankan mekanisme adaptif Adam, tetapi memperbarui parameter dengan menambahkan penalti weight decay secara langsung pada bobot model. Pendekatan ini terbukti mampu meningkatkan

stabilitas pelatihan dan memperbaiki kemampuan generalisasi model, terutama pada jaringan saraf dalam yang memiliki jumlah parameter besar.

Keunggulan AdamW dibandingkan Adam terletak pada keseimbangan antara kecepatan konvergensi dan kemampuan generalisasi. Oleh karena itu, AdamW banyak digunakan dalam pelatihan model modern, khususnya pada arsitektur CNN dan transformer, yang memerlukan regularisasi yang lebih eksplisit untuk mencegah overfitting.

#### **2.3.4 Weight Decay dalam Pembelajaran Jaringan Saraf**

Weight decay merupakan salah satu teknik regularisasi yang umum digunakan dalam pelatihan jaringan saraf tiruan untuk mengurangi risiko overfitting. Prinsip dasar weight decay adalah dengan menambahkan penalti terhadap besarnya nilai bobot model ke dalam fungsi kerugian, sehingga proses pelatihan tidak hanya berfokus pada minimisasi kesalahan prediksi, tetapi juga pada pembatasan kompleksitas model. Dengan demikian, model didorong untuk mempelajari representasi yang lebih sederhana dan stabil.

Secara konseptual, weight decay bekerja dengan mengendalikan pertumbuhan nilai bobot selama proses pembaruan parameter. Bobot dengan nilai yang terlalu besar cenderung membuat model menjadi terlalu sensitif terhadap fluktuasi kecil pada data, yang dapat berdampak pada penurunan kemampuan generalisasi. Dengan menerapkan penalti terhadap bobot tersebut, weight decay membantu model untuk menghindari pembelajaran pola yang terlalu spesifik terhadap data pelatihan.

Secara matematis, weight decay umumnya diimplementasikan dalam bentuk regularisasi L2, di mana fungsi kerugian dimodifikasi dengan menambahkan komponen penalti sebagai berikut:

$$L_{total} = L_{data} + \lambda \sum_i w_i^2 \quad (2.3)$$

Di mana  $L_{data}$  adalah fungsi kerugian utama,  $w_i$  merepresentasikan bobot model, dan  $\lambda$  merupakan koefisien weight decay yang mengontrol kekuatan regularisasi. Nilai  $\lambda$  yang lebih besar akan memberikan penalti yang lebih kuat terhadap bobot, sehingga mendorong model untuk memiliki bobot yang lebih kecil.

Dalam praktik optimasi, weight decay memengaruhi proses pembaruan bobot dengan cara mengecilkan nilai bobot secara bertahap pada setiap iterasi. Efek ini membantu menjaga stabilitas pelatihan dan mengurangi kecenderungan model untuk menyesuaikan diri secara berlebihan terhadap data pelatihan. Namun, pemilihan nilai weight decay yang tidak tepat dapat berdampak negatif. Nilai yang terlalu besar dapat menyebabkan underfitting, sedangkan nilai yang terlalu kecil mungkin tidak cukup efektif dalam mencegah overfitting.

Perlu dicatat bahwa implementasi weight decay dapat berbeda tergantung pada jenis optimizer yang digunakan. Pada beberapa optimizer adaptif, seperti Adam, weight decay sering kali digabungkan secara implisit dengan mekanisme pembaruan gradien, sehingga efek regularisasi menjadi kurang optimal. Untuk mengatasi hal tersebut, pendekatan decoupled weight decay diperkenalkan, di mana regularisasi bobot dipisahkan dari pembaruan gradien. Pendekatan ini memungkinkan kontrol yang lebih jelas terhadap efek regularisasi dan terbukti meningkatkan stabilitas serta kemampuan generalisasi model.

Secara teoretis, weight decay berperan sebagai mekanisme pengendalian kompleksitas model yang efektif dan sederhana. Dengan membatasi besarnya bobot, model cenderung mempelajari pola yang lebih umum dan robust terhadap variasi data. Oleh karena itu, weight decay banyak digunakan dalam pelatihan model deep learning modern sebagai bagian dari strategi regularisasi untuk menghasilkan model yang memiliki performa stabil dan kemampuan generalisasi yang baik.

### **2.3.5 Dropout dalam Jaringan Saraf Tiruan**

Dropout merupakan salah satu teknik regularisasi yang diperkenalkan untuk mengurangi risiko overfitting pada jaringan saraf tiruan, khususnya pada arsitektur deep learning dengan jumlah parameter yang besar. Konsep dropout pertama kali diperkenalkan oleh [17] sebagai metode sederhana namun efektif untuk meningkatkan kemampuan generalisasi model dengan cara mengurangi ketergantungan berlebihan antar neuron selama proses pelatihan.

Prinsip dasar dropout adalah dengan menonaktifkan (dropping) sejumlah neuron secara acak pada setiap iterasi pelatihan. Neuron yang dinonaktifkan tidak berkontribusi dalam proses forward pass maupun backpropagation. Dengan demikian, jaringan tidak dapat bergantung pada neuron tertentu secara konsisten dan dipaksa untuk mempelajari representasi fitur yang lebih tersebar dan robust. Proses ini dapat dianalogikan sebagai pelatihan sejumlah sub-jaringan (thinned networks) yang berbagi bobot yang sama.

Dropout rate merupakan parameter yang menentukan proporsi neuron yang dinonaktifkan selama pelatihan. Sebagai contoh, dropout rate sebesar 0,5 berarti

bahwa 50% neuron pada suatu lapisan akan dinonaktifkan secara acak pada setiap iterasi. Pemilihan nilai dropout rate memiliki pengaruh signifikan terhadap perilaku pembelajaran model. Nilai dropout yang terlalu kecil mungkin tidak cukup efektif dalam mencegah overfitting, sedangkan nilai yang terlalu besar dapat menghambat proses pembelajaran dan menyebabkan underfitting.

Secara matematis, dropout dapat dipandang sebagai proses pengalihan aktivasi neuron menggunakan variabel acak Bernoulli. Setiap neuron memiliki probabilitas tertentu untuk dipertahankan atau dinonaktifkan selama pelatihan. Pada fase inferensi, dropout tidak lagi diterapkan, dan bobot jaringan biasanya diskalakan untuk memastikan bahwa ekspektasi aktivasi neuron tetap konsisten dengan fase pelatihan. Pendekatan ini memungkinkan model mempertahankan stabilitas prediksi pada saat pengujian.

Keunggulan utama dropout terletak pada kemampuannya untuk mengurangi co-adaptation antar neuron, yaitu kondisi di mana neuron-neuron tertentu hanya efektif ketika bekerja bersama neuron lain secara spesifik. Dengan menghilangkan ketergantungan tersebut, dropout mendorong jaringan untuk mempelajari fitur yang lebih umum dan tidak terlalu spesifik terhadap data pelatihan. Hal ini berkontribusi langsung terhadap peningkatan kemampuan generalisasi model.

Meskipun dropout sangat efektif sebagai teknik regularisasi, penerapannya perlu disesuaikan dengan jenis arsitektur jaringan yang digunakan. Pada jaringan konvolusional, dropout umumnya diterapkan pada lapisan fully connected atau setelah proses ekstraksi fitur utama, karena penerapan dropout secara agresif pada lapisan konvolusi awal dapat mengganggu pembelajaran pola spasial dasar. Selain

itu, interaksi dropout dengan teknik regularisasi lain, seperti weight decay, perlu dipertimbangkan agar tidak menghasilkan regularisasi berlebihan.

Secara teoretis, dropout merupakan metode regularisasi yang sederhana namun kuat dalam meningkatkan stabilitas dan kemampuan generalisasi jaringan saraf tiruan. Dengan mengontrol tingkat dropout secara tepat, model dapat menghindari pembelajaran pola yang terlalu spesifik terhadap data pelatihan, sekaligus mempertahankan kapasitas representasi yang memadai untuk menangkap pola penting dalam data.

### 2.3.6 Metrik Evaluasi Kinerja Model Klasifikasi

Untuk menilai performa model klasifikasi citra, digunakan beberapa metrik evaluasi yang dihitung berdasarkan confusion matrix. Confusion matrix adalah tabel kontingensi yang merangkum hasil prediksi model terhadap data uji dengan membandingkan prediksi vs. kelas aktual untuk setiap kategori [18]. Pada kasus klasifikasi biner, confusion matrix berukuran  $2 \times 2$  berisi:

- True Positive (TP) – prediksi positif benar,
- True Negative (TN) – prediksi negatif benar,
- False Positive (FP) – prediksi positif keliru, dan
- False Negative (FN) – prediksi negatif keliru.

Untuk klasifikasi multi-kelas, matriks ini meluas menjadi  $N \times N$  (dengan  $N$  jumlah kelas) dan tiap entri  $(i, j)$  merepresentasikan jumlah sampel kelas actual  $i$  yang diprediksi sebagai kelas  $j$  [18].

Dari confusion matrix inilah berbagai metrik kinerja diturunkan:

- Akurasi (Accuracy)

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.4)$$

Akurasi dengan persamaan (2.4) menggambarkan proporsi sampel yang diklasifikasikan dengan tepat oleh model secara keseluruhan.

- Presisi (Precision)

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2.5)$$

Presisi dengan persamaan 2.2 mengukur ketepatan model dalam memprediksi kelas positif—berapa banyak dari semua prediksi “positif” yang benar-benar positif (Munshi et al., 2025). Nilai presisi yang tinggi berarti sedikit prediksi positif yang keliru, penting dalam konteks di mana kesalahan positif harus diminimalisir (misalnya deteksi penyakit).

- Recall (Sensitivitas / True Positive Rate)

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.6)$$

Recall yang di tunjukan pada persamaan (2.6) mengukur tingkat keberhasilan model dalam menangkap semua sampel yang seharusnya positif[18]. Recall tinggi berarti model mampu menemukan sebagian besar instance positif, sehingga berguna untuk kasus di mana miss (FN) harus dikurangi[19].

- F1-Score

$$F1 = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (2.7)$$

F1-score merupakan rata-rata harmonis antara presisi dan recall, menggabungkan keduanya dalam satu nilai yang seimbang[18]. Nilai ini tinggi hanya jika kedua metrik sama-sama tinggi; jika salah satu menurun, F1 juga menurun drastis.

Dengan demikian, F1-score sangat cocok digunakan saat distribusi kelas tidak merata, karena akurasi saja bisa menyesatkan (contohnya, model yang selalu memprediksi kelas mayoritas mungkin akurasinya tinggi tetapi recall untuk kelas minoritas = 0).

Selain keempat metrik utama tersebut, laporan hasil klasifikasi biasanya juga menyajikan confusion matrix utuh untuk memberikan gambaran lebih detail terhadap kesalahan per kelas, serta metrik turunan lain seperti AUC-ROC atau kapasitas generalisasi tergantung kebutuhan. Dalam penelitian ini, akurasi, presisi, recall, dan F1-score digunakan sebagai indikator numerik utama kinerja model, sedangkan confusion matrix digunakan untuk analisis kesalahan tiap kelas secara lebih mendalam.



## **BAB 3**

### **METODE PENELITIAN**

#### **3.1 Jenis, Sifat, dan Pendekatan Penelitian**

Penelitian ini merupakan penelitian eksperimental dengan pendekatan kuantitatif, yang bertujuan untuk mengevaluasi kinerja model deep learning berbasis EfficientNet-B0 dalam mengklasifikasikan tingkat kesegaran ikan berdasarkan citra mata ikan. Penelitian ini bersifat terapan (applied research) karena hasil yang diperoleh diarahkan untuk mendukung pengembangan sistem penilaian kesegaran ikan secara otomatis yang dapat diimplementasikan pada konteks industri perikanan dan pengolahan hasil laut.

Pendekatan eksperimental dipilih karena penelitian ini menitikberatkan pada pengujian langsung performa model melalui eksplorasi dan evaluasi kombinasi hyperparameter yang berbeda dalam skenario pelatihan terkontrol. Seluruh proses penelitian dilakukan secara komputasi, mencakup tahap pemodelan, optimasi parameter, hingga evaluasi kinerja model menggunakan metrik kuantitatif. Tidak dilakukan intervensi berupa augmentasi data, sehingga performa model sepenuhnya merefleksikan kemampuan generalisasi terhadap distribusi asli dataset.

Proses pelatihan dan evaluasi model dilakukan menggunakan lingkungan pemrograman berbasis Python, dengan memanfaatkan pustaka deep learning yang mendukung arsitektur EfficientNet serta integrasi Optuna sebagai kerangka optimasi hyperparameter berbasis Bayesian optimization. Pendekatan ini

memungkinkan pencarian konfigurasi parameter yang optimal secara sistematis dan efisien, sekaligus memberikan dasar kuantitatif yang kuat dalam membandingkan stabilitas dan konsistensi performa antar percobaan.

Secara keseluruhan, metodologi yang digunakan dalam penelitian ini dirancang untuk menghasilkan evaluasi yang objektif, terukur, dan dapat direproduksi terhadap kemampuan model EfficientNet-B0 dalam melakukan klasifikasi tingkat kesegaran ikan berbasis citra mata, dengan fokus utama pada keseimbangan antara performa akurasi dan kemampuan generalisasi model.

### **3.2 Metode Pengumpulan Data**

Data yang digunakan dalam penelitian ini berasal dari platform Mandeley Data dengan judul *The Freshness of the Fish Eyes* [20]. Dataset ini berisi citra mata ikan dari delapan spesies berbeda, yaitu *Chanos chanos*, *Eleutheronema tetradactylum*, *Johnius trachycephalus*, *Nibea albiflora*, *Oreochromis mossambicus*, *Oreochromis niloticus*, *Rastrelliger faughni*, dan *Upeneus moluccensis*.

Setiap spesies ikan memiliki tiga kategori tingkat kesegaran, yaitu Highly Fresh, Fresh, dan Not Fresh. Kategori tersebut ditentukan berdasarkan kondisi visual pada bagian mata ikan, seperti tingkat kejernihan kornea, perubahan warna, dan tingkat kekeruhan pupil. Kombinasi antara delapan spesies dan tiga kategori tingkat kesegaran menghasilkan 24 kelas data.

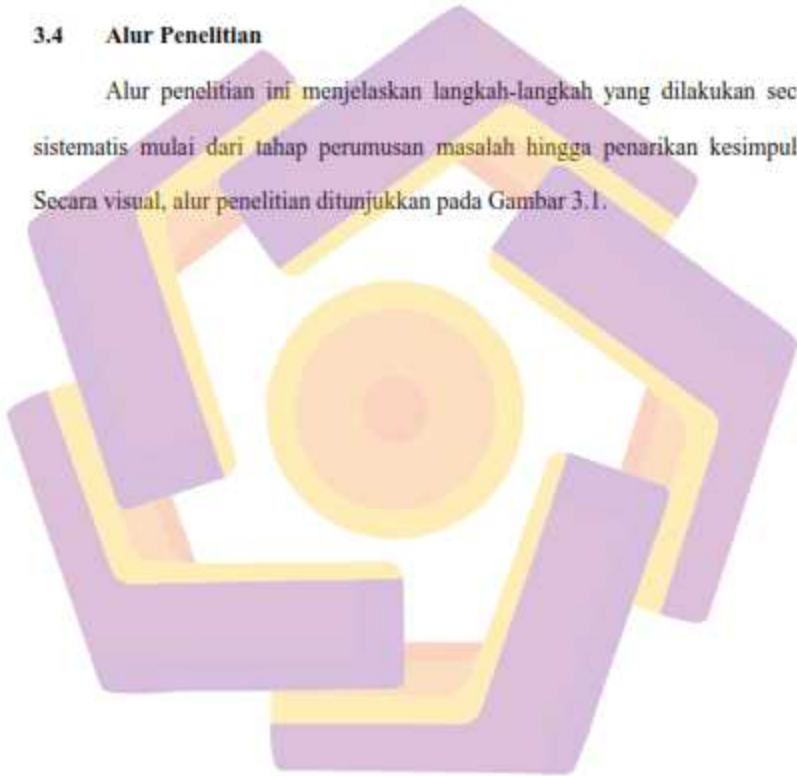
### **3.3 Metode Analisis Data**

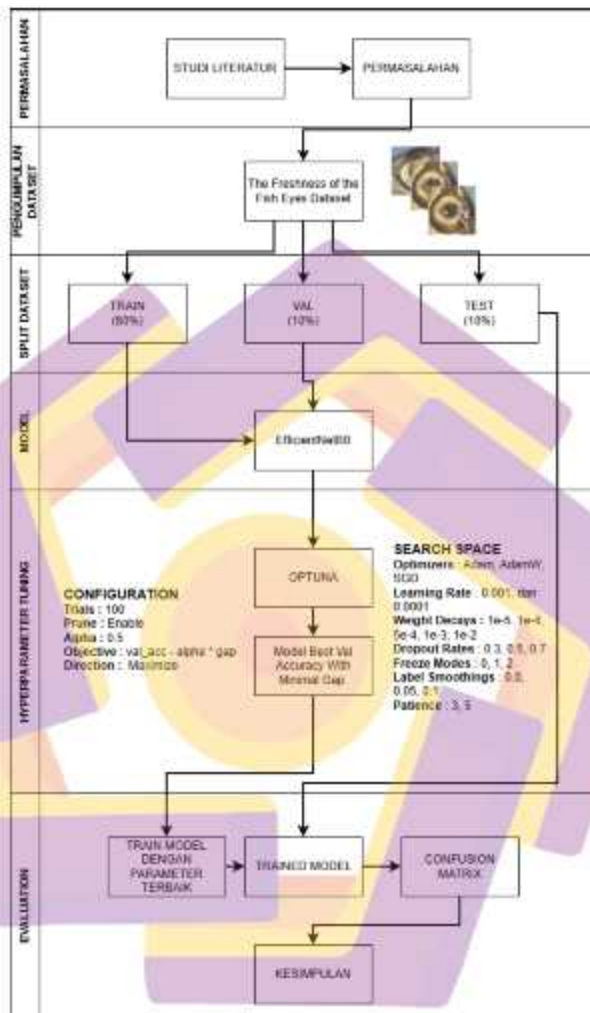
Bagian ini menjelaskan metode yang akan digunakan untuk menganalisis dataset yang telah dikumpulkan. Analisis data akan dilakukan untuk memahami

karakteristik citra, distribusi antar kelas, serta potensi permasalahan yang mungkin memengaruhi kinerja model pada tahap pelatihan. Hasil dari proses ini akan menjadi dasar dalam menentukan strategi pra-pemrosesan dan pendekatan klasifikasi yang sesuai.

### **3.4 Alur Penelitian**

Alur penelitian ini menjelaskan langkah-langkah yang dilakukan secara sistematis mulai dari tahap perumusan masalah hingga penarikan kesimpulan. Secara visual, alur penelitian ditunjukkan pada Gambar 3.1.





Gambar 3.1 Alur Penelitian

Pada Gambar 3.1 ditunjukkan secara menyeluruh alur penelitian yang dilakukan dalam studi ini, yang dimulai dari tahap identifikasi permasalahan hingga penarikan kesimpulan akhir. Proses penelitian diawali dengan identifikasi masalah

terkait kesulitan dalam mengklasifikasikan tingkat kesegaran ikan secara objektif hanya berdasarkan pengamatan visual manusia, khususnya melalui citra mata ikan. Selanjutnya, dilakukan studi literatur untuk mengkaji pendekatan-pendekatan yang telah digunakan pada penelitian sebelumnya, baik dari sisi dataset, model deep learning, maupun metode evaluasi yang diterapkan.

Setelah tahap konseptual tersebut, dilakukan pengumpulan dataset citra mata ikan yang kemudian dibagi ke dalam tiga subset utama, yaitu data latih, data validasi, dan data uji. Pembagian ini bertujuan untuk memastikan proses pelatihan, pemilihan model, dan evaluasi dilakukan secara terpisah dan objektif. Pada penelitian ini, tidak dilakukan proses augmentasi data, sehingga seluruh citra yang digunakan merepresentasikan kondisi asli dataset. Pendekatan ini dipilih untuk menilai kemampuan model dalam melakukan generalisasi secara langsung terhadap variasi alami pada data tanpa intervensi buatan.

Data latih dan validasi selanjutnya digunakan dalam proses pemodelan dengan arsitektur deep learning EfficientNet-B0, yang dipilih karena keseimbangan antara kompleksitas model dan efisiensi komputasi. Untuk memperoleh konfigurasi model yang optimal, dilakukan proses tuning hyperparameter menggunakan Optuna dengan pendekatan Bayesian optimization. Proses ini mengeksplorasi berbagai kombinasi parameter penting seperti optimizer, learning rate, weight decay, dropout rate, label smoothing, dan patience, dengan kriteria pemilihan model terbaik didasarkan pada nilai Gap Penalty Score guna menyeimbangkan performa pelatihan dan kemampuan generalisasi.

Model terbaik hasil proses optimasi kemudian dilatih ulang dan dievaluasi menggunakan data uji. Evaluasi performa dilakukan secara komprehensif melalui metrik akurasi, precision, recall, serta analisis confusion matrix untuk mengidentifikasi pola keberhasilan dan kesalahan klasifikasi antar kelas kesegaran dan spesies ikan. Selain itu, analisis tambahan seperti riwayat pelatihan (training history) dan interpretasi prediksi juga digunakan untuk memperkuat pemahaman terhadap perilaku model.

Secara keseluruhan, alur penelitian pada Gambar 3.1 menggambarkan proses yang sistematis dan terstruktur, mulai dari perumusan masalah, pemodelan berbasis deep learning yang dioptimalkan secara adaptif, hingga evaluasi menyeluruh terhadap kinerja model. Penjelasan lebih rinci mengenai setiap tahapan dalam alur penelitian ini akan dibahas pada subbagian-subbagian selanjutnya.

#### **3.4.1 Perumusan Masalah dan Studi Literatur**

Tahap pertama dalam penelitian ini dimulai dengan perumusan masalah, yaitu bagaimana menentukan tingkat kesegaran ikan secara objektif melalui analisis citra mata ikan menggunakan metode deep learning. Permasalahan ini muncul karena metode tradisional dalam menilai kesegaran ikan masih bersifat subjektif dan memerlukan tenaga ahli berpengalaman. Oleh karena itu, penelitian ini berupaya menghadirkan sistem klasifikasi berbasis citra yang mampu memberikan hasil yang lebih objektif, cepat, dan konsisten.

Setelah permasalahan dirumuskan, dilakukan studi literatur terhadap berbagai penelitian terdahulu yang relevan dengan topik ini. Kajian pustaka meliputi teori pengolahan citra digital, metode klasifikasi berbasis deep learning,

serta arsitektur jaringan EfficientNet sebagai model utama dalam penelitian ini. Selain itu, diperiksa juga penelitian terkait strategi optimasi model, termasuk hyperparameter tuning menggunakan Optuna, yang menjadi komponen penting dalam meningkatkan performa model meskipun tanpa penggunaan teknik augmentasi atau oversampling tambahan.

### **3.4.2 Pengumpulan Dataset**

Dataset yang digunakan dalam penelitian ini adalah The Freshness of the Fish Eyes Dataset, yang berisi kumpulan citra mata ikan dengan berbagai tingkat kesegaran. Dataset ini berperan sebagai data utama untuk proses pelatihan, validasi, dan pengujian model klasifikasi.

Citra-citra pada dataset dikategorikan berdasarkan tingkat kesegaran ikan, seperti segar, agak segar, dan tidak segar. Pengumpulan data ini dilakukan dengan memperhatikan kualitas dan konsistensi citra agar model dapat mengenali pola visual yang relevan dengan karakteristik tingkat kesegaran ikan.

### **3.4.3 Pembagian Dataset**

Setelah dataset terkumpul, dilakukan proses pembagian dataset menjadi tiga bagian yaitu data latih, data validasi, dan data uji. Data latih digunakan dalam proses pelatihan model, data validasi berfungsi untuk menguji performa model selama proses training dan membantu dalam penyetelan parameter, sedangkan data uji digunakan untuk mengukur kemampuan generalisasi model terhadap data baru.

Dalam penelitian ini, pembagian dataset dilakukan dengan proporsi 80% untuk data latih, 10% untuk data validasi, dan 10% untuk data uji. Pembagian ini dilakukan secara acak namun tetap memperhatikan keseimbangan distribusi antar kelas kesegaran.

### 3.4.4 Pemodelan Deep Learning

Tahap ini merupakan inti dari penelitian, yaitu membangun dan mengevaluasi performa model deep learning dalam melakukan klasifikasi tingkat kesegaran ikan. Pada penelitian ini digunakan arsitektur EfficientNet. Pemilihan EfficientNet didasarkan pada efisiensi arsitekturnya yang mampu mencapai keseimbangan optimal antara akurasi dan kompleksitas model melalui teknik compound scaling. Pendekatan ini memungkinkan peningkatan kedalaman, lebar, dan resolusi jaringan secara proporsional sehingga model dapat mengekstraksi fitur visual dengan lebih efektif.

Setiap varian EfficientNet dilatih menggunakan dataset yang telah melalui proses augmentasi, dan performanya dibandingkan untuk menentukan arsitektur yang paling optimal dalam mengklasifikasikan tingkat kesegaran ikan berdasarkan citra mata. Dengan fokus pada satu keluarga arsitektur yang konsisten, penelitian ini memastikan evaluasi yang lebih terkontrol sekaligus memanfaatkan kekuatan EfficientNet dalam menangani variasi visual pada data citra.

### 3.4.5 Tuning Hyperparameter dengan Optuna

Untuk memperoleh hasil pelatihan yang optimal, penelitian ini menggunakan proses hyperparameter tuning berbasis Optuna, yaitu sebuah framework otomatis yang melakukan pencarian kombinasi parameter terbaik berdasarkan performa validasi model. Proses tuning dilakukan secara sistematis dengan ruang pencarian (search space) yang cukup luas, mencakup beberapa aspek penting dalam pelatihan model. Seperti optimizer meliputi Adam, AdamW, SGD. Rentang learning rate yang dievaluasi terdiri dari dua pilihan (0.0001 dan 0.001), disertai variasi weight decay sebanyak lima nilai ( $1e-5$ ,  $1e-4$ ,  $5e-4$ ,  $1e-3$ , dan  $1e-2$ ).

Selain itu, dropout rate diuji pada tiga tingkat (0.3, 0.5, 0.7), sedangkan batch size ditetapkan tetap pada nilai 16 untuk menjaga konsistensi proses pelatihan. Penelitian ini juga menguji tiga mode layer freezing (0, 1, dan 2), tiga nilai label smoothing (0.0, 0.05, 0.1), serta dua nilai early stopping patience (3 dan 5). Melalui eksplorasi ruang pencarian yang komprehensif ini, Optuna menghasilkan konfigurasi hyperparameter terbaik yang kemudian digunakan untuk proses pelatihan final dan evaluasi model.

#### **3.4.6 Evaluasi Hasil Tuning Hyperparameter**

Evaluasi dilakukan terhadap model yang telah melalui proses tuning untuk menilai performanya dalam mengklasifikasikan tingkat kesegaran ikan. Pengujian dilakukan menggunakan data uji yang belum pernah digunakan selama pelatihan, sehingga hasil evaluasi benar-benar mencerminkan kemampuan generalisasi model.

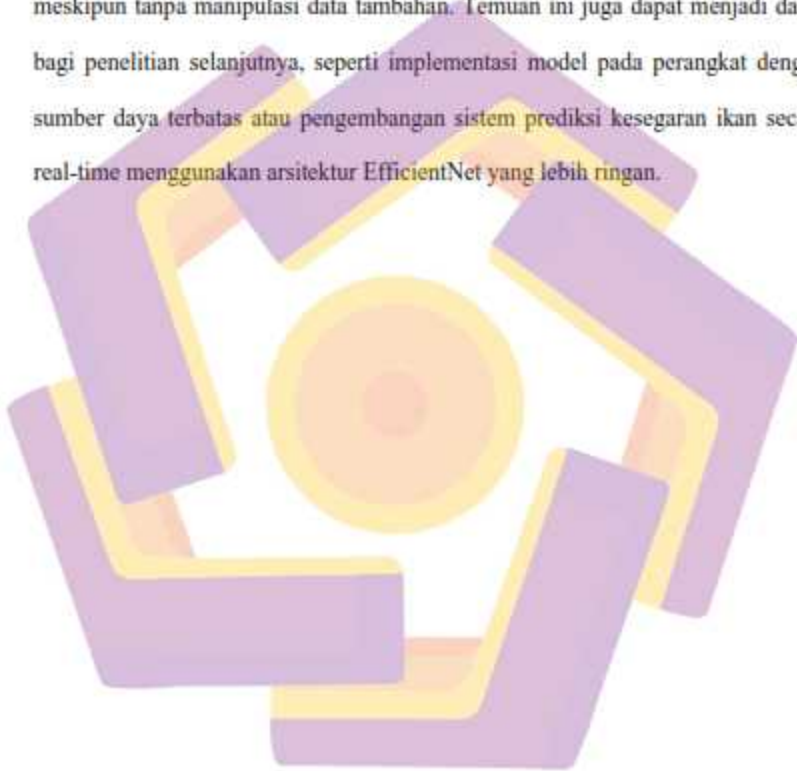
Beberapa metrik yang digunakan dalam evaluasi antara lain akurasi, presisi, recall, dan F1-score. Selain itu, confusion matrix juga digunakan untuk menilai seberapa baik model mampu mengenali setiap kelas kesegaran ikan. Berdasarkan hasil evaluasi tersebut, ditentukan model terbaik yang digunakan sebagai hasil akhir penelitian.

#### **3.4.7 Penarikan Kesimpulan**

Tahap terakhir dalam alur penelitian adalah penarikan kesimpulan. Berdasarkan hasil pengujian dan evaluasi, diperoleh varian EfficientNet terbaik yang mampu melakukan klasifikasi tingkat kesegaran ikan dengan performa paling tinggi. Selain itu, dilakukan analisis terhadap pengaruh proses hyperparameter tuning menggunakan Optuna dalam meningkatkan kinerja model, mengingat

penelitian ini tidak menggunakan teknik augmentasi maupun oversampling tambahan.

Hasil penelitian ini diharapkan dapat berkontribusi pada pengembangan sistem deteksi kesegaran ikan berbasis citra digital yang efisien, akurat, dan stabil meskipun tanpa manipulasi data tambahan. Temuan ini juga dapat menjadi dasar bagi penelitian selanjutnya, seperti implementasi model pada perangkat dengan sumber daya terbatas atau pengembangan sistem prediksi kesegaran ikan secara real-time menggunakan arsitektur EfficientNet yang lebih ringan.



## BAB 4

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1 Hasil Pengumpulan dan Persiapan Dataset

Pada subbab ini, pembahasan difokuskan pada penyajian data dan analisis implikasinya terhadap proses pelatihan model, dengan penekanan pada hubungan antara distribusi dataset dan potensi dampaknya terhadap kinerja model.

Dataset yang digunakan dalam penelitian ini adalah The Freshness of the Fish Eyes Dataset[20] yang diperoleh dari platform Mandeley Data dan terdiri dari 24 kelas, hasil kombinasi 8 spesies ikan dan 3 tingkat kesegaran (Highly Fresh, Fresh, dan Not Fresh). Dataset awal berjumlah 4.200 citra, dan setelah proses data cleaning (penghapusan citra rusak, tidak terbaca, dan duplikat), tersisa 4.050 citra valid atau sekitar 96,4% dari data awal. Angka ini menunjukkan bahwa dataset memiliki kualitas awal yang baik dan layak digunakan untuk analisis lebih lanjut.

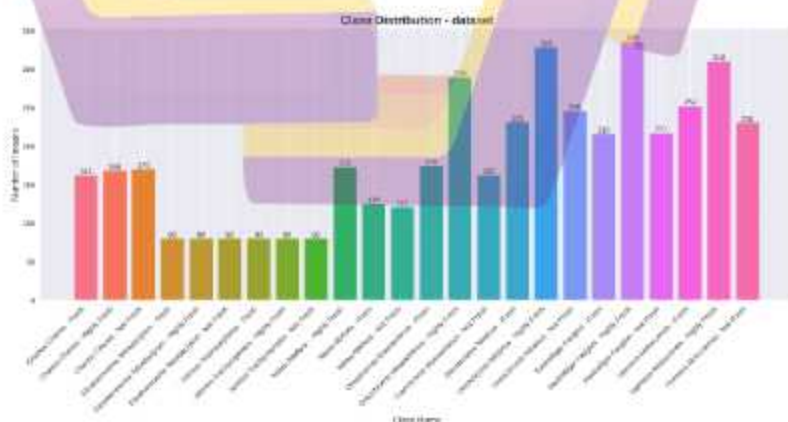
Untuk menggambarkan komposisi dataset secara rinci, distribusi jumlah citra pada setiap kelas disajikan pada Tabel 4.1.

Tabel 4.1 Distribusi Dataset Berdasarkan Kelas

No	Spesies dan Tingkat Kesegaran	Total Citra
1	<i>Chanos chanos</i> – Fresh	162
2	<i>Upeneus moluccensis</i> – Not Fresh	230
3	<i>Johnius trachycephalus</i> – Not Fresh	80
4	<i>Elcutheronema tetradactylum</i> – Highly Fresh	80
5	<i>Chanos chanos</i> – Not Fresh	170
6	<i>Nibea albiflora</i> – Not Fresh	121
7	<i>Upeneus moluccensis</i> – Highly Fresh	310
8	<i>Oreochromis niloticus</i> – Not Fresh	246
9	<i>Nibea albiflora</i> – Fresh	125
10	<i>Upeneus moluccensis</i> – Fresh	252

No	Spesies dan Tingkat Kesegaran	Total Citra
11	Eleutheronema tetradactylum – Not Fresh	80
12	Oreochromis mossambicus – Not Fresh	162
13	Johnius trachycephalus – Highly Fresh	80
14	Oreochromis niloticus – Fresh	231
15	Eleutheronema tetradactylum – Fresh	80
16	Johnius trachycephalus – Fresh	80
17	Oreochromis mossambicus – Highly Fresh	289
18	Oreochromis niloticus – Highly Fresh	328
19	Chanos chanos – Highly Fresh	168
20	Nibea albiflora – Highly Fresh	173
21	Oreochromis mossambicus – Fresh	174
22	Rastrelliger faughni – Highly Fresh	336
23	Rastrelliger faughni – Fresh	216
24	Rastrelliger faughni – Not Fresh	217

Tabel Tabel 4.1 menunjukkan variasi jumlah citra yang cukup besar antar kombinasi spesies dan tingkat kesegaran. Selanjutnya, untuk memudahkan interpretasi visual terhadap ketimpangan tersebut, distribusi kelas divisualisasikan dalam bentuk diagram batang pada Gambar 4.1.



Gambar 4.1 Distribusi Dataset Berdasarkan Kelas

Berdasarkan Tabel 4.1 dan Gambar 4.1, terlihat bahwa jumlah citra per kelas berada pada rentang 80 hingga 336 citra. Kelas dengan jumlah citra tertinggi adalah *Rastrelliger faughni* – Highly Fresh dengan 336 citra, sedangkan beberapa kelas seperti *Johnius trachycephalus* – Fresh dan *Eleutheronema tetradactylum* – Fresh hanya memiliki 80 citra. Perbedaan ini menunjukkan rasio ketimpangan lebih dari 4:1 antara kelas terbesar dan terkecil.

Fenomena ketidakseimbangan ini mengindikasikan adanya class imbalance struktural pada dataset, yang berpotensi memengaruhi proses pembelajaran model, khususnya dalam kecenderungan bias terhadap kelas dengan jumlah sampel lebih besar [21]. Namun, penelitian ini secara sengaja tidak menerapkan teknik balancing, oversampling, maupun undersampling.

Keputusan tersebut diambil dengan pertimbangan bahwa distribusi alami dataset perlu dipertahankan agar performa model mencerminkan kondisi data nyata [22], [23]. Selain itu, pendekatan ini memungkinkan evaluasi yang lebih objektif terhadap kemampuan model EfficientNet dalam menangani ketidakseimbangan kelas tanpa intervensi tambahan. Dengan demikian, hasil yang diperoleh tidak hanya merepresentasikan akurasi numerik, tetapi juga mencerminkan robustness model terhadap variasi distribusi data.

## 4.2 Pembagian Dataset

Setelah tahap persiapan data selesai, dataset dibagi menjadi tiga subset utama, yaitu 80% data latih, 10% data validasi, dan 10% data uji. Pembagian data dilakukan secara acak dengan pendekatan stratified splitting untuk memastikan

bahwa proporsi jumlah citra pada setiap kelas tetap terjaga pada masing-masing subset [24].

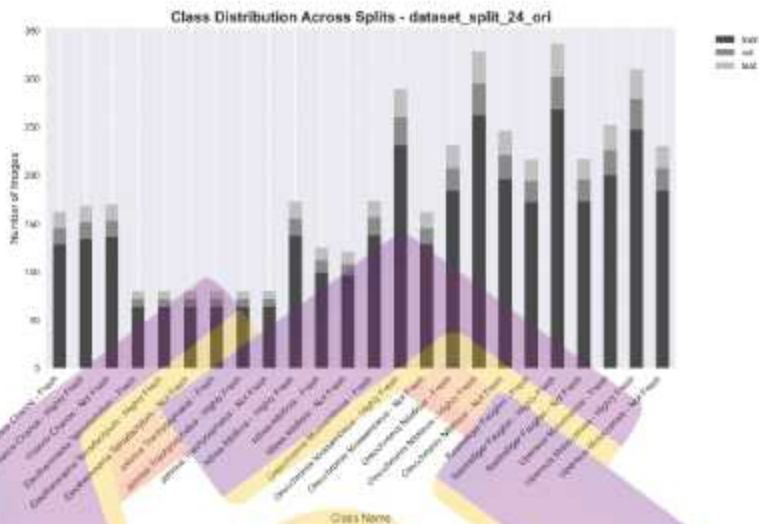
Dari total 4.050 citra, sebanyak  $\pm 3.240$  citra digunakan sebagai data latih,  $\pm 405$  citra sebagai data validasi, dan  $\pm 405$  citra sebagai data uji. Proporsi ini dipilih agar model memperoleh jumlah data pelatihan yang cukup besar untuk mempelajari pola visual yang kompleks, sekaligus menyediakan data validasi dan data uji yang memadai untuk evaluasi objektif.

Ringkasan hasil pembagian dataset berdasarkan proporsi tersebut disajikan pada Tabel 4.2, yang menunjukkan jumlah citra pada setiap subset secara agregat.

Tabel 4.2 Proporsi Pembagian Dataset

Jumlah Kelas	Total Citra	Data Latih (70%)	Data Validasi (10%)	Data Uji (20%)
24	$\pm 4.050$	$\pm 2.835$	$\pm 405$	$\pm 810$

Untuk memastikan bahwa pembagian data dilakukan secara proporsional pada seluruh kelas, distribusi dataset setelah pembagian ke dalam data latih, validasi, dan uji divisualisasikan pada Gambar 4.2. Visualisasi ini digunakan untuk memverifikasi bahwa setiap kelas mempertahankan proporsi sampel yang konsisten pada masing-masing subset data, sehingga tidak terjadi dominasi atau kekurangan representasi kelas tertentu pada data latih, validasi, maupun uji. Dengan penyajian ini, potensi bias distribusi akibat proses pembagian data dapat diidentifikasi sejak awal sebelum tahap pelatihan model dilakukan.



Gambar 4.2 Distribusi Dataset Setelah Pembagian Data Latih, Validasi, dan Uji

Gambar 4.2 menunjukkan bahwa setiap kelas mempertahankan proporsi yang relatif seragam pada ketiga subset data, baik pada data latih, data validasi, maupun data uji. Keseragaman proporsi ini menandakan bahwa tidak terjadi pergeseran distribusi kelas yang signifikan akibat proses pembagian data, sehingga setiap subset tetap merepresentasikan karakteristik dataset secara keseluruhan. Fenomena ini mengindikasikan bahwa proses stratified splitting berhasil mencegah dominasi kelas tertentu pada salah satu subset sekaligus menghindari kondisi di mana kelas dengan jumlah sampel kecil menjadi kurang terwakili. Dengan demikian, pembagian data yang dilakukan mendukung proses pelatihan dan evaluasi model yang lebih adil serta mengurangi potensi bias distribusi kelas pada tahap analisis performa.

Dengan demikian, dapat disimpulkan bahwa pembagian dataset telah dilakukan secara konsisten dan representatif, sehingga data latih, validasi, dan uji siap digunakan pada tahap pelatihan dan evaluasi model tanpa menimbulkan bias distribusi kelas.

### **4.3 Pelatihan dan Evaluasi Model**

Tahap pelatihan dan evaluasi model merupakan inti dari penelitian ini. Dataset yang telah dipersiapkan digunakan untuk melatih model deep learning yang bertujuan mengklasifikasikan tingkat kesegaran ikan berdasarkan citra mata. Proses pelatihan dirancang untuk menghasilkan model dengan performa tinggi sekaligus kemampuan generalisasi yang stabil.

#### **4.3.1 Arsitektur Model**

Penelitian ini menggunakan arsitektur EfficientNet-B0 sebagai model utama. EfficientNet dipilih karena kemampuannya mencapai keseimbangan optimal antara akurasi dan efisiensi komputasi melalui mekanisme compound scaling, yang secara serentak menyesuaikan kedalaman, lebar, dan resolusi input jaringan.

Model EfficientNet-B0 yang digunakan memanfaatkan bobot pralatih (pre-trained weights) dari ImageNet sebagai dasar ekstraksi fitur. Bagian backbone bertugas mengekstraksi representasi visual tingkat tinggi dari citra mata ikan, sementara lapisan klasifikasi akhir disesuaikan dengan jumlah kelas pada dataset, yaitu 24 kelas yang merepresentasikan kombinasi spesies dan tingkat kesegaran.

Seluruh proses pelatihan menggunakan fungsi kerugian categorical cross-entropy dengan metrik evaluasi meliputi accuracy, precision, recall, dan F1-score.

Pemantauan training loss dan validation loss dilakukan untuk menganalisis stabilitas pembelajaran serta mendeteksi potensi overfitting sejak fase awal pelatihan.

#### 4.3.2 Pelatihan Model Baseline dengan Parameter Default

Sebelum dilakukan proses optimasi hyperparameter menggunakan Optuna, model EfficientNet-B0 terlebih dahulu dilatih menggunakan parameter default sebagai baseline untuk memberikan titik perbandingan yang jelas terhadap dampak optimasi. Konfigurasi awal ini menggunakan optimizer Adam dengan learning rate awal 0,001, tanpa penyesuaian khusus pada parameter regularisasi seperti dropout rate, label smoothing, maupun weight decay di luar nilai bawaan arsitektur.

Proses pelatihan baseline dilakukan selama 100 epoch, dengan pemantauan training loss, validation loss, training accuracy, dan validation accuracy pada setiap epoch. Dinamika pelatihan divisualisasikan dalam bentuk grafik perbandingan training dan validation accuracy, yang digunakan untuk mengamati laju konvergensi model serta potensi munculnya overfitting sejak fase awal pelatihan.



Gambar 4.3 Train histori, Train Acuaracy vs Val Acuaracy Baseline/Default

Berdasarkan grafik yang disajikan pada Gambar 4.3, terlihat bahwa pada fase awal pelatihan (epoch 1–5), model mengalami peningkatan akurasi pelatihan dan validasi yang cukup cepat, yang menunjukkan bahwa arsitektur EfficientNet-B0 mampu mempelajari pola visual dasar dari data tanpa kesulitan berarti. Namun, seiring bertambahnya epoch, training accuracy terus meningkat hingga mendekati kondisi sempurna, sementara validation accuracy cenderung berfluktuasi dan tidak menunjukkan peningkatan yang konsisten. Fenomena ini diiringi oleh penurunan training loss yang tajam, sementara validation loss relatif stagnan dan bahkan meningkat pada beberapa epoch.

Kondisi tersebut mengindikasikan terjadinya overfitting progresif pada pelatihan dengan parameter default, di mana model semakin menyesuaikan diri terhadap data latih tanpa diikuti peningkatan kemampuan generalisasi pada data

validasi. Meskipun learning rate scheduling secara implisit menurunkan nilai learning rate pada epoch-epoch akhir, mekanisme ini belum cukup untuk menekan kesenjangan performa antara data latih dan validasi secara efektif.

Evaluasi pada data uji menunjukkan bahwa model baseline mencapai test accuracy sebesar 80,09%, dengan 358 prediksi benar dari 447 sampel uji. Distribusi akurasi per kelas memperlihatkan bahwa beberapa kelas, khususnya pada tingkat kesegaran Highly Fresh, mampu diklasifikasikan dengan baik, sementara kelas Fresh dan beberapa kombinasi spesies-kesegaran menunjukkan penurunan performa yang cukup signifikan. Pola ini menegaskan bahwa meskipun model baseline mampu mencapai akurasi pengujian yang kompetitif, kestabilan generalisasi antar kelas dan antar tingkat kesegaran masih belum merata.

Hasil pelatihan baseline ini menjadi dasar pembandingan utama untuk mengevaluasi sejauh mana optimasi hyperparameter menggunakan Optuna mampu memperbaiki kestabilan pelatihan, mengurangi train-validation gap, serta meningkatkan konsistensi performa model pada tahap evaluasi selanjutnya.

#### 4.4 Pengaruh Optuna terhadap Generalisasi EfficientNet-B0

Hyperparameter tuning dilakukan menggunakan Optuna dengan algoritma Tree-structured Parzen Estimator (TPE) [25]. Ruang pencarian hyperparameter mencakup variasi optimizer, learning rate, weight decay, dropout rate, freeze mode, label smoothing, dan patience, sebagaimana ditampilkan pada Tabel 4.3.

Tabel 4.3 Ruang Pencarian (Search Space) Hyperparameter

Jenis Parameter	Nilai yang Diuji	Keterangan
EfficientNet Variants	B0	Variasi arsitektur EfficientNet yang diuji dalam eksperimen.

Optimizers	Adam, AdamW, SGD	Algoritma optimasi yang digunakan untuk memperbarui bobot model.
Learning Rates	0.0001, 0.001	Kecepatan pembelajaran untuk proses training.
Weight Decays	1e-5, 1e-4, 5e-4, 1e-3, 1e-2	Regulasi untuk mencegah overfitting dengan penalti terhadap bobot besar. [26], [27]
Dropout Rates	0.3, 0.5, 0.7	Tingkat dropout untuk mengurangi risiko overfitting [28].
Freeze Modes	0, 1, 2	0 = full train, 1-2 = freeze backbone layers.
Label Smoothings	0.0, 0.05, 0.1	Teknik untuk mengurangi overconfidence model dalam prediksi kelas [29].
Patience Values	3, 5	Parameter <i>early stopping</i> untuk menentukan epoch berhenti jika tidak ada kemajuan untuk membuat model tidak overfit [30].

#### 4.4.1 Penjelasan Ruang Pencarian Hyperparameter

Tabel 4.3 mendefinisikan ruang pencarian (search space) yang digunakan Optuna selama proses hyperparameter tuning. Setiap parameter dipilih untuk mengontrol aspek berbeda dari dinamika pelatihan dan generalisasi model.

Pertama, varian EfficientNet dibatasi pada B0 untuk menjaga konsistensi arsitektur selama seluruh rangkaian eksperimen dan memastikan bahwa perbedaan performa yang diamati sepenuhnya berasal dari konfigurasi hyperparameter yang diuji, bukan dari perbedaan kompleksitas atau kapasitas model. Dengan membatasi skala arsitektur, analisis yang dilakukan dapat lebih terfokus pada pengaruh strategi optimasi dan regularisasi terhadap kinerja model, sekaligus menghindari bias yang muncul akibat penggunaan varian EfficientNet dengan jumlah parameter yang lebih besar.

Kedua, optimizer yang diuji meliputi Adam, AdamW, dan SGD. Adam dan AdamW dipilih karena stabilitas konvergensinya pada learning rate kecil,

sementara SGD disertakan sebagai pembanding berbasis momentum. Perbedaan ini berpengaruh langsung terhadap kecepatan konvergensi dan risiko overfitting.

Ketiga, learning rate diuji pada 0,0001 dan 0,001 untuk mengontrol laju pembaruan bobot. Nilai yang lebih kecil cenderung menghasilkan pembelajaran lebih stabil, sedangkan nilai lebih besar berpotensi mempercepat konvergensi namun meningkatkan risiko instabilitas.

Keempat, weight decay diuji pada rentang  $1e-5$  hingga  $1e-2$  sebagai mekanisme regularisasi untuk menekan kompleksitas model dengan memberikan penalti terhadap bobot yang memiliki nilai besar selama proses optimasi. Penerapan variasi nilai weight decay ini memungkinkan pengamatan yang lebih sistematis terhadap sejauh mana regularisasi berbasis penalti bobot mampu mengendalikan kecenderungan overfitting, yang tercermin dari perubahan train-validation gap. Dengan menguji rentang nilai yang cukup luas, penelitian ini dapat mengevaluasi keseimbangan antara kemampuan model dalam mempelajari pola data latih dan kestabilan generalisasi pada data validasi.

Kelima, dropout rate (0,3; 0,5; 0,7) digunakan untuk mengontrol tingkat regularisasi berbasis unit dropping dengan cara menonaktifkan sebagian neuron secara acak selama proses pelatihan. Mekanisme ini memaksa model untuk tidak terlalu bergantung pada aktivasi neuron tertentu, sehingga mendorong pembelajaran representasi fitur yang lebih tersebar dan robust. Pengujian beberapa tingkat dropout memungkinkan analisis pengaruh intensitas regularisasi terhadap kestabilan pelatihan, di mana nilai dropout yang lebih tinggi diharapkan mampu

menekan kecenderungan overfitting secara lebih efektif, terutama pada kondisi pelatihan tanpa penerapan augmentasi data.

Keenam, freeze mode (0, 1, 2) mengatur jumlah layer backbone yang dibekukan selama pelatihan, sehingga menentukan sejauh mana bobot pralatih (pre-trained weights) dipertahankan atau diperbarui selama proses training. Pada freeze mode 0, seluruh layer dilatih secara end-to-end, memungkinkan model menyesuaikan representasi fitur secara penuh terhadap karakteristik dataset. Sebaliknya, pada freeze mode 1 dan 2, sebagian layer awal dibekukan untuk mempertahankan fitur dasar hasil transfer learning, sementara layer yang lebih dalam tetap dilatih. Variasi ini digunakan untuk mengevaluasi trade-off antara pemanfaatan pengetahuan pralatih dan fleksibilitas pembelajaran end-to-end, serta dampaknya terhadap kecepatan konvergensi, stabilitas pelatihan, dan kemampuan generalisasi model.

Ketujuh, label smoothing (0,0; 0,05; 0,1) diterapkan untuk mengurangi overconfidence pada prediksi model dengan cara melunakkan distribusi probabilitas target selama proses pelatihan. Teknik ini mencegah model memberikan probabilitas yang terlalu ekstrem pada satu kelas tertentu, sehingga membantu mengurangi sensitivitas model terhadap noise label dan variasi kecil pada data. Dengan menurunkan tingkat keyakinan berlebih tersebut, label smoothing berkontribusi pada pembelajaran representasi yang lebih robust dan berpotensi meningkatkan stabilitas generalisasi model ketika dihadapkan pada data yang belum pernah dilihat sebelumnya.

Terakhir, *patience* (3 dan 5) menentukan sensitivitas mekanisme *early stopping* dalam menghentikan proses pelatihan ketika tidak terjadi peningkatan kinerja pada data validasi. Nilai *patience* yang lebih kecil menyebabkan pelatihan dihentikan lebih cepat, sehingga berfungsi sebagai mekanisme proteksi awal terhadap *overfitting*, sedangkan nilai yang lebih besar memberikan ruang tambahan bagi model untuk mencapai konvergensi yang lebih stabil [31]. Dengan demikian, variasi nilai *patience* memengaruhi durasi pelatihan sekaligus mengendalikan risiko *overfitting* pada fase akhir training tanpa mengorbankan potensi peningkatan performa model.

Secara keseluruhan, ruang pencarian pada Tabel 4.3 dirancang untuk menangkap interaksi antara optimasi, regularisasi, dan strategi pelatihan secara komprehensif, sehingga setiap kombinasi hyperparameter yang dievaluasi tidak berdiri sendiri, melainkan merefleksikan dinamika pelatihan model secara menyeluruh. Dengan pendekatan ini, konfigurasi terbaik yang diperoleh tidak hanya unggul dari sisi akurasi validasi, tetapi juga menunjukkan kestabilan pembelajaran, pengendalian *overfitting*, serta kemampuan generalisasi yang lebih konsisten pada data yang belum pernah dilihat sebelumnya.

#### 4.4.2 Gap Penalty Score

Untuk menghindari pemilihan model yang mengalami *overfitting*, penelitian ini mengusulkan penggunaan Gap Penalty Score sebagai fungsi objektif utama dalam proses tuning. Nilai gap dihitung sebagai selisih antara akurasi pelatihan dan akurasi validasi:

$$\text{gap} = |\text{train\_acc} - \text{val\_acc}| \quad (4.1)$$

Nilai objektif kemudian dihitung sebagai:

$$\text{objective} = \text{val\_acc} - \alpha \times \text{gap} \quad (4.2)$$

Dengan nilai  $\alpha = 0.5$ . Pendekatan ini memberikan penalti eksplisit terhadap model yang memiliki selisih performa besar antara data latih dan validasi, sehingga mendorong pemilihan model dengan kemampuan generalisasi yang lebih stabil.

Optuna juga menerapkan mekanisme pruning, yaitu penghentian dini pada trial yang menunjukkan performa tidak menjanjikan, sehingga proses tuning menjadi lebih efisien secara komputasi [32].

#### 4.4.3 Alur Optimasi Hyperparameter Menggunakan Optuna

Untuk memperjelas mekanisme optimasi hyperparameter yang diterapkan dalam penelitian ini, alur kerja Optuna dengan pendekatan Tree-structured Parzen Estimator (TPE) dan fungsi objektif berbasis Gap Penalty Score disajikan dalam bentuk pseudocode pada **Algoritma 4.1**. Penyajian ini bertujuan untuk memberikan gambaran prosedural yang eksplisit mengenai bagaimana setiap trial dijalankan, dievaluasi, dan dipilih selama proses optimasi berlangsung.

---

#### Algoritma 4.1 Mekanisme Optuna Bekerja

---

*Mulai*

- 1 *Definisikan objective\_function(trial):*
- 2 *Pilih hyperparameter dari search space:*
- 3 *optimizer*
- 4 *learning\_rate*
- 5 *weight\_decay*
- 6 *dropout\_rate*
- 7 *label\_smoothing*
- 8 *layer\_freezing*
- 9 *patience*
- 10 *Bangun model EfficientNet-B0*
- 11 *Terapkan hyperparameter*
- 12 *Latih model pada data latih*
- 13 *Evaluasi pada data validasi*
- 14 *Hitung Gap\_Penalty\_Score*
- 15 *Kembalikan Gap\_Penalty\_Score*

```

16  Inisialisasi:
17  daftar_trial ← kosong
18  jumlah_trial ← N
19  gamma ← 0.15  # kuantil pemisah trial baik-buruk (sesuai NIPS 2011)
20  trial_awal ← beberapa trial awal (eksplorasi)
21  Untuk setiap trial ke-i:
22  Jika  $i \leq \text{trial\_awal}$ :
23  # fase eksplorasi awal
24  Pilih hyperparameter secara acak
25  Jika  $i > \text{trial\_awal}$ :
26  # fase TPE
27  Urutkan daftar_trial berdasarkan Gap_Penalty_Score (ascending)
28  Tentukan:
29  n_baik ← gamma * jumlah_trial_sebelumnya
30  Kelompok_BAIK ← n_baik trial terbaik
31  Kelompok_BURUK ← sisa trial
32  Bangun distribusi probabilitas:
33  l(x) dari Kelompok_BAIK
34  g(x) dari Kelompok_BURUK
35  Untuk setiap kandidat hyperparameter x:
36  Hitung rasio  $R(x) = l(x) / g(x)$ 
37  Pilih hyperparameter x dengan nilai R(x) terbesar
38  Jalankan objective_function(trial)
39  Simpan hasil trial ke daftar_trial
40  Akhir untuk
41  Pilih hyperparameter dengan Gap_Penalty_Score terbaik
42  Latih ulang model terbaik
43  Evaluasi pada data uji
44  Selesai

```

Penyajian pseudocode pada **Algoritma 4.1** menegaskan bahwa proses optimasi hyperparameter dilakukan secara sistematis dan terstruktur. Seleksi konfigurasi terbaik tidak hanya didasarkan pada performa validasi tertinggi, tetapi pada keseimbangan antara performa dan kestabilan generalisasi yang diukur secara eksplisit melalui Gap Penalty Score. Dengan demikian, algoritma ini menjadi penghubung konseptual antara ruang pencarian hyperparameter, fungsi objektif yang digunakan, dan hasil eksperimen yang dibahas pada subbab-subsekuen.

#### 4.5 Jumlah Trial dalam Eksplorasi Hyperparameter dan Stabilitas Model

Penentuan jumlah trial optimal pada proses hyperparameter tuning dilakukan untuk memastikan bahwa ruang pencarian hyperparameter dieksplorasi secara memadai tanpa pemborosan sumber daya komputasi. Evaluasi difokuskan

pada average gap antara akurasi pelatihan (train accuracy) dan akurasi validasi (validation accuracy), di mana nilai gap yang lebih besar diinterpretasikan sebagai indikasi eksplorasi ruang hyperparameter yang lebih luas dan kemampuan membedakan kualitas konfigurasi yang lebih jelas.

Dalam eksperimen ini, tiga konfigurasi jumlah trial diuji, yaitu 20, 50, dan 100 trial. Masing-masing konfigurasi dijalankan sebanyak tiga kali percobaan independen untuk mengurangi bias akibat faktor stokastik.

#### 4.5.1 Evaluasi Konfigurasi 20 Trial

Tabel 4.4 Hasil Evaluasi Gap pada Konfigurasi 20 Trial

Try	Max Gap	Min Gap	Max Val	Min Val	Avg Gap
1	80,00	49,21	82,51	52,47	70,16
2	78,20	51,84	82,51	55,38	70,36
3	75,71	52,11	82,51	54,26	66,21
Rata Rata	77,97	51,05	82,51	54,04	68,91

Berdasarkan Tabel 4.4, nilai average gap pada tiga percobaan berada pada rentang 66,21% hingga 70,36%, dengan rata-rata keseluruhan sebesar 68,91%. Nilai maximum validation accuracy pada seluruh percobaan berhenti pada angka 82,51%, sementara minimum validation accuracy masih turun hingga kisaran 52–55%.

Fenomena ini menunjukkan bahwa meskipun konfigurasi 20 trial mampu menghasilkan beberapa model dengan performa validasi yang cukup baik, pemisahan performa antar konfigurasi masih terbatas. Hal ini mengindikasikan bahwa jumlah trial yang relatif kecil belum cukup untuk mengeksplorasi ruang hyperparameter secara menyeluruh.

Dengan demikian, konfigurasi 20 trial dinilai belum optimal karena menghasilkan average gap terendah dibandingkan konfigurasi lainnya, sehingga kurang efektif dalam membedakan konfigurasi model yang benar-benar unggul.

#### 4.5.2 Evaluasi Konfigurasi 50 Trial

Tabel 4.5 Hasil Evaluasi Gap pada Konfigurasi 50 Trial

Try	Max Gap	Min Gap	Max Val	Min Val	Avg Gap
1	79,46167182	46,09796866	84,30493274	47,75784753	68,9448419
2	79,02484641	49,11968106	84,97757848	52,91479821	70,21356989
3	79,96402426	45,47113413	84,52914798	45,96412556	70,55718732
Rata Rata	79,48351416	46,89626128	84,6038864	48,87892377	69,9051997

Pada Tabel 4.5, nilai average gap meningkat dan berada pada rentang 68,94% hingga 70,56%, dengan rata-rata keseluruhan sebesar 69,91%. Selain itu, maximum validation accuracy meningkat hingga 84,98%, lebih tinggi dibandingkan konfigurasi 20 trial.

Peningkatan ini menunjukkan bahwa penambahan jumlah trial memberikan ruang eksplorasi yang lebih luas bagi Optuna untuk menemukan kombinasi hyperparameter yang lebih beragam. Variasi nilai minimum validation accuracy yang lebih besar juga menandakan bahwa ruang pencarian mulai mencakup konfigurasi yang sangat baik hingga kurang optimal.

Meskipun demikian, selisih average gap antara konfigurasi 50 trial dan 20 trial masih relatif kecil (sekitar 1%), sehingga eksplorasi yang dilakukan belum mencapai tingkat pemisahan performa yang maksimal.

#### 4.5.3 Evaluasi Konfigurasi 100 Trial

Tabel 4.6 Hasil Evaluasi Gap pada Konfigurasi 100 Trial

Try	Max Gap	Min Gap	Max Val	Min Val	Avg Gap
1	78,86862666	45,7879656	84,97757848	47,30941704	71,45979248

2	81,60829077	44,29642448	84,52914798	47,75784753	72,46199575
3	79,4313768	50,86403336	84,52914798	52,91479821	73,69238211
Rata -Rata	79,96943141	46,98280781	84,67862481	49,32735426	72,53805678

Berdasarkan Tabel 4.6, nilai average gap meningkat secara signifikan dan berada pada rentang 71,46% hingga 73,69%, dengan rata-rata keseluruhan sebesar 72,54%. Nilai ini merupakan yang tertinggi dibandingkan dua konfigurasi sebelumnya. Selain itu, maximum validation accuracy tetap berada pada kisaran tinggi, yaitu sekitar 84,53–84,98%, dengan stabilitas yang lebih baik antar percobaan.

Fenomena ini menunjukkan bahwa dengan jumlah trial yang lebih besar, eksplorasi ruang hyperparameter menjadi lebih agresif dan menyeluruh. Perbedaan performa antar konfigurasi menjadi semakin jelas, sehingga proses seleksi model berbasis Gap Penalty Score dapat dilakukan dengan lebih tegas dan reliabel.

#### 4.5.4 Penetapan Jumlah Trial Optimal

Berdasarkan perbandingan ketiga konfigurasi tersebut, 100 trial dipilih sebagai jumlah trial optimal karena menghasilkan average gap paling besar dan paling konsisten. Nilai gap yang tinggi ini menunjukkan bahwa eksplorasi ruang hyperparameter berlangsung secara komprehensif, sehingga konfigurasi terbaik yang diperoleh lebih representatif terhadap performa dan generalisasi model.

Dengan demikian, penggunaan 100 trial memberikan dasar yang paling kuat dan rasional untuk tahap penentuan konfigurasi model terbaik yang dibahas pada Subbab 4.6.

#### 4.6 Konfigurasi Optimal EfficientNet-B0 Berdasarkan Generalisasi

Penentuan konfigurasi model terbaik dilakukan berdasarkan hasil hyperparameter tuning dengan 100 trial yang dijalankan sebanyak tiga percobaan independen. Analisis pada subbab ini difokuskan pada evaluasi keseimbangan performa dan generalisasi, yang diukur menggunakan Gap Penalty Score sebagai metrik utama.

##### 4.6.1 Analisis Hasil Percobaan 1

Tabel 4.7 Hasil Terbaik Percobaan 1 (Top-10 dari 100 Trial)

Trial	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Total Epochs
73	Adam W	0,0001	0,0001	0,5	0	0,1	5	80,72	84,42	3,70	78,87	11
52	Adam	0,0001	0,0001	0,3	0	0,05	5	84,98	98,06	13,08	78,44	19
14	Adam	0,0001	0,0001	0,5	0	0	5	84,08	96,00	11,92	78,12	15
15	Adam	0,0001	0,0001	0,3	0	0,1	5	83,63	95,14	11,51	77,88	14
12	Adam	0,0001	0,0001	0,3	0	0	5	81,84	89,82	7,98	77,83	12
85	Adam W	0,0001	0,0001	0,3	0	0	5	81,34	89,99	8,15	77,76	12
77	Adam W	0,0001	0,0001	0,3	0	0	5	83,86	96,17	12,31	77,70	16
70	Adam W	0,0001	0,0001	0,5	0	0,1	5	82,29	91,51	9,22	77,68	13
41	Adam	0,0001	0,0001	0,3	0	0	5	84,08	97,34	13,26	77,45	18
44	Adam	0,0001	0,0001	0,3	0	0	5	84,08	97,63	13,55	77,31	20

Pada Tabel 4.7 trial dengan validation accuracy tertinggi pada percobaan ini adalah Trial 52 dengan nilai 84,98%, yang secara sekilas menunjukkan performa validasi yang sangat baik dibandingkan trial lainnya. Namun demikian, train accuracy yang dicapai pada trial ini juga sangat tinggi, yaitu 98,06%, sehingga selisih antara performa pada data latih dan data validasi (train-validation gap) membesar hingga 13,08%. Gap yang besar ini mengindikasikan bahwa model telah menyesuaikan diri secara berlebihan terhadap data latih dan belum mampu

mempertahankan tingkat performa yang sebanding pada data validasi [33]. Akibatnya, meskipun nilai validation accuracy relatif tinggi, mekanisme penalti yang diterapkan melalui Gap Penalty Score menurunkan nilai objektif trial ini menjadi 78,44, sehingga Trial 52 tidak dipilih sebagai konfigurasi terbaik pada percobaan pertama.

Sebaliknya, Trial 75 menghasilkan validation accuracy yang lebih rendah, yaitu 80,72%, namun diiringi oleh train accuracy yang relatif seimbang sebesar 84,42%, sehingga train-validation gap yang terbentuk hanya 3,70%. Selisih yang kecil ini menunjukkan bahwa performa model pada data latih dan data validasi berada pada tingkat yang hampir sebanding, sehingga risiko overfitting dapat ditekan secara efektif. Meskipun nilai validation accuracy tidak setinggi trial lain seperti Trial 52, kestabilan performa antar subset data memberikan kontribusi yang lebih besar terhadap fungsi objektif. Kombinasi antara akurasi validasi yang cukup tinggi dan gap yang minimal ini menghasilkan Gap Penalty Score tertinggi pada percobaan pertama, yaitu 78,87, sehingga Trial 75 dipandang sebagai konfigurasi yang paling seimbang dari sisi performa dan kemampuan generalisasi pada percobaan ini.

Hasil ini menunjukkan bahwa peningkatan validation accuracy hingga hampir 85% tidak menjamin kualitas model apabila disertai gap di atas 10% [34], dan bahwa kestabilan generalisasi memberikan kontribusi yang lebih besar terhadap nilai objektif.

#### 4.6.2 Analisis Hasil Percobaan 2

Tabel 4.8 Hasil Terbaik Percobaan 2 (Top-10 dari 100 Trial)

Trial	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Total Epochs
7	AdamW	0,0001	0,0001	0,7	0	0	3	81,84	82,30	0,46	81,61	10
95	AdamW	0,0001	0,001	0,5	0	0,1	5	83,86	92,79	8,94	79,39	14
72	AdamW	0,0001	1,00E-05	0,5	0	0,1	5	82,29	88,79	6,50	79,04	12
94	AdamW	0,0001	0,001	0,5	0	0,1	5	82,06	88,65	6,58	78,77	12
13	AdamW	0,0001	0,0001	0,7	0	0,1	3	79,82	82,59	2,76	78,44	10
24	AdamW	0,0001	1,00E-05	0,7	0	0,1	3	81,39	87,85	6,46	78,16	11
14	AdamW	0,0001	0,0001	0,7	0	0,1	3	83,86	96,31	12,45	77,63	17
58	AdamW	0,0001	1,00E-05	0,5	0	0,1	3	83,63	95,65	12,02	77,62	14
84	AdamW	0,0001	0,0001	0,5	0	0,1	5	84,53	98,57	14,04	77,51	23
17	AdamW	0,0001	0,0001	0,7	0	0,1	3	81,61	90,51	8,89	77,17	12

Percobaan ini yang ditunjukkan pada Tabel 4.8 menghasilkan kandidat terbaik secara keseluruhan, yaitu Trial 7, dengan validation accuracy sebesar 81,84% dan train accuracy sebesar 82,30%. Kedekatan nilai antara performa pada data latih dan data validasi ini menghasilkan selisih performa yang sangat kecil, hanya 0,46%, yang menunjukkan bahwa model mampu mempertahankan kinerja yang hampir identik pada kedua subset data tersebut. Kondisi ini merefleksikan tingkat generalisasi yang sangat baik, karena model tidak menunjukkan kecenderungan untuk menyesuaikan diri secara berlebihan terhadap data latih. Selisih yang minimal ini memberikan kontribusi signifikan terhadap nilai objektif, sehingga Gap Penalty Score yang dihasilkan menjadi yang tertinggi pada seluruh percobaan, yaitu 81,61, dan menempatkan Trial 7 sebagai konfigurasi paling seimbang dari sisi performa dan kestabilan generalisasi.

Sebagai perbandingan, beberapa trial lain pada tabel yang sama memang mampu mencapai validation accuracy yang lebih tinggi, seperti 84,53%, sehingga secara sekilas tampak lebih unggul dari sisi performa validasi. Namun, capaian tersebut disertai oleh train-validation gap yang sangat besar, hingga 14,04%, yang menunjukkan adanya ketidakseimbangan yang signifikan antara performa pada data latih dan data validasi. Gap yang lebar ini mengindikasikan kecenderungan overfitting yang kuat, di mana model terlalu menyesuaikan diri terhadap data pelatihan dan gagal mempertahankan performa yang konsisten pada data validasi. Akibatnya, meskipun validation accuracy absolut lebih tinggi, mekanisme penalti pada fungsi objektif menurunkan Gap Penalty Score trial-trial tersebut ke kisaran 77-79, sehingga nilainya tetap berada di bawah konfigurasi dengan keseimbangan generalisasi yang lebih baik.

Hasil ini menunjukkan bahwa pada percobaan kedua, kontribusi minimisasi gap terhadap Gap Penalty Score lebih dominan dibandingkan peningkatan validation accuracy absolut. Trial 7 menjadi konfigurasi dengan keseimbangan terbaik antara performa dan generalisasi.

#### 4.6.3 Analisis Hasil Percobaan 3

Tabel 4.9 Hasil Terbaik Percobaan 3 (Top-10 dari 100 Trial)

Trial	Optimizer	Learning Rate	Weight Decay	Drop out Rate	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Total Epochs
52	Adam	0,0001	0,0005	0,7	0	0,1	5	84,30	94,05	9,75	79,43	16
57	Adam	0,0001	0,0005	0,3	0	0,1	5	84,30	94,31	10,00	79,30	13
55	Adam	0,0001	0,0005	0,7	0	0,1	5	80,72	77,52	3,19	79,12	11
54	Adam	0,0001	0,0005	0,7	0	0,1	5	80,49	83,53	3,04	78,98	12
13	Adam	0,001	0,0001	0,7	0	0,05	5	81,39	86,27	4,88	78,95	11
76	Adam	0,0001	0,0005	0,5	0	0,1	3	80,94	84,99	4,05	78,92	9

70	Adam W	0,000 1	0,000 5	0,3	0	0,1	3	84,30	95,40	11,09	78,76	12
92	Adam W	0,000 1	0,000 5	0,3	0	0,1	3	84,30	95,60	11,29	78,66	12
59	Adam W	0,000 1	0,000 5	0,3	0	0,1	5	84,30	96,37	12,06	78,27	15
94	Adam W	0,000 1	0,000 5	0,3	0	0,1	3	83,86	95,42	11,57	78,07	12

Trial terbaik pada percobaan ini mencapai validation accuracy hingga 84,30%, yang secara nominal termasuk salah satu nilai tertinggi yang diperoleh pada seluruh rangkaian eksperimen. Namun demikian, capaian akurasi validasi tersebut masih disertai oleh train-validation gap yang relatif besar, yaitu berada pada kisaran 9,75% hingga 12,06%. Selisih performa yang cukup lebar ini menunjukkan bahwa model cenderung mempelajari pola-pola spesifik pada data latih secara lebih dominan dibandingkan mempertahankan kinerja yang sebanding pada data validasi. Akibatnya, meskipun nilai validation accuracy terlihat tinggi, kestabilan generalisasi model belum dapat dikategorikan optimal. Kondisi ini tercermin pada nilai Gap Penalty Score tertinggi pada percobaan ini yang hanya mencapai 79,43, sehingga tetap lebih rendah dibandingkan hasil terbaik pada percobaan kedua yang menunjukkan keseimbangan performa latih dan validasi yang jauh lebih baik.

Meskipun nilai validation accuracy pada percobaan ketiga termasuk yang tertinggi di seluruh eksperimen, besarnya train-validation gap yang masih bertahan menunjukkan bahwa peningkatan nilai weight decay hingga 0,0005 belum mampu menekan overfitting secara konsisten pada berbagai kombinasi hyperparameter. Kondisi ini mengindikasikan bahwa regularisasi berbasis penalti bobot saja belum cukup untuk menyeimbangkan kompleksitas model dengan kemampuan generalisasi pada dataset ini [35]. Dengan kata lain, meskipun weight decay yang

lebih besar mampu membatasi pertumbuhan bobot secara numerik, mekanisme tersebut belum sepenuhnya efektif dalam mengendalikan kecenderungan model untuk menyesuaikan diri secara berlebihan terhadap data latih, terutama pada konfigurasi dengan kapasitas representasi yang tinggi.

#### **4.6.4 Penetapan Konfigurasi Akhir**

Untuk menentukan konfigurasi akhir, satu trial terbaik dari setiap percobaan dibandingkan secara langsung berdasarkan Gap Penalty Score. Perbandingan ini menunjukkan bahwa Trial 7 pada percobaan kedua memiliki nilai objektif tertinggi (81,61) dan train-validation gap terkecil (0,46%).

Berdasarkan hasil tersebut, konfigurasi akhir model ditetapkan menggunakan optimizer AdamW, learning rate 0,0001, weight decay 0,0001, dropout rate 0,7, freeze mode 0, label smoothing 0,1, dan patience 5.

Penetapan ini menegaskan bahwa pemilihan konfigurasi terbaik tidak didasarkan pada satu metrik tunggal, melainkan pada keseimbangan terukur antara performa validasi dan kestabilan generalisasi yang terukur secara konsisten melalui Gap Penalty Score.

#### **4.6.5 Perbandingan Performa Model Baseline dan Model Optuna**

Untuk menjawab secara eksplisit sejauh mana optimasi hyperparameter menggunakan Optuna meningkatkan kinerja dan kestabilan model, dilakukan perbandingan langsung antara model baseline dengan parameter default (Subbab 4.3) dan model hasil optimasi Optuna dengan 100 trial (Subbab 4.6). Perbandingan ini difokuskan pada perilaku pelatihan, kestabilan generalisasi, serta performa pada data uji.

Pada pelatihan baseline, dinamika training menunjukkan kecenderungan overfitting yang jelas. Training accuracy meningkat secara monoton hingga mendekati kondisi sempurna pada epoch-epoch akhir, sementara validation accuracy berfluktuasi dan tidak menunjukkan pola peningkatan yang konsisten. Kondisi ini disertai dengan gap yang semakin melebar antara performa data latih dan validasi, yang mengindikasikan bahwa model dengan parameter default cenderung menyesuaikan diri secara berlebihan terhadap data latih. Meskipun baseline masih mampu mencapai test accuracy sekitar 80%, performa tersebut diperoleh dengan stabilitas generalisasi yang tidak merata antar kelas dan sangat bergantung pada dinamika pelatihan tertentu.

Sebaliknya, model hasil optimasi Optuna dengan 100 trial menunjukkan perilaku pelatihan yang jauh lebih stabil. Konfigurasi terpilih menghasilkan training accuracy dan validation accuracy yang berdekatan sejak fase awal pelatihan, dengan gap yang secara konsisten kecil. Hal ini menunjukkan bahwa optimasi hyperparameter tidak hanya meningkatkan performa numerik, tetapi juga berhasil mengendalikan kompleksitas model melalui kombinasi regularisasi (dropout tinggi, label smoothing, dan weight decay) serta pengaturan learning rate dan early stopping yang lebih tepat.

Dengan demikian, perbandingan antara model baseline dan model hasil optimasi Optuna menunjukkan bahwa kontribusi utama Optuna dalam penelitian ini bukan terletak pada peningkatan akurasi secara drastis, melainkan pada kemampuannya dalam menemukan konfigurasi hyperparameter yang menghasilkan perilaku pelatihan yang lebih stabil dan performa yang lebih

konsisten. Hal ini tercermin dari kecilnya generalization gap, kedekatan nilai training dan validation accuracy, serta konsistensi hasil pada pelatihan ulang model

Dari sisi evaluasi pada data uji, model teroptimasi mempertahankan test accuracy yang sebanding bahkan sedikit lebih stabil dibandingkan baseline, namun dengan distribusi performa antar kelas yang lebih merata. Perbedaan utama tidak terletak pada lonjakan akurasi absolut, melainkan pada peningkatan kestabilan dan reliabilitas generalisasi, yang tercermin dari kecilnya selisih performa latihan–validasi dan konsistensi hasil pada pelatihan ulang (dibahas pada Subbab 4.7).

Dengan demikian, perbandingan ini menegaskan bahwa kontribusi utama Optuna dalam penelitian ini bukan semata-mata meningkatkan nilai akurasi, melainkan menggeser perilaku model dari kondisi overfitting pada baseline menuju konfigurasi yang lebih stabil, seimbang, dan generalis. Hasil ini memperkuat justifikasi penggunaan optimasi hyperparameter berbasis Optuna dengan fungsi objektif Gap Penalty Score sebagai langkah krusial dalam meningkatkan kualitas model secara menyeluruh.

#### 4.7 Kinerja dan Konsistensi Model Tanpa Augmentasi Data

Setelah konfigurasi terbaik ditetapkan pada Subbab sebelumnya, model dilatih ulang untuk mengevaluasi konsistensi dan stabilitas performa. Pelatihan ulang dilakukan sebanyak lima kali dengan konfigurasi dan pembagian data yang identik, sehingga variasi performa—apabila muncul—hanya berasal dari faktor stokastik seperti inisialisasi bobot dan urutan mini-batch.

Tabel 4.10 Hasil Pelatihan Ulang Model Terbaik (5 Kali Run)

train	best train accuracy	best val accuracy	test accuracy	best epoch	total epochs
1	83,6145	81,1659	79,8658	7	12

2	83,6145	81,1659	79,8658	7	12
3	83,6145	81,1659	79,8658	7	12
4	83,6145	81,1659	79,8658	7	12
5	83,6145	81,1659	79,8658	7	12
	<b>83,6145</b>	<b>81,1659</b>	<b>79,8658</b>		

Berdasarkan Tabel 4.10, seluruh run menghasilkan metrik yang identik: best train accuracy sebesar 83,61%, best validation accuracy sebesar 81,17%, dan test accuracy sebesar 79,87%. Selain itu, seluruh run mencapai best epoch yang sama, yaitu epoch ke-7, dan berhenti pada epoch ke-12 sesuai mekanisme early stopping.

Fenomena tidak adanya variasi antar-run ini menunjukkan bahwa konfigurasi hyperparameter yang dipilih memiliki stabilitas pelatihan yang sangat tinggi dan tidak sensitif terhadap faktor stokastik. Selisih antara train accuracy dan validation accuracy yang konsisten, yaitu sekitar 2,44%, mengindikasikan bahwa model tidak mengalami fluktuasi overfitting maupun underfitting antar-pelatihan.

Sebagai penegasan, identiknya hasil pada kelima run merepresentasikan determinisme praktis dari konfigurasi model terpilih. Temuan ini memperkuat klaim bahwa seleksi berbasis Gap Penalty Score berhasil mengarahkan proses tuning menuju konfigurasi yang tidak hanya optimal secara numerik, tetapi juga konsisten dan stabil dalam kemampuan generalisasi.

#### **4.8 Kinerja Klasifikasi Kurva Pelatihan dan Confusion Matrix**

Untuk menganalisis dinamika proses pelatihan secara lebih mendalam, hubungan antara training accuracy dan validation accuracy terhadap epoch divisualisasikan pada Gambar 4.4. Visualisasi ini digunakan untuk mengamati laju konvergensi model, kestabilan proses pembelajaran, serta munculnya indikasi awal

overfitting atau underfitting selama pelatihan berlangsung, sehingga perilaku model pada setiap fase training dapat dianalisis secara terukur dan sistematis.



Gambar 4.4 Train histori, Train Acuaracy vs Val Acuaracy

Pada epoch awal (epoch ke-1), training accuracy masih berada pada kisaran 16,3%, sedangkan validation accuracy sudah mencapai sekitar 34,98%. Hingga epoch ke-5, training accuracy meningkat tajam hingga sekitar 63,97%, sementara validation accuracy meningkat lebih cepat hingga 73,09%.

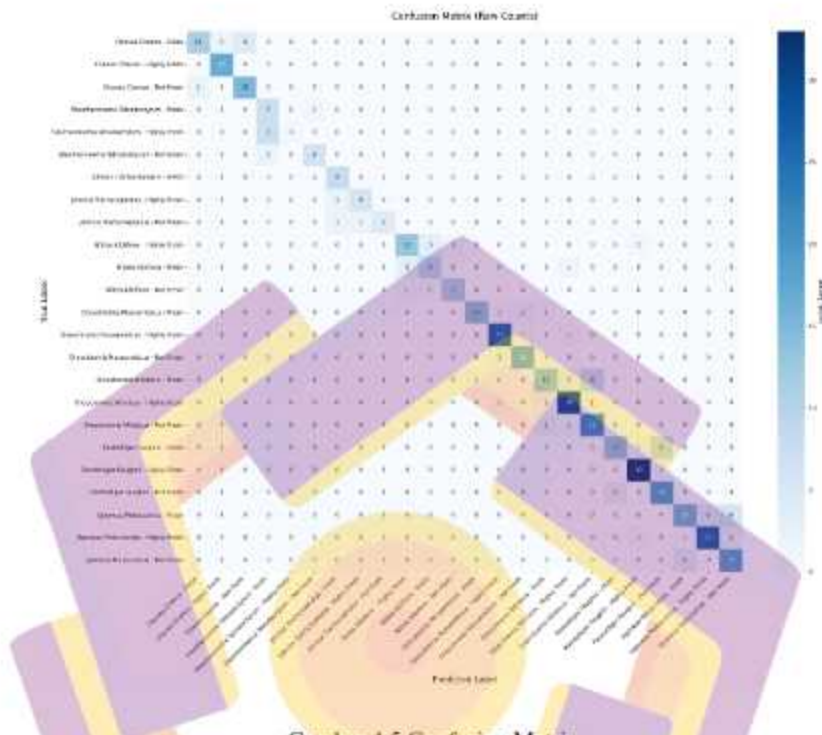
Fenomena ini menunjukkan bahwa pada fase awal pelatihan, model lebih cepat melakukan generalisasi dibandingkan melakukan penyesuaian spesifik terhadap data latih. Kondisi ini selaras dengan penggunaan dropout tinggi (0,7) dan label smoothing (0,1), yang secara eksplisit menahan laju pembelajaran model pada data training.

Puncak validation accuracy terjadi pada epoch ke-7 dengan nilai 81,17%, sementara training accuracy pada epoch yang sama berada pada kisaran 83,61%, menghasilkan gap sebesar 2,44%. Gap yang relatif kecil ini menandakan kondisi generalisasi yang optimal.

Setelah epoch ke-7, training accuracy terus meningkat hingga mencapai 94,42% pada epoch ke-12, sedangkan validation accuracy justru menurun hingga sekitar 79,82%. Fenomena divergensi ini merupakan indikasi awal terjadinya overfitting.

Dengan demikian, mekanisme early stopping dengan patience 5 epoch berhasil menghentikan pelatihan pada epoch ke-12 sebelum gap berkembang lebih besar, sehingga performa generalisasi model tetap terjaga.

Untuk mengevaluasi pola kesalahan prediksi model secara lebih rinci, Confusion Matrix disajikan pada Gambar 4.5.



Gambar 4.5 Confusion Matrix

Confusion Matrix merefleksikan capaian test accuracy keseluruhan sebesar 79,87%, yang berarti bahwa dari sekitar 405 citra data uji, sebanyak  $\pm 323$  citra berhasil diklasifikasikan dengan benar, sementara sisanya mengalami salah klasifikasi. Dominasi nilai pada diagonal utama matriks memperlihatkan bahwa sebagian besar prediksi tepat berada pada kelas yang sesuai dengan label sebenarnya.

Secara lebih rinci, beberapa kelas menunjukkan performa klasifikasi yang sangat kuat. Kelas *Rastrelliger faughni* – *Highly Fresh* mencatat 33 prediksi benar dengan hanya 1 kesalahan ke kelas *Fresh*, sedangkan *Oreochromis niloticus* – *Highly Fresh* memiliki 30 prediksi benar dengan kesalahan terbatas pada kelas

kesegaran yang berdekatan. Pola ini menunjukkan bahwa pada kelas Highly Fresh, model mampu mempertahankan konsistensi prediksi yang tinggi.

Sebaliknya, kesalahan prediksi hampir sepenuhnya terjadi antar tingkat kesegaran dalam spesies yang sama. Pada *Chanos chanos* – Fresh, dari total 17 sampel uji, hanya 11 sampel yang diprediksi dengan benar, sementara 2 sampel keliru diprediksi sebagai Highly Fresh dan 4 sampel sebagai Not Fresh. Pola serupa juga terlihat pada *Nibea albiflora* – Fresh, di mana 9 dari 12 sampel diprediksi dengan benar, sementara sisanya bergeser ke kelas kesegaran lain dalam spesies yang sama.

Kelas Fresh secara konsisten menunjukkan tingkat kesalahan tertinggi. Pada beberapa spesies, lebih dari sepertiga sampel Fresh mengalami salah klasifikasi, sedangkan kelas Highly Fresh dan Not Fresh cenderung memiliki jumlah prediksi benar yang lebih dominan. Tidak ditemukan kasus signifikan di mana sampel dari satu spesies diprediksi sebagai spesies lain, yang menegaskan bahwa kesalahan model bersifat intra-spesies, bukan lintas spesies.

Dengan demikian, Confusion Matrix secara menegaskan bahwa model memiliki kemampuan yang sangat kuat dalam membedakan spesies ikan, tetapi menghadapi tantangan utama pada pemisahan tingkat kesegaran yang bersifat gradual, khususnya pada kelas Fresh. Temuan ini memperjelas bahwa keterbatasan model tidak terletak pada identifikasi objek, melainkan pada ambiguitas visual antar level kesegaran.

#### **4.9 Analisis Robustness EfficientNet-B0 Tanpa Oversampling**

Subbab ini membahas perilaku pengambilan keputusan model EfficientNet-B0 secara lebih mendalam melalui analisis kesalahan prediksi (*incorrect detection*), visualisasi Grad-CAM, serta evaluasi robustness model terhadap berbagai jenis gangguan visual. Analisis dilakukan untuk memahami tidak hanya apa yang diprediksi model, tetapi juga mengapa keputusan tersebut diambil dan sejauh mana keputusan tersebut tetap stabil ketika kondisi visual mengalami perubahan.

Berbeda dengan Confusion Matrix pada Subbab 4.8 yang menyajikan pola kesalahan secara agregat pada tingkat kelas, pembahasan pada subbab ini difokuskan pada level individual citra. Pendekatan ini memungkinkan penelusuran langsung terhadap karakteristik visual yang menyebabkan kesalahan prediksi maupun perubahan tingkat keyakinan model.

##### **4.9.1 Analisis Kasus Salah Klasifikasi (Incorrect Detection Analysis)**

Analisis pada subbab ini bertujuan untuk menelaah secara lebih rinci sampel-sampel data uji yang mengalami salah klasifikasi oleh model. Jika Confusion Matrix pada Subbab 4.8 digunakan untuk mengidentifikasi pola kesalahan secara agregat pada tingkat kelas, maka analisis *incorrect detection* difokuskan pada kasus individual yang merepresentasikan kegagalan model dalam membedakan kelas tertentu.

Daftar sampel salah klasifikasi disusun dengan mencatat path citra, label sebenarnya (*ground truth*), label hasil prediksi, indeks kelas, serta nilai confidence yang dihasilkan model. Penyajian informasi ini memungkinkan penelusuran langsung terhadap karakteristik visual citra yang salah diklasifikasikan sekaligus

tingkat keyakinan model pada keputusan yang keliru tersebut. Tabel berikut menampilkan 10 sampel pertama dari data hasil prediksi model. Seluruh data lengkap tersedia pada lampiran.

Tabel 4.11 Sampel incorrect classification pada data uji

image_path	ground truth	predicted	gt_idx	pred_idx	confidence
Chanos Chanos - Fresh/IMG_20191003_065053.jpg	Fresh	Not Fresh	0	2	0.6009
Chanos Chanos - Fresh/IMG_20191003_064729.jpg	Fresh	Not Fresh	0	2	0.4444
Chanos Chanos - Fresh/IMG_20191002_062711.jpg	Fresh	Highly Fresh	0	1	0.8411
Chanos Chanos - Fresh/IMG_20191112_064326.jpg	Fresh	Highly Fresh	0	1	0.8850
Chanos Chanos - Fresh/IMG_20191003_064420.jpg	Fresh	Not Fresh	0	2	0.6112
Chanos Chanos - Fresh/IMG_20191003_064459.jpg	Fresh	Not Fresh	0	2	0.7020
Chanos Chanos - Not Fresh/IMG_20191114_054148.jpg	Not Fresh	Fresh	2	0	0.5330
Chanos Chanos - Not Fresh/IMG_20191004_063606.jpg	Not Fresh	Fresh	2	0	0.5188
Eleutheronema Tetradactylum - Fresh/IMG_20191017_070934.jpg	Fresh	Not Fresh	3	5	0.4536
Eleutheronema Tetradactylum - Highly Fresh/IMG_20191015_071556.jpg	Highly Fresh	Fresh	4	3	0.6967

Tabel 4.11 menyajikan daftar sampel data uji yang mengalami salah klasifikasi, yang disusun berdasarkan hasil inferensi model pada data uji. Setiap baris tabel memuat informasi image path, ground truth, label hasil prediksi, indeks kelas, serta nilai confidence model. Berdasarkan tabel tersebut, kesalahan prediksi paling banyak muncul pada kelas dengan tingkat kesegaran Fresh, baik yang diprediksi sebagai Highly Fresh maupun Not Fresh.

Sebagai contoh, pada kelas Chanos chanos – Fresh, terdapat 6 sampel yang salah diklasifikasikan, dengan 4 sampel diprediksi sebagai Not Fresh dan 2 sampel sebagai Highly Fresh. Nilai confidence pada kesalahan ini berada pada rentang 0,44 hingga 0,88, yang menunjukkan bahwa kesalahan tidak selalu terjadi pada kondisi

ketidakpastian ekstrem. Pola serupa juga terlihat pada kelas *Eleutheronema tetradactylum* – Highly Fresh, di mana 7 sampel diprediksi sebagai Fresh dengan nilai confidence berkisar antara 0,45 hingga 0,69.

Selain itu, terdapat pula kesalahan dengan tingkat keyakinan yang sangat tinggi (*overconfident error*). Salah satu contohnya adalah pada kelas *Nibea albiflora* – Not Fresh, yang diprediksi sebagai Fresh dengan nilai confidence mencapai 0,94. Keberadaan kesalahan dengan keyakinan tinggi ini menunjukkan bahwa model terkadang memberikan keputusan yang keliru meskipun sinyal probabilistik yang dihasilkan tampak kuat. Fenomena ini mengindikasikan adanya kemiripan visual yang signifikan antar tingkat kesegaran dalam spesies yang sama, terutama pada transisi antara kelas Fresh dan kelas di sekitarnya.

Tabel 4.11 digunakan sebagai dasar pemilihan contoh citra pada analisis Grad-CAM klasifikasi salah di Subbab 4.9.3. Dengan demikian, visualisasi Grad-CAM yang dibahas tidak dipilih secara acak, melainkan merepresentasikan pola kesalahan yang benar-benar terjadi pada data uji. Pendekatan ini memastikan bahwa analisis visual yang dilakukan bersifat sistematis, dapat ditelusuri kembali ke data numerik, serta konsisten dengan temuan pada Confusion Matrix sebelumnya.

Secara umum, tabel ini menunjukkan bahwa sebagian besar kesalahan prediksi terjadi pada kelas dengan tingkat kesegaran yang berdekatan dalam spesies yang sama, khususnya pada kelas Fresh. Selain itu, beberapa kasus salah klasifikasi juga disertai nilai confidence yang relatif tinggi, yang mengindikasikan bahwa kesalahan tersebut bukan semata-mata akibat ketidakpastian model, melainkan disebabkan oleh kemiripan visual antar kelas. Temuan ini memperkuat hasil analisis

Confusion Matrix bahwa tantangan utama model terletak pada pemisahan tingkat kesegaran yang bersifat gradual, bukan pada identifikasi spesies ikan.

#### 4.9.2 Analisis Grad-CAM pada Klasifikasi Benar



Gambar 4.6. Grad-CAM pada Klasifikasi Benar Chanos chanos (Fresh)

Pada Gambar 4.6, model memprediksi kelas *Chanos chanos* – Fresh dengan confidence sebesar 98,79%. Dua probabilitas tertinggi berikutnya berada di bawah 0,3%, menunjukkan margin keputusan yang sangat tajam.

Fenomena ini diperkuat oleh visualisasi Grad-CAM yang menunjukkan aktivasi terfokus pada area mata dan kepala ikan, yang secara biologis merupakan indikator utama tingkat kesegaran. Aktivasi tidak menyebar ke latar belakang, menandakan bahwa model memanfaatkan fitur yang relevan secara semantik.

Dengan demikian, dapat disimpulkan bahwa pada kasus klasifikasi benar, model tidak hanya menghasilkan prediksi yang akurat secara numerik, tetapi juga mendasarkan keputusannya pada fitur visual yang tepat dan dapat dijelaskan.



#### 4.9.3 Analisis Grad-CAM pada Klasifikasi Salah

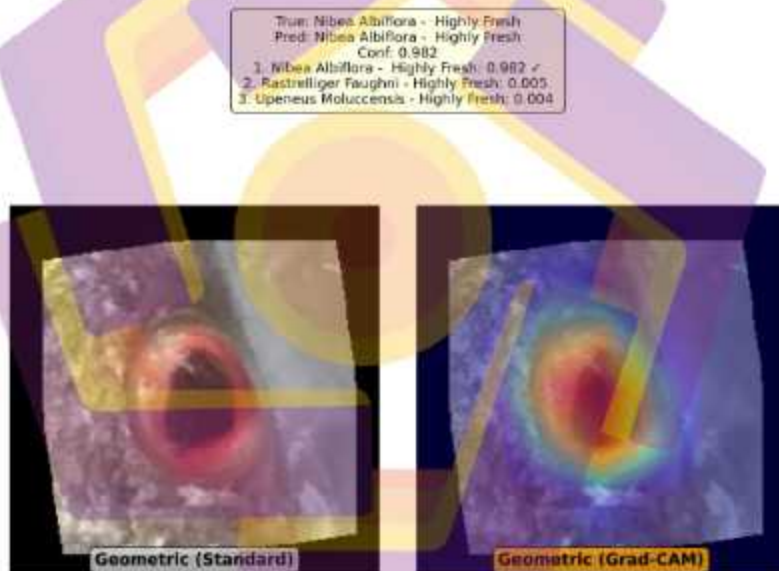


Pada Gambar 4.7, sampel dengan label asli Chanos chanos – Fresh diprediksi sebagai Not Fresh dengan confidence 60,09%, sementara probabilitas kelas sebenarnya hanya 5,11%.

Visualisasi Grad-CAM menunjukkan bahwa model tetap memfokuskan perhatian pada area mata ikan. Namun, overlap karakteristik visual antara kelas Fresh dan Not Fresh menyebabkan interpretasi fitur bergeser ke kelas yang salah.

Kesimpulannya, kesalahan klasifikasi ini bukan disebabkan oleh kegagalan model dalam menentukan area penting, melainkan oleh kemiripan visual antar kelas kesegaran yang bersifat gradual.

#### 4.9.4 Robustness terhadap Geometric Attack



Gambar 4.8. Geometrical Attack pada Nibea albiflora – Highly Fresh

Pada Gambar 4.8, citra Nibea albiflora – Highly Fresh tetap diprediksi dengan benar dengan confidence 98,17%, hanya mengalami penurunan kurang dari 1% dibandingkan kondisi normal. Nilai confidence yang masih sangat tinggi ini

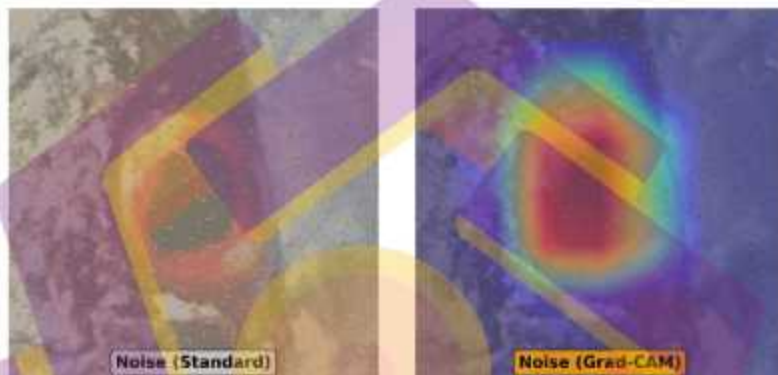
menunjukkan bahwa transformasi geometris yang diterapkan tidak secara signifikan memengaruhi representasi fitur utama yang digunakan model dalam proses pengambilan keputusan. Dengan kata lain, meskipun terjadi perubahan posisi atau skala objek, model tetap mampu mempertahankan keyakinan prediksi yang hampir setara dengan kondisi citra asli.

Fenomena ini menunjukkan bahwa transformasi spasial seperti rotasi dan skala tidak mengganggu representasi fitur utama yang dipelajari model. Ketahanan ini mengindikasikan bahwa model telah mempelajari fitur yang bersifat invarian terhadap perubahan posisi dan ukuran objek, sehingga proses ekstraksi ciri tidak bergantung secara ketat pada orientasi atau skala citra. Dengan demikian, model tetap mampu mengenali pola visual kunci yang relevan meskipun terjadi variasi spasial pada data masukan.

Dengan demikian, model memiliki robustness yang sangat baik terhadap gangguan geometris dan bersifat invarian terhadap perubahan posisi objek.

#### 4.9.5 Robustness terhadap Noise Attack

True: Nibea Albiflora - Highly Fresh  
 Pred: Nibea Albiflora - Highly Fresh  
 Conf: 0.272  
 1. Nibea Albiflora - Highly Fresh: 0.272 ✓  
 2. Upeneus Moluccensis - Highly Fresh: 0.120  
 3. Upeneus Moluccensis - Fresh: 0.088



Gambar 4.9 . Noise Attack pada Nibea albiflora – Highly Fresh

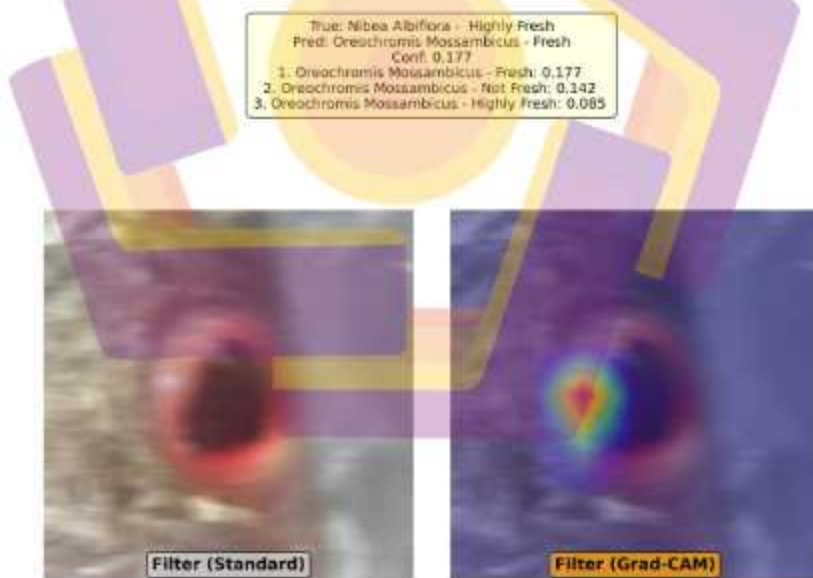
Pada Gambar 4.9, meskipun model masih berhasil memprediksi kelas yang benar, tingkat keyakinan (confidence) mengalami penurunan yang sangat signifikan hingga hanya mencapai 27,15%, jauh lebih rendah dibandingkan kondisi tanpa gangguan maupun pada pengujian dengan geometric attack. Penurunan ini menunjukkan bahwa meskipun keputusan akhir model masih tepat secara kelas, kestabilan probabilistik prediksi telah terpengaruh secara substansial oleh keberadaan noise pada citra masukan.

Fenomena ini menunjukkan bahwa keberadaan noise secara signifikan merusak detail tekstur halus dan pola lokal pada citra yang selama ini menjadi dasar

utama pengambilan keputusan model. Gangguan berbasis noise menyebabkan informasi visual penting, seperti perbedaan intensitas, kontur halus, dan transisi tekstur antar area objek, menjadi tidak lagi terdefinisi dengan jelas, sehingga representasi fitur yang diekstraksi oleh model menjadi kurang stabil dan ambigu. Akibatnya, meskipun model masih mampu mempertahankan prediksi kelas yang benar, tingkat keyakinan terhadap keputusan tersebut menurun secara drastis karena sinyal visual yang mendukung klasifikasi tidak lagi cukup kuat dan konsisten.

Kesimpulannya, model masih bersifat fungsional terhadap noise, tetapi reliabilitas prediksi menurun secara signifikan.

#### 4.9.6 Robustness terhadap Filter Attack



Gambar 4.10 . Filter Attack pada Nibeal albiflora – Highly Fresh

Pada Gambar 4.10, model gagal melakukan klasifikasi dengan benar dan memprediksi kelas *Oreochromis mossambicus* – Fresh dengan confidence hanya 17,68%, yang menunjukkan tingkat keyakinan yang sangat rendah dibandingkan seluruh skenario pengujian sebelumnya. Nilai confidence yang kecil ini mengindikasikan bahwa model berada dalam kondisi ketidakpastian tinggi, di mana fitur-fitur visual yang diekstraksi tidak cukup kuat untuk mendukung keputusan klasifikasi yang tegas. Kondisi tersebut mencerminkan bahwa filter attack secara signifikan telah merusak karakteristik tekstur dan ketajaman citra, sehingga representasi fitur yang terbentuk menjadi kurang informatif dan tidak lagi merepresentasikan kelas aslinya secara jelas.

Distribusi probabilitas yang menyebar menunjukkan degradasi representasi fitur yang parah akibat modifikasi tekstur dan ketajaman citra. Penyebaran probabilitas ini menandakan bahwa model tidak lagi mampu membentuk batas keputusan yang jelas antar kelas, karena fitur-fitur diskriminatif yang sebelumnya menjadi dasar pemisahan kelas telah mengalami distorsi signifikan. Akibatnya, kontribusi setiap kelas terhadap keputusan akhir menjadi relatif merata dan tidak dominan, yang mencerminkan hilangnya kejelasan representasi visual serta meningkatnya ambiguitas dalam proses klasifikasi.

Dengan demikian, dapat disimpulkan bahwa filter attack merupakan jenis gangguan paling destruktif bagi model dibandingkan geometric dan noise attack.

#### **4.10 Evaluasi generalisasi model**

Evaluasi generalisasi model pada penelitian ini tidak hanya dilakukan melalui pembagian data latih, validasi, dan uji secara acak, tetapi juga diperluas

dengan skema leave-source-out (LSO). Pendekatan LSO digunakan untuk secara eksplisit menjawab pertanyaan terkait kemampuan generalisasi model terhadap spesies ikan yang sama sekali tidak pernah muncul pada tahap pelatihan.

Dalam skema LSO, satu spesies ikan dikeluarkan sepenuhnya dari data latih dan digunakan sebagai data uji, sementara data dari tujuh spesies lainnya digunakan untuk melatih model. Proses ini diulang hingga setiap spesies pernah berperan sebagai spesies uji. Dengan pendekatan ini, performa model merefleksikan kemampuan transfer pengetahuan antar spesies, bukan sekadar generalisasi intra-spesies.

Tabel 4.12 Ringkasan hasil evaluasi LSO disajikan

No	Test Species	Accuracy (%)	Macro Precision	Macro Recall	Macro F1	Test Samples	Best Val Acc (%)
1	Chanos Chanos	51	0.5094	0.5079	0.5037	500	79.43
2	Eleutheronema Tetradactylum	44.58	0.4216	0.4458	0.3784	240	79.28
3	Johnius Trachycephalus	54.58	0.5321	0.5458	0.4625	240	80
4	Nibea Albiflora	56.32	0.5581	0.562	0.5593	419	77.61
5	Oreochromis Mossambicus	70.4	0.6851	0.6884	0.6864	625	75.03
6	Oreochromis Niloticus	65.09	0.6452	0.6327	0.6352	805	80.89
7	Rastrelliger Faughni	60.86	0.6102	0.6014	0.5886	769	76.69
8	Upeneus Moluccensis	55.56	0.5428	0.5428	0.5424	792	79.44

Berdasarkan Tabel 4.12, terlihat bahwa performa model bervariasi cukup signifikan antar spesies uji. Spesies *Oreochromis mossambicus* menunjukkan performa tertinggi dengan akurasi 70,40%, disertai nilai macro precision, macro recall, dan macro F1 yang relatif seimbang. Hasil ini mengindikasikan bahwa karakteristik visual mata ikan pada spesies ini memiliki kemiripan yang cukup kuat

dengan spesies lain dalam data latih, sehingga representasi fitur yang dipelajari model dapat ditransfer dengan lebih efektif.

Sebaliknya, spesies *Eleutheronema tetradactylum* menghasilkan performa terendah dengan akurasi 44,58% dan nilai macro F1 yang juga paling kecil dibandingkan spesies lainnya. Fenomena ini menunjukkan bahwa pola visual mata ikan pada spesies tersebut memiliki karakteristik yang lebih spesifik dan kurang terwakili oleh spesies lain dalam data latih. Akibatnya, fitur yang dipelajari model dari tujuh spesies lain tidak sepenuhnya mampu menangkap variasi visual yang relevan pada spesies ini.

Spesies lain seperti *Chanos chanos*, *Johnius trachycephalus*, *Nibea albiflora*, *Oreochromis niloticus*, *Rastrelliger faughni*, dan *Upeneus moluccensis* menunjukkan performa menengah dengan akurasi berada pada rentang 51% hingga 65%. Pola ini mengindikasikan bahwa meskipun model masih mampu melakukan klasifikasi tingkat kesegaran secara moderat pada spesies yang tidak pernah dilihat sebelumnya, kemampuan generalisasi antar spesies belum sepenuhnya merata. Variasi performa ini mencerminkan perbedaan tingkat kemiripan visual antar spesies, baik dari segi struktur mata, tekstur, maupun pola degradasi kesegaran.

Perlu dicermati bahwa nilai *best validation accuracy* selama pelatihan pada skema LSO tetap berada pada kisaran tinggi, yaitu sekitar 75% hingga 80%, meskipun akurasi pengujian pada spesies yang ditinggalkan menurun secara signifikan. Fenomena ini menunjukkan adanya kesenjangan antara generalisasi intra-spesies dan generalisasi antar-spesies. Model mampu mencapai performa validasi yang baik pada spesies yang termasuk dalam data latih, namun performa

tersebut tidak sepenuhnya dapat dipertahankan ketika dihadapkan pada spesies baru.

Secara keseluruhan, hasil evaluasi LSO menegaskan bahwa model EfficientNet-B0 yang diusulkan memiliki kemampuan generalisasi yang cukup baik pada variasi tingkat kesegaran dalam spesies yang telah dilihat, tetapi masih menghadapi tantangan dalam mentransfer pengetahuan secara konsisten ke spesies yang sepenuhnya baru. Temuan ini memperkuat interpretasi bahwa performa tinggi pada skema pembagian data konvensional tidak serta-merta menjamin robustnya generalisasi lintas spesies, serta menegaskan pentingnya evaluasi LSO sebagai pelengkap analisis kinerja model pada dataset multi-spesies.

#### 4.11 Perbandingan dengan Penelitian Sebelumnya

Subbab ini membahas posisi dan kontribusi penelitian yang dilakukan dibandingkan dengan penelitian-penelitian terdahulu yang memiliki fokus serupa pada klasifikasi tingkat kesegaran ikan berbasis citra mata. Perbandingan difokuskan pada metode yang digunakan, strategi optimasi, serta capaian performa terukur pada dataset yang sama.

Tabel 4.13 Perbandingan Penelitian Sebelumnya dengan Penelitian Ini

Author / Tahun	Dataset	Preprocessing / Augmentasi	Hyperparameter / Optimasi	Metode / Model	Hasil / Output
Prasetyo et al., 2022, <i>Information Processing in Agriculture</i>	The Freshness of the Fish Eyes Dataset	-	-	DSC-BE, Residual Transition, MobileNetV1 (MB-BE)	Accuracy 57.5%; Precision 72.9%; Recall 57.5%
Guo et al., 2023, <i>EITCE Conference</i>	The Freshness of the Fish Eyes Dataset	-	-	MobileNetV1 + Attention (SENet, ECA, CA, CBAM)	ECA Validation Accuracy 73.21%

Author / Tahun	Dataset	Preprocessing / Augmentasi	Hyperparameter / Optimasi	Metode / Model	Hasil / Output
Yildiz et al., 2024, <i>European Food Research &amp; Technology</i>	The Freshness of the Fish Eyes Dataset	-	-	VGG19 + ANN; SqueezeNet + ML (k-NN, SVM, ANN, RF)	VGG19 + ANN; 77.3% Accuracy
<b>Penelitian Ini (2025)</b>	The Freshness of the Fish Eyes Dataset	-	Optuna	EfficientNetb0	Accuracy 79,87 %; Precision 80%; Recal 77%

Berdasarkan Tabel 4.13, penelitian-penelitian terdahulu seperti Prasetyo et al. (2022), Guo et al. (2023), dan Yildiz et al. (2024) umumnya masih mengandalkan arsitektur CNN konvensional atau lightweight seperti MobileNetV1 dan VGG19, dengan strategi pelatihan yang relatif sederhana dan pemilihan hyperparameter yang bersifat statis tanpa optimasi sistematis. Konfigurasi model pada studi-studi tersebut pada umumnya ditentukan secara manual atau mengikuti pengaturan baku dari arsitektur yang digunakan, sehingga ruang eksplorasi parameter pelatihan menjadi terbatas. Akibatnya, performa model sangat sensitif terhadap konfigurasi awal yang dipilih dan kurang merefleksikan potensi maksimal arsitektur yang digunakan. Hal ini tercermin dari capaian performa yang dilaporkan, di mana nilai akurasi berada pada rentang 57,5% hingga 77,3%, dengan variasi kinerja yang cukup besar antar pendekatan dan belum menunjukkan kestabilan generalisasi yang kuat pada dataset multi-spesies dengan kompleksitas tinggi.

Sebaliknya, penelitian ini menggunakan arsitektur EfficientNet-B0 yang dipadukan dengan optimasi hyperparameter berbasis Optuna. Pendekatan ini memungkinkan eksplorasi ruang hyperparameter secara adaptif melalui Bayesian

optimization, sehingga konfigurasi yang diperoleh tidak hanya berfokus pada peningkatan akurasi validasi, tetapi juga pada kestabilan generalisasi model.

Model yang diusulkan mencapai test accuracy sebesar 79,87%, dengan precision 80% dan recall 77% pada dataset yang mencakup 8 spesies ikan dan 3 tingkat kesegaran, tanpa menerapkan augmentasi. Nilai ini melampaui capaian seluruh penelitian perbandingan yang menggunakan dataset yang sama, sekaligus dicapai pada kondisi pelatihan yang lebih ketat.

Selain peningkatan performa numerik, kontribusi utama penelitian ini terletak pada penggunaan Gap Penalty Score sebagai kriteria seleksi model. Pendekatan ini secara eksplisit mempertimbangkan keseimbangan antara training accuracy dan validation accuracy, sehingga konfigurasi dengan risiko overfitting tinggi tidak terpilih meskipun memiliki akurasi validasi yang besar.

Lebih lanjut, penelitian ini memperluas ruang evaluasi dengan melakukan analisis robustness terhadap geometric attack, noise attack, dan filter attack. Aspek ini belum dibahas secara eksplisit pada penelitian-penelitian sebelumnya, padahal sangat relevan untuk skenario penerapan dunia nyata.

Dengan demikian, dapat disimpulkan bahwa penelitian ini tidak hanya memberikan peningkatan performa klasifikasi, tetapi juga menawarkan kontribusi metodologis berupa seleksi model berbasis keseimbangan generalisasi dan evaluasi robustness tanpa ketergantungan pada augmentasi data. Kombinasi ini menempatkan penelitian ini sebagai pendekatan yang lebih stabil, generalis, dan aplikatif dibandingkan studi-studi terdahulu.

## BAB 5 PENUTUP

### 5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, serta mengacu pada dua rumusan masalah yang telah ditetapkan, dapat disimpulkan sebagai berikut:

1. Kinerja model EfficientNet-B0 dalam mengklasifikasikan tingkat kesegaran ikan berbasis citra mata pada dataset multi-spesies dapat dikategorikan cukup baik dan stabil. Model mampu mencapai akurasi data uji sebesar 79,87%, dengan akurasi data latih 83,61% dan akurasi data validasi 81,17% pada skenario klasifikasi 24 kelas yang berasal dari 8 spesies ikan. Selisih performa antara data latih dan data validasi yang relatif kecil menunjukkan bahwa dari sisi kinerja, model tidak mengalami overfitting yang signifikan meskipun dilatih tanpa penerapan augmentasi data, sehingga EfficientNet-B0 mampu menjalankan fungsi klasifikasi kesegaran ikan secara konsisten pada dataset multi-spesies.
2. Optimasi hiperparameter berbasis Optuna berpengaruh positif terhadap stabilitas pelatihan dan kemampuan generalisasi model EfficientNet-B0. Pengaruh tersebut ditunjukkan melalui penurunan selisih performa antara data latih dan data validasi dengan nilai generalization gap sebesar  $\pm 2,45\%$ , yang mencerminkan peningkatan stabilitas proses pembelajaran. Selain itu,

pada evaluasi generalisasi lintas spesies menggunakan skema leave-source-out (LSO), model yang dioptimasi menggunakan Optuna tetap mampu melakukan klasifikasi pada spesies yang tidak dilibatkan dalam proses pelatihan, dengan akurasi pengujian berada pada rentang 44,58% hingga 70,40% serta nilai macro F1-score pada rentang 0,3784 hingga 0,6864. Konsistensi performa yang diperoleh dari pelatihan ulang model terbaik sebanyak lima kali semakin memperkuat bahwa Optuna memberikan pengaruh dalam menghasilkan konfigurasi hiperparameter yang stabil dan mendukung kemampuan generalisasi lintas spesies.

## 5.2 Saran

Berdasarkan hasil penelitian, beberapa saran yang dapat diberikan adalah sebagai berikut:

1. Penelitian selanjutnya dapat menguji penerapan augmentasi data atau teknik penyeimbangan kelas untuk mengevaluasi potensi peningkatan akurasi pada kelas minoritas.
2. Varian EfficientNet lain atau arsitektur CNN ringan dapat dibandingkan untuk mengkaji trade-off antara akurasi dan kompleksitas komputasi.
3. Model dengan akurasi 79,87% dan generalisasi stabil ini dapat dikembangkan ke dalam sistem real-time atau aplikasi berbasis perangkat bergerak.
4. Penelitian lanjutan dapat menggabungkan citra mata dengan indikator visual lain seperti insang guna meningkatkan performa klasifikasi.

## DAFTAR PUSTAKA

- [1] M. Hasan, N. Vasker, M. M. Hossain, M. I. Bhuiyan, J. Biswas, and M. R. Ahmmad Rashid, "Framework for fish freshness detection and rotten fish removal in Bangladesh using mask R-CNN method with robotic arm and fisheye analysis," *Journal of Agriculture and Food Research*, vol. 16, p. 101139, Jun. 2024, doi: 10.1016/j.jafr.2024.101139.
- [2] M. N. Madhubhashini, C. P. Liyanage, A. U. Alahakoon, and R. P. Liyanage, "Current applications and future trends of artificial senses in fish freshness determination: A review," *Journal of Food Science*, vol. 89, no. 1, Art. no. 1, Jan. 2024, doi: 10.1111/1750-3841.16865.
- [3] D. F. Anas, I. Jaya, and Nurjanah, "Design and implementation of fish freshness detection algorithm using deep learning," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 944, no. 1, p. 012007, Dec. 2021, doi: 10.1088/1755-1315/944/1/012007.
- [4] P. G. Cardoso, O. Gonçalves, M. F. Carvalho, R. Ozório, and P. Vaz-Pires, "Seasonal Evaluation of Freshness Profile of Commercially Important Fish Species," *Foods*, vol. 10, no. 7, Art. no. 7, Jul. 2021, doi: 10.3390/foods10071567.
- [5] J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *MAKE*, vol. 5, no. 4, Art. no. 4, Nov. 2023, doi: 10.3390/make5040083.
- [6] E. Prasetyo, R. Purbaningtyas, R. D. Adityo, N. Suciati, and C. Fatichah, "Combining MobileNetV1 and Depthwise Separable convolution bottleneck with Expansion for classifying the freshness of fish eyes," *Information Processing in Agriculture*, vol. 9, no. 4, Art. no. 4, Dec. 2022, doi: 10.1016/j.inpa.2022.01.002.
- [7] X. Guo, S. Gu, and D. Wei, "Fish freshness evaluation based on MobileNetV1 and attention mechanism," in *Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering*, Xiamen China: ACM, Oct. 2022, pp. 270–276, doi: 10.1145/3573428.3573475.
- [8] M. B. Yildiz, E. T. Yasin, and M. Koklu, "Fisheye freshness detection using common deep learning algorithms and machine learning methods with a developed mobile application," *Eur Food Res Technol*, vol. 250, no. 7, Art. no. 7, Jul. 2024, doi: 10.1007/s00217-024-04493-0.
- [9] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," 2019.
- [10] P. Pokhrel and A. Lazar, "A Comparison of AutoML Hyperparameter Optimization Tools For Tabular Data," *FLAIRS*, vol. 36, May 2023, doi: 10.32473/flairs.36.133357.
- [11] T. Leinonen *et al.*, "Empirical investigation of multi-source cross-validation in clinical ECG classification," *Computers in Biology and Medicine*, vol. 183, p. 109271, Dec. 2024, doi: 10.1016/j.compbiomed.2024.109271.
- [12] E. Helmud, "Image-Based Fish Freshness Classification Using Two-Phase Transfer Learning with Deep Learning Fusion Model," *J. Appl. Data Sci.*, vol. 6, no. 4, pp. 2974–2988, Sep. 2025, doi: 10.47738/jads.v6i4.988.

- [13] C. Balm, N. Olgun, and M. Çalışan, "Leveraging Feature Fusion of Image Features and Laser Reflectance for Automated Fish Freshness Classification," *Sensors*, vol. 25, no. 14, p. 4374, Jul. 2025, doi: 10.3390/s25144374.
- [14] H. Almarzooq and U. Bin Waheed, "Automating hyperparameter optimization in geophysics with Optuna: A comparative study," *Geophysical Prospecting*, vol. 72, no. 5, pp. 1778–1788, Jun. 2024, doi: 10.1111/1365-2478.13484.
- [15] R. M. Munshi *et al.*, "Optimising hyperparameters with a tree structured Parzen estimator to improve diabetes prediction," *Sci Rep*, vol. 15, no. 1, p. 35430, Oct. 2025, doi: 10.1038/s41598-025-19295-x.
- [16] T. Zheng *et al.*, "A video object segmentation-based fish individual recognition method for underwater complex environments," *Ecological Informatics*, vol. 82, p. 102689, Sep. 2024, doi: 10.1016/j.ecoinf.2024.102689.
- [17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," 2014.
- [18] S. Sathyanarayanan, "Confusion Matrix-Based Performance Evaluation Metrics," *AJBR*, pp. 4023–4031, Nov. 2024, doi: 10.53555/AJBR.v27i4S.4345.
- [19] H. Gao *et al.*, "Multi-modal denoised data-driven milling chatter detection using an optimized hybrid neural network architecture," *Sci Rep*, vol. 15, no. 1, p. 3953, Jan. 2025, doi: 10.1038/s41598-025-88242-7.
- [20] E. Prasetyo, R. D. Adityo, N. Suciati, and C. Fatichah, "The Freshness of the Fish Eyes Dataset," vol. 1, Feb. 2022, doi: 10.17632/xzyx7pbr3w.1.
- [21] L. Zhao *et al.*, "A Survey on Class Imbalance Learning Algorithms in Complex Scenarios," *IEEE Access*, vol. 13, pp. 180799–180833, 2025, doi: 10.1109/ACCESS.2025.3618909.
- [22] J. Miller *et al.*, "Accuracy on the Line: On the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization".
- [23] I. A. Reshma, S. Cussat-Blanc, R. T. Ionescu, H. Luga, and J. Mothe, "Natural vs balanced distribution in deep learning on whole slide images for cancer detection," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, Virtual Event Republic of Korea: ACM, Mar. 2021, pp. 18–25. doi: 10.1145/3412841.3441884.
- [24] R. J. May, H. R. Maier, and G. C. Dandy, "Data splitting for artificial neural networks using SOM-based stratified sampling," *Neural Networks*, vol. 23, no. 2, pp. 283–294, Mar. 2010, doi: 10.1016/j.neunet.2009.11.009.
- [25] A. M. Khani, A. Mohaghar, A. Rezasoltani, and S. H. Hosseinian, "Advanced Hyperparameter Optimization and Adaptive Synthetic Sampling in Machine Learning for Predictive Maintenance of Industrial Machinery," *Int. J. Res. Ind. Eng.*, no. Online First, Sep. 2025, doi: 10.22105/riej.2025.500994.1528.
- [26] F. D'Angelo, M. Andriushchenko, A. Varre, and N. Flammarion, "Why Do We Need Weight Decay in Modern Deep Learning?," Nov. 04, 2024, *arXiv: arXiv:2310.04415*. doi: 10.48550/arXiv.2310.04415.
- [27] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," Jan. 04, 2019, *arXiv: arXiv:1711.05101*. doi: 10.48550/arXiv.1711.05101.
- [28] F. Mori and F. Mignacco, "Analytic theory of dropout regularization," *Phys. Rev. E*, vol. 112, no. 4, p. 045301, Oct. 2025, doi: 10.1103/jmdx-x3gr.

- [29] G. Xia, O. Laurent, G. Franchi, and C.-S. Bouganis, "Towards Understanding Why Label Smoothing Degrades Selective Classification and How to Fix It," Feb. 20, 2025, *arXiv*: arXiv:2403.14715. doi: 10.48550/arXiv.2403.14715.
- [30] L. Prechelt, "2 Early Stopping — But When?," Jan. 2012.
- [31] A. Vardasbi, M. De Rijke, and M. Dehghani, "Intersection of Parallels as an Early Stopping Criterion," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, Atlanta GA USA: ACM, Oct. 2022, pp. 1965–1974. doi: 10.1145/3511808.3557366.
- [32] B. A. Dada, N. I. Nwulu, and S. O. Olukanmi, "Bayesian optimization with Optuna for enhanced soil nutrient prediction: a comparative study with genetic algorithm and particle swarm optimization," *Smart Agricultural Technology*, vol. 12, p. 101136, Dec. 2025, doi: 10.1016/j.atech.2025.101136.
- [33] J. P. Gygi, S. H. Kleinstein, and L. Guan, "Predictive overfitting in immunological applications: Pitfalls and solutions," *Human Vaccines & Immunotherapeutics*, vol. 19, no. 2, p. 2251830, Aug. 2023, doi: 10.1080/21645515.2023.2251830.
- [34] V. C. Araujo Santos, L. Cardoso, and R. Alves, "The quest for the reliability of machine learning models in binary classification on tabular data," *Sci Rep*, vol. 13, no. 1, p. 18464, Oct. 2023, doi: 10.1038/s41598-023-45876-9.
- [35] Y. Tian and Y. Zhang, "A comprehensive survey on regularization strategies in machine learning," *Information Fusion*, vol. 80, pp. 146–166, Apr. 2022, doi: 10.1016/j.inffus.2021.11.005.



## LAMPIRAN

Lampiran 1 Data Percobaan Pertama 100 Trial

Trial	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
75	AdamW	0,0001	0,0001	0,5	16	0	0,1	5	80,717 48879	84,415 21304	3,6977 24251	78,868 62666	0,5	11	0,7983 4046	1,1233 4453
52	Adam	0,0001	0,0001	0,3	16	0	0,05	5	84,977 57848	98,055 47612	13,077 89765	78,438 62965	0,5	19	0,4276 70865	0,9331 55346
14	Adam	0,0001	0,0001	0,3	16	0	0	5	84,080 71749	95,996 56849	11,915 851	78,122 79199	0,5	15	0,0752 43522	0,6337 25073
15	Adam	0,0001	0,0001	0,3	16	0	0,1	5	83,632 287	95,138 69031	11,506 40331	77,879 08534	0,5	14	0,7229 836	1,1017 08106
12	Adam	0,0001	0,0001	0,3	16	0	0	5	81,838 56502	89,819 84558	7,9812 8056	77,847 92474	0,5	12	0,1162 75044	0,6082 22735
85	AdamW	0,0001	0,0001	0,3	16	0	0	5	81,838 56502	89,991 42122	8,1528 56196	77,762 13692	0,5	12	0,1260 60711	0,5949 13475
77	AdamW	0,0001	0,0001	0,3	16	0	0	5	83,856 50224	96,168 14412	12,311 64188	77,700 6813	0,5	16	0,0691 96641	0,5502 10511
70	AdamW	0,0001	0,0001	0,5	16	0	0,1	5	82,286 99552	91,507 00601	9,2200 10489	77,676 99027	0,5	13	0,7539 58602	1,1389 18904
41	Adam	0,0001	0,0001	0,3	16	0	0	5	84,080 71749	97,340 57764	13,259 86015	77,450 78741	0,5	18	0,0666 93661	0,6229 53959
44	Adam	0,0001	0,0001	0,3	16	0	0	5	84,080 71749	97,626 53703	13,545 81954	77,307 80772	0,5	20	0,0476 80373	0,6039 20985
19	Adam	0,0001	0,0001	0,5	16	0	0,1	5	83,632 287	96,883 04261	13,250 75561	77,006 90919	0,5	16	0,7274 90669	1,1125 34383
13	Adam	0,0001	0,0001	0,3	16	0	0	5	82,062 78027	92,307 69231	10,244 91204	76,940 32425	0,5	13	0,0992 63268	0,6336 76289
16	Adam	0,0001	0,0001	0,3	16	0	0,1	5	84,080 71749	98,884 75836	14,804 04088	76,678 69705	0,5	23	0,6797 70342	1,1303 11579

Trail	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
18	Adam	0,0001	0,0001	0,3	16	0	0,1	5	82,511 21076	94,881 32685	12,370 11609	76,326 15272	0,5	14	0,7284 76759	1,1400 59554
92	Adam	0,0001	0,0001	0,3	16	0	0	5	82,735 42601	95,824 99285	13,089 56684	76,190 64259	0,5	16	0,0821 18819	0,6512 56691
96	AdamW	0,0001	0,0001	0,3	16	0	0	5	83,183 8565	97,540 74921	14,356 89271	76,005 41015	0,5	18	0,0684 89324	0,6858 98443
21	Adam	0,0001	0,0001	0,3	16	0	0	5	83,408 07175	98,513 01115	15,104 9394	75,855 60205	0,5	22	0,0574 27852	0,7414 70113
10	Adam	0,0001	0,0001	0,3	16	0	0	5	82,959 64126	97,483 55733	14,523 91608	75,697 68322	0,5	18	0,0553 15675	0,6400 08949
31	Adam	0,0001	0,0001	0,5	16	0	0,1	5	81,165 91928	92,136 11667	10,970 19739	75,680 82059	0,5	13	0,7597 33475	1,2254 43585
55	Adam	0,0001	0,0001	0,3	16	0	0,05	5	83,183 8565	98,455 81927	15,271 96277	75,547 87512	0,5	20	0,4142 29052	0,9539 87578
43	Adam	0,0001	0,0001	0,3	16	0	0	5	82,511 21076	96,454 10352	13,942 89275	75,539 76438	0,5	16	0,0845 68909	0,6620 1851
73	AdamW	0,0001	0,0001	0,5	16	0	0,1	5	82,735 42601	97,226 19388	14,490 76787	75,490 04207	0,5	17	0,7263 45529	1,1050 08875
24	Adam	0,0001	0,0001	0,3	16	0	0	5	82,735 42601	98,741 77867	16,006 35266	74,732 24968	0,5	27	0,0359 85872	0,7462 84778
23	Adam	0,0001	0,0001	0,3	16	0	0,05	5	80,941 70404	93,480 12582	12,538 42179	74,672 49314	0,5	13	0,4658 41395	0,8996 47688
49	Adam	0,0001	1,00E-05	0,3	16	0	0	3	81,838 56502	97,169 002	15,330 43698	74,173 34653	0,5	16	0,0693 92494	0,6909 54351
25	Adam	0,0001	1,00E-05	0,3	16	0	0,1	5	82,062 78027	98,341 43552	16,278 65525	73,923 45265	0,5	18	0,6950 11482	1,1451 84666
11	Adam	0,0001	0,0001	0,3	16	0	0	5	81,165 91928	98,141 26394	16,975 34466	72,678 24695	0,5	20	0,0515 01583	0,8161 58226

Trial	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
4	Adam	0,0001	0,01	0,3	16	0	0	5	78,699 55157	91,221 04661	12,521 49504	72,438 80405	0,5	16	0,1907 78935	0,7834 52185
8	AdamW	0,001	0,001	0,5	16	1	0,1	3	59,417 04036	59,422 36202	0,0053 21666	59,414 37953	0,5	14	1,6562 20186	1,6494 71704
3	AdamW	0,001	0,0005	0,7	16	2	0	3	59,192 82511	48,498 71318	10,694 11193	53,845 76915	0,5	13	1,4779 07172	1,2793 26682
5	SGD	0,001	0,001	0,3	16	2	0	3	55,156 95067	60,451 81584	5,2948 6517	52,509 51809	0,5	23	1,3478 78185	1,4096 85888
2	Adam	0,0001	0,001	0,3	16	2	0	3	54,260 08969	58,364 31227	4,1042 22582	52,207 9784	0,5	26	1,4685 92533	1,5454 70297
1	AdamW	0,0001	1,00E-05	0,7	16	2	0,05	3	53,587 44395	49,156 41979	4,4310 24158	51,371 93187	0,5	33	1,8425 93718	1,7901 27988
9	AdamW	0,001	0,01	0,7	16	2	0,05	3	57,174 88789	45,467 54361	11,707 34428	51,321 21575	0,5	9	1,7236 60881	1,5721 0187
0	SGD	0,0001	1,00E-05	0,5	16	1	0,1	5	47,309 41704	44,266 51415	3,0429 02885	45,787 9656	0,5	50	2,2652 37175	2,2654 8346

Lampiran 2 Data Percobaan Kedua 100 Trial

Trial	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
7	AdamW	0,0001	0,0001	0,7	16	0	0	3	81,838 56502	82,299 11353	0,4605 48503	81,608 29077	0,5	10	0,2921 3317	0,5301 0423
95	AdamW	0,0001	0,001	0,5	16	0	0,1	5	83,856 50224	92,793 82328	8,9373 21035	79,387 84172	0,5	14	0,7552 09682	1,0783 17527
72	AdamW	0,0001	1,00E-05	0,5	16	0	0,1	5	82,286 99552	88,790 39176	6,5033 96249	79,035 29739	0,5	12	0,7912 5041	1,1037 69656

Trial	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
									82,062	88,647	6,5846	78,770			0,7708	1,1066
94	AdamW	0,0001	0,001	0,5	16	0	0,1	5	78027	41207	31798	46437	0,5	12	26518	34732
13	AdamW	0,0001	0,0001	0,7	16	0	0,1	3	79,820	82,585	2,7644	78,438	0,5	10	0,9141	1,1274
									6278	07292	45117	40524			53349	45894
24	AdamW	0,0001	1,00E-05	0,7	16	0	0,1	3	81,390	87,846	6,4565	78,161	0,5	11	0,8627	1,1353
									13453	72576	91236	83891			84001	74005
14	AdamW	0,0001	0,0001	0,7	16	0	0,1	3	83,856	96,311	12,454	77,629	0,5	17	0,7585	1,1390
									50224	12382	62158	19145			55245	95187
58	AdamW	0,0001	1,00E-05	0,5	16	0	0,1	3	83,632	95,653	12,021	77,621	0,5	14	0,7467	1,1525
									287	41721	13022	72189			94129	88597
84	AdamW	0,0001	0,0001	0,5	16	0	0,1	5	84,529	98,570	14,041	77,508	0,5	23	0,6951	1,0934
									14798	20303	05505	62046			49559	76623
17	AdamW	0,0001	0,0001	0,7	16	0	0,1	3	81,614	90,506	8,8917	77,168	0,5	12	0,8438	1,1402
									34978	14813	98351	4506			11892	60407
79	AdamW	0,0001	0,001	0,5	16	0	0,1	5	82,959	94,766	11,807	77,055	0,5	14	0,7340	1,1459
									64126	94309	30184	99034			76314	0502
68	AdamW	0,0001	1,00E-05	0,5	16	0	0,1	5	84,304	98,913	14,608	77,000	0,5	27	0,6808	1,1190
									93274	3543	42157	72195			94511	14359
38	Adam	0,0001	0,0005	0,7	16	0	0	3	80,044	73,177	6,8678	76,610	0,5	8	0,4187	0,5577
									84305	00886	34185	92596			80498	39895
18	Adam	0,0001	0,0001	0,3	16	0	0	5	83,856	98,655	14,799	76,456	0,5	25	0,0486	0,6930
									50224	99085	48861	75794			33295	71831
12	AdamW	0,0001	0,0001	0,7	16	0	0,1	3	82,286	93,966	11,679	76,447	0,5	14	0,7913	1,1312
									99552	25679	26128	36488			03804	58707
11	AdamW	0,0001	0,0001	0,7	16	0	0,05	3	81,614	92,307	10,693	76,267	0,5	13	0,5375	0,9553
									34978	69231	34253	67851			503	90245
89	AdamW	0,0001	0,0001	0,5	16	0	0,05	3	82,735	95,939	13,203	76,133	0,5	14	0,4792	0,8703
									42601	37661	9506	45071			94422	78437

Trail	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
78	AdamW	0,0001	1,00E-05	0,3	16	0	0,05	5	82,511 21076	95,367 45782	12,856 24706	76,083 08723	0,5	14	0,4551 66637	0,8934 70649
46	AdamW	0,0001	0,0001	0,3	16	0	0,1	3	81,838 56502	94,052 04461	12,213 47959	75,731 82523	0,5	11	0,7667 49373	1,1363 73618
4	AdamW	0,0001	0,0005	0,7	16	0	0,05	3	82,286 99552	95,796 39691	13,509 4014	75,532 29482	0,5	16	0,4906 19068	0,8617 5876
55	Adam	0,0001	0,0005	0,7	16	0	0	3	80,717 48879	91,735 77352	11,018 28473	75,208 34642	0,5	13	0,1661 18155	0,5958 06317
91	AdamW	0,0001	0,0001	0,5	16	0	0,1	5	82,735 42601	98,026 88018	15,291 45417	75,089 69892	0,5	18	0,7128 96758	1,0900 52485
69	AdamW	0,0001	1,00E-05	0,5	16	0	0,1	5	82,286 99552	96,797 25479	14,510 25927	75,031 86588	0,5	17	0,7186 80551	1,0904 90369
73	AdamW	0,0001	1,00E-05	0,5	16	0	0,1	5	82,286 99552	97,369 17358	15,082 17806	74,745 90648	0,5	19	0,7076 31296	1,1574 14341
61	AdamW	0,0001	1,00E-05	0,5	16	0	0,1	3	82,062 78027	96,968 83043	14,906 05016	74,609 75519	0,5	15	0,7420 19037	1,1395 28653
10	Adam	0,0001	0,0001	0,7	16	0	0,1	3	80,269 0583	93,680 2974	13,411 2391	73,563 43875	0,5	14	0,7927 24	1,1273 92298
93	AdamW	0,0001	0,0001	0,5	16	0	0,1	5	81,390 13453	97,569 34515	16,179 21062	73,300 52922	0,5	18	0,7120 05939	1,1694 25524
6	AdamW	0,001	0,0005	0,3	16	1	0	5	61,210 76233	65,656 27681	4,4455 14477	58,988 00509	0,5	16	0,9471 92545	1,1412 31537
1	Adam	0,001	0,01	0,3	16	1	0	3	59,417 04036	61,052 33057	1,6352 9021	58,599 39525	0,5	13	1,3191 98491	1,4010 58844
3	SGD	0,001	1,00E-05	0,5	16	2	0	5	57,847 53363	56,219 61681	1,6279 16818	57,033 57522	0,5	25	1,3957 60298	1,3873 16578
5	AdamW	0,001	0,0005	0,7	16	1	0	3	59,192 82511	47,440 66343	11,752 16169	53,316 74427	0,5	12	1,4951 97107	1,2950 38579

Trial	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
0	AdamW	0,0001	0,001	0,7	16	1	0	3	52,242 15247	45,381 75579	6,8603 96676	48,811 95413	0,5	23	1,8635 42392	1,8004 30864
2	SGD	0,0001	1,00E-05	0,7	16	1	0	3	47,757 84753	40,835 00143	6,9228 46104	44,296 42448	0,5	50	2,1067 68708	2,0601 63766

Lampiran 3 Data Percobaan Ketiga 100 Trial

Trial	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
52	AdamW	0,0001	0,0005	0,7	16	0	0,1	5	84,304 93274	94,052 04461	9,7471 11874	79,431 3768	0,5	16	0,7692 16485	1,1017 10254
57	AdamW	0,0001	0,0005	0,3	16	0	0,1	5	84,304 93274	94,309 40806	10,004 47533	79,302 69507	0,5	13	0,7323 84215	1,1005 36429
55	AdamW	0,0001	0,0005	0,7	16	0	0,1	5	80,717 48879	77,523 59165	3,1938 97139	79,120 54022	0,5	11	0,8684 68906	1,0895 72291
54	AdamW	0,0001	0,0005	0,7	16	0	0,1	5	80,493 27354	83,528 73892	3,0354 65376	78,975 54085	0,5	12	0,8484 22242	1,1465 25042
13	AdamW	0,001	0,0001	0,7	16	0	0,05	5	81,390 13453	86,273 9491	4,8838 1457	78,948 22724	0,5	11	0,5528 63511	1,0096 63916
76	Adam	0,0001	0,0005	0,5	16	0	0,1	3	80,941 70404	84,987 13183	4,0454 27791	78,918 99014	0,5	9	0,8573 66217	1,0918 02561
70	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	84,304 93274	95,396 05376	11,091 12102	78,759 37222	0,5	12	0,7391 67735	1,1082 26433
92	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	84,304 93274	95,596 22534	11,291 2926	78,659 28644	0,5	12	0,7426 68203	1,0870 69167
59	AdamW	0,0001	0,0005	0,3	16	0	0,1	5	84,304 93274	96,368 3157	12,063 38296	78,273 24125	0,5	15	0,7069 52314	1,1116 7258

Trial	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
									83,856	95,424	11,568	78,072			0,7437	1,1041
94	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	50224	6497	14746	42851	0,5	12	9032	57314
72	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	84,529	97,969	13,440	77,808	0,5	16	0,7049	1,0590
									14798	6883	54032	87782			9362	87247
67	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	83,632	95,281	11,649	77,807	0,5	12	0,7449	1,1780
									287	67	38301	59549			89397	46512
87	Adam	0,0001	0,001	0,7	16	0	0,1	3	81,614	89,619	8,0053	77,611	0,5	12	0,8340	1,1384
									34978	67401	24231	68766			75626	0071
91	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	82,280	91,764	9,4773	77,548	0,5	10	0,7775	1,1058
									99552	36946	73944	30854			99325	14631
50	AdamW	0,0001	0,0005	0,7	16	0	0,1	5	84,529	98,798	14,269	77,394	0,5	26	0,7031	1,0932
									14798	97055	82256	2367			0179	05967
8	SGD	0,001	0,0001	0,7	16	0	0,1	5	82,511	93,194	10,682	77,169	0,5	32	0,8768	1,1185
									21076	16643	95567	73293			60092	79733
21	AdamW	0,001	0,0001	0,7	16	0	0	5	78,251	80,640	2,3894	77,056	0,5	10	0,2545	0,8408
									12108	54904	27966	40709			46949	02182
82	Adam	0,0001	0,0005	0,5	16	0	0,1	3	82,959	94,909	11,950	76,984	0,5	12	0,7687	1,0766
									64126	92279	28154	50049			99704	36529
71	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	79,372	84,157	4,7856	76,979	0,5	8	0,8575	1,1326
									19731	84959	52276	37117			66384	48132
74	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	81,838	91,850	10,011	76,832	0,5	10	0,7795	1,1475
									56502	15728	59226	76889			16636	23867
86	AdamW	0,0001	0,0005	0,3	16	0	0,1	5	83,856	98,655	14,799	76,456	0,5	24	0,6756	1,1400
									50224	99085	48861	75794			23939	64627
60	AdamW	0,0001	0,0005	0,3	16	0	0,1	5	83,856	98,941	15,085	76,313	0,5	22	0,6866	1,1095
									50224	95024	448	77824			05671	75974
61	AdamW	0,0001	0,0005	0,3	16	0	0,1	5	83,856	98,970	15,114	76,299	0,5	22	0,6815	1,1379
									50224	54618	04394	48027			70099	34904

Trial	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
									80,269 0583	88,704 60395	8,4355 4565	76,051 28547			0.8171 58031	1,1567 72139
83	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	81,838 56502	93,766 08522	11,927 52019	75,874 80493	0,5	9	0,7279 74402	1,0923 13213
62	AdamW	0,0001	0,0005	0,3	16	0	0,1	5	83,183 8565	97,826 70861	14,642 85211	75,862 43045	0,5	20	0,7040 69052	1,1273 81465
80	Adam	0,0001	0,0005	0,5	16	0	0,1	5	83,408 07175	98,598 79897	15,190 72722	75,812 70814	0,5	27	0,6977 31299	1,1270 64962
53	AdamW	0,0001	0,0005	0,7	16	0	0,1	5	81,614 34978	93,451 52988	11,837 18011	75,695 75972	0,5	11	0,7599 1075	1,0906 10838
84	AdamW	0,0001	1,00E-05	0,3	16	0	0,1	3	79,372 19731	87,446 38261	8,0741 85304	75,335 10466	0,5	13	0,8176 91894	1,1246 25996
56	Adam	0,0001	0,0005	0,7	16	0	0,1	5	81,838 56502	94,909 92279	13,071 35777	75,302 88614	0,5	17	0,7606 84393	1,1296 73788
51	AdamW	0,0001	0,0005	0,7	16	0	0,1	5	81,838 81,838	94,995 94,995	13,157 13,157	75,259 75,259	0,5	12	0,7511 15334	1,1148 48971
96	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	56502 81,614	71061 94,452	14559 12,838	99223 75,195	0,5	13	0,7371 84073	1,1362 30571
64	AdamW	0,0001	0,0005	0,3	16	0	0,1	5	34978 76,905	38776 80,354	03799 3,4487	33078 75,181	0,5	10	0,8073 74525	30571 76813
14	AdamW	0,001	0,0001	0,7	16	0	0	5	8296 78,026	58965 84,157	60052 6,1309	44957 74,961	0,5	11	0,5299 5607	0,9551 15201
42	AdamW	0,001	0,0001	0,7	16	0	0,05	5	90583 82,511	84959 97,712	43756 15,201	43395 74,910	0,5	11	0,5607 59493	1,5201 01362
90	AdamW	0,0001	0,01	0,3	16	0	0,1	3	21076 82,735	32485 98,455	11409 15,720	65372 74,875	0,5	13	0,7273 08614	1,1148 62466
81	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	42601 82,062	81927 96,597	39326 14,534	22938 74,795	0,5	16	0,7080 70778	1,1050 47789
93	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	78027 82,062	08321 96,597	30295 14,534	6288 74,795	0,5	13	0,7078 70778	47789 47789

Trail	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
69	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	81,390 13453	94,909 92279	13,519 78826	74,630 2404	0,5	12	0,7451 76184	1,1046 36727
73	AdamW	0,0001	0,0005	0,3	16	0	0,1	3	80,941 70404	93,680 2974	12,738 59336	74,572 40735	0,5	11	0,7547 65732	1,1515 8295
63	AdamW	0,0001	0,0005	0,3	16	0	0,1	5	82,062 78027	97,054 61824	14,991 83798	74,566 86128	0,5	16	0,7018 73211	1,1416 62402
11	AdamW	0,001	0,0001	0,7	16	0	0,05	5	79,147 98206	89,104 9471	9,9569 65035	74,169 49955	0,5	12	0,5448 69109	1,1599 11434
45	AdamW	0,0001	0,0005	0,7	16	0	0,05	5	81,838 56502	97,197 59794	15,359 03292	74,159 04856	0,5	19	0,4583 77851	0,9460 57366
10	AdamW	0,001	0,0001	0,7	16	0	0,05	5	74,887 89238	76,894 48098	2,0065 88607	73,884 59807	0,5	9	0,5943 7845	1,2251 27109
77	Adam	0,0001	0,0005	0,5	16	0	0,1	3	80,717 48879	94,766 94309	14,049 4543	73,692 76164	0,5	12	0,7628 8649	1,1326 44206
75	AdamW	0,0001	0,001	0,3	16	0	0,1	3	81,838 56502	98,684 58679	16,846 02177	73,415 55414	0,5	19	0,6921 71686	1,0977 77018
65	AdamW	0,0001	0,0005	0,3	16	0	0,1	5	81,838 56502	98,770 37461	16,931 80958	73,372 66023	0,5	20	0,6845 31674	1,1453 52134
58	AdamW	0,0001	0,0005	0,3	16	0	0,1	5	81,390 13453	98,112 668	16,722 53347	73,028 86779	0,5	17	0,6994 01815	1,1602 74802
41	AdamW	0,001	0,0001	0,7	16	0	0,05	5	80,941 70404	97,026 0223	16,084 31827	72,899 5449	0,5	19	0,4880 94251	1,0899 05024
16	AdamW	0,001	0,001	0,7	16	0	0,05	5	80,269 0583	95,024 30655	14,755 24825	72,891 43417	0,5	18	0,4760 56173	1,3687 62368
43	AdamW	0,001	0,0001	0,7	16	0	0,05	5	78,026 90583	88,876 17958	10,849 27375	72,602 26895	0,5	12	0,5286 5336	1,0334 36562
12	AdamW	0,001	0,001	0,7	16	0	0,05	5	79,596 41256	93,823 27709	14,226 86454	72,482 98029	0,5	15	0,4939 04101	1,0299 29892

trial	Optimizer	Learning Rate	Weight Decay	Dropout Rate	Batch Size	Freeze Mode	Label Smoothing	Patience	Best Val Accuracy	Best Train Accuracy	Train-Val Gap	Gap Penalty Score	Gap Penalty Alpha	Total Epochs	Final Train Loss	Final Val Loss
23	AdamW	0,001	0,0001	0,7	16	0	0	5	79,372 19731	93,094 85273	14,622 65542	72,060 8696	0,5	19	0,1639 19934	1,8853 27945
18	AdamW	0,001	0,0001	0,7	16	0	0,1	5	77,578 47534	90,020 01716	12,441 54182	71,357 70443	0,5	12	0,8075 2304	1,2437 86267
19	Adam	0,001	0,0001	0,7	16	0	0	5	76,905 8296	88,790 39176	11,884 56217	70,963 54851	0,5	14	0,1618 50032	1,0302 13065
22	AdamW	0,001	0,0001	0,7	16	0	0	5	76,008 96861	89,819 84558	13,810 87697	69,103 53012	0,5	14	0,2241 59523	1,2150 38178
24	AdamW	0,001	0,0001	0,7	16	0	0	5	75,560 53812	89,333 71461	13,773 1765	68,673 94987	0,5	14	0,1853 84255	1,1139 32983
32	AdamW	0,001	0,0001	0,7	16	0	0	5	76,681 61435	93,094 85273	17,313 23838	68,024 99516	0,5	17	0,1776 09939	1,3556 34951
3	AdamW	0,001	1,00E-05	0,5	16	2	0,1	3	56,726 4574	56,305 40463	0,4210 52767	56,515 93102	0,5	10	1,6781 04207	1,6679 6333
2	SGD	0,001	0,0005	0,5	16	2	0	3	56,502 24215	56,505 57621	0,0033 34056	56,500 57512	0,5	23	1,4081 90981	1,4304 66597
4	Adam	0,0001	0,01	0,5	16	1	0,1	5	56,278 02691	59,136 40263	2,8583 75725	54,848 83904	0,5	47	1,8206 0791	1,8551 03459
1	SGD	0,001	0,0001	0,3	16	1	0,1	3	56,278 02691	61,223 90621	4,9458 79299	53,805 08726	0,5	22	1,7209 4432	1,7742 10806
0	Adam	0,0001	0,01	0,5	16	1	0,05	3	52,914 79821	48,813 26852	4,1015 2969	50,864 03336	0,5	18	1,9695 26428	1,9622 67199

Lampiran 4 Data Kesalahan Klasifikasi

image_path	ground truth	predicted	ground truth idx	predicted idx	confidence
Chanos Chanos - Fresh/IMG_20191003_065053.jpg	Chanos Chanos - Fresh	Chanos Chanos - Not Fresh	0	2	0.6009

image_path	ground truth	predicted	ground_truth_idx	predicted_idx	confidence
Chanos Chanos - Fresh/IMG_20191003_064720.jpg	Chanos Chanos - Fresh	Chanos Chanos - Not Fresh	0	2	0.4444
Chanos Chanos - Fresh/IMG_20191002_062711.jpg	Chanos Chanos - Fresh	Chanos Chanos - Highly Fresh	0	1	0.8411
Chanos Chanos - Fresh/IMG_20191112_064326.jpg	Chanos Chanos - Fresh	Chanos Chanos - Highly Fresh	0	1	0.8850
Chanos Chanos - Fresh/IMG_20191003_064420.jpg	Chanos Chanos - Fresh	Chanos Chanos - Not Fresh	0	2	0.6112
Chanos Chanos - Fresh/IMG_20191003_064459.jpg	Chanos Chanos - Fresh	Chanos Chanos - Not Fresh	0	2	0.7020
Chanos Chanos - Not Fresh/IMG_20191114_054148.jpg	Chanos Chanos - Not Fresh	Chanos Chanos - Fresh	2	0	0.5330
Chanos Chanos - Not Fresh/IMG_20191004_063606.jpg	Chanos Chanos - Not Fresh	Chanos Chanos - Fresh	2	0	0.5188
Eleutheronema Tetradactylum - Fresh/IMG_20191017_070934.jpg	Eleutheronema Tetradactylum - Fresh	Eleutheronema Tetradactylum - Not Fresh	3	5	0.4536
Eleutheronema Tetradactylum - Highly Fresh/IMG_20191015_071556.jpg	Highly Fresh	Fresh	4	3	0.6967
Eleutheronema Tetradactylum - Highly Fresh/IMG_20191015_072146.jpg	Highly Fresh	Fresh	4	3	0.4525
Eleutheronema Tetradactylum - Highly Fresh/IMG_20191014_073023.jpg	Highly Fresh	Fresh	4	3	0.5108
Eleutheronema Tetradactylum - Highly Fresh/IMG_20191015_071620.jpg	Highly Fresh	Fresh	4	3	0.6609
Eleutheronema Tetradactylum - Highly Fresh/IMG_20191015_071645.jpg	Highly Fresh	Fresh	4	3	0.5708
Eleutheronema Tetradactylum - Highly Fresh/IMG_20191015_071702.jpg	Highly Fresh	Fresh	4	3	0.5527
Eleutheronema Tetradactylum - Highly Fresh/IMG_20191015_071410.jpg	Highly Fresh	Fresh	4	3	0.5617
Eleutheronema Tetradactylum - Not Fresh/IMG_20191019_063800.jpg	Not Fresh	Fresh	5	3	0.5271

image_path	ground_truth	predicted	ground_truth_idx	predicted_idx	confidence
Eleutheronema Tetradaetylum - Not Fresh/IMG_20191019_063444.jpg	Not Fresh	Fresh	5	3	0.7791
Johnius Trachycephalus - Highly Fresh/IMG_20191015_065259.jpg	Highly Fresh	Fresh	7	6	0.7299
Johnius Trachycephalus - Highly Fresh/IMG_20191015_065843.jpg	Highly Fresh	Fresh	7	6	0.6068
Johnius Trachycephalus - Not Fresh/IMG_20191018_060609.jpg	Not Fresh	Fresh	8	6	0.4321
Johnius Trachycephalus - Not Fresh/IMG_20191019_063056.jpg	Not Fresh	Fresh	8	6	0.7904
Johnius Trachycephalus - Not Fresh/IMG_20191019_061850.jpg	Not Fresh	Highly Fresh	8	7	0.6742
Johnius Trachycephalus - Not Fresh/IMG_20191019_062226.jpg	Not Fresh	Highly Fresh	8	7	0.5993
Nibeal Albiflora - Highly Fresh/IMG_20191110_055936_4.jpg	Highly Fresh	Rastrelliger Faughni - Highly Fresh	9	19	0.8644
Nibeal Albiflora - Highly Fresh/IMG_20191111_063134_4.jpg	Highly Fresh	Fresh	9	10	0.7560
Nibeal Albiflora - Highly Fresh/IMG_20191111_062947.jpg	Highly Fresh	Not Fresh	9	11	0.8001
Nibeal Albiflora - Highly Fresh/IMG_20191111_063526_3.jpg	Highly Fresh	Fresh	9	10	0.4853
Nibeal Albiflora - Highly Fresh/IMG_20191111_063309.jpg	Highly Fresh	Fresh	9	10	0.5625
Nibeal Albiflora - Fresh/IMG_20191112_064033_2.jpg	Fresh	Oreochromis Niloticus - Highly Fresh	10	16	0.2927
Nibeal Albiflora - Fresh/IMG_20191112_061729_2.jpg	Fresh	Highly Fresh	10	9	0.9306
Nibeal Albiflora - Fresh/IMG_20191113_060756_1.jpg	Fresh	Highly Fresh	10	9	0.6183
Nibeal Albiflora - Fresh/IMG_20191112_061715_2.jpg	Fresh	Highly Fresh	10	9	0.8309

image_path	ground_truth	predicted	ground_truth_idx	predicted_idx	confidence
Nibeia Albiflora - Not Fresh/IMG_20191115_062001_1.jpg	Not Fresh	Fresh	11	10	0.9440
Nibeia Albiflora - Not Fresh/IMG_20191114_062656.jpg	Not Fresh	Highly Fresh	11	9	0.5300
Oreochromis Mossambicus - Fresh/IMG_20191113_054918.jpg	Fresh	Oreochromis Niloticus - Fresh	12	15	0.4537
Oreochromis Mossambicus - Fresh/IMG_20191113_054741_2.jpg	Fresh	Highly Fresh	12	13	0.5815
Oreochromis Mossambicus - Fresh/IMG_20191112_055232_4.jpg	Fresh	Not Fresh	12	14	0.6599
Oreochromis Mossambicus - Fresh/IMG_20191112_055102_2.jpg	Fresh	Not Fresh	12	14	0.7692
Oreochromis Mossambicus - Highly Fresh/IMG_20191111_055053_2.jpg	Highly Fresh	Not Fresh	13	14	0.4545
Oreochromis Mossambicus - Highly Fresh/IMG_20191111_055530.jpg	Highly Fresh	Oreochromis Niloticus - Highly Fresh	13	16	0.7034
Oreochromis Mossambicus - Not Fresh/IMG_20191114_060547.jpg	Not Fresh	Highly Fresh	14	13	0.3746
Oreochromis Niloticus - Fresh/IMG_20191002_064233.jpg	Fresh	Not Fresh	15	17	0.9260
Oreochromis Niloticus - Fresh/IMG_20191003_063448.jpg	Fresh	Not Fresh	15	17	0.9886
Oreochromis Niloticus - Fresh/IMG_20191003_063831.jpg	Fresh	Not Fresh	15	17	0.4187
Oreochromis Niloticus - Fresh/IMG_20191112_053256.jpg	Fresh	Highly Fresh	15	16	0.5264
Oreochromis Niloticus - Fresh/IMG_20191112_053858_2.jpg	Fresh	Highly Fresh	15	16	0.6908
Oreochromis Niloticus - Fresh/IMG_20191003_063650.jpg	Fresh	Not Fresh	15	17	0.9654
Oreochromis Niloticus - Fresh/IMG_20191112_064253_1.jpg	Fresh	Not Fresh	15	17	0.5060

image_path	ground_truth	predicted	ground_truth_idx	predicted_idx	confidence
Oreochromis Niloticus - Fresh/IMG_20191112_053655.jpg	Fresh	Not Fresh	15	17	0.8790
Oreochromis Niloticus - Fresh/IMG_20191002_064643.jpg	Fresh	Not Fresh	15	17	0.8644
Oreochromis Niloticus - Fresh/IMG_20191002_063956.jpg	Fresh	Not Fresh	15	17	0.8175
Oreochromis Niloticus - Fresh/IMG_20191113_053427_1.jpg	Fresh	Oreochromis Mossambicus - Fresh	15	12	0.5950
Oreochromis Niloticus - Highly Fresh/IMG_20191111_054650_5.jpg	Highly Fresh	Fresh	16	15	0.4972
Oreochromis Niloticus - Highly Fresh/IMG_20191111_054215_2.jpg	Highly Fresh	Not Fresh	16	17	0.7414
Oreochromis Niloticus - Highly Fresh/IMG_20191110_073144_4.jpg	Highly Fresh	Oreochromis Mossambicus - Highly Fresh	16	13	0.9087
Oreochromis Niloticus - Not Fresh/IMG_20191005_070806.jpg	Not Fresh	Highly Fresh	17	16	0.6390
Oreochromis Niloticus - Not Fresh/IMG_20191005_071341.jpg	Not Fresh	Fresh	17	15	0.8215
Oreochromis Niloticus - Not Fresh/IMG_20191004_064135.jpg	Not Fresh	Fresh	17	15	0.4214
Rastrelliger Faughni - Fresh/IMG_20191113_055553_1.jpg	Fresh	Not Fresh	18	20	0.8120
Rastrelliger Faughni - Fresh/IMG_20191112_063832_3.jpg	Fresh	Highly Fresh	18	19	0.5328
Rastrelliger Faughni - Fresh/IMG_20191112_063528.jpg	Fresh	Not Fresh	18	20	0.6570
Rastrelliger Faughni - Fresh/IMG_20191113_055644_2.jpg	Fresh	Not Fresh	18	20	0.7897
Rastrelliger Faughni - Fresh/IMG_20191112_062447.jpg	Fresh	Not Fresh	18	20	0.9046

image_path	ground_truth	predicted	ground_truth_idx	predicted_idx	confidence
Rastrelliger Faughni - Fresh/IMG_20191112_063526.jpg	Fresh	Not Fresh	18	20	0.6324
Rastrelliger Faughni - Fresh/IMG_20191112_062702_2.jpg	Fresh	Not Fresh	18	20	0.7041
Rastrelliger Faughni - Fresh/IMG_20191112_062412.jpg	Fresh	Not Fresh	18	20	0.3971
Rastrelliger Faughni - Fresh/IMG_20191112_063742_2.jpg	Fresh	Not Fresh	18	20	0.7821
Rastrelliger Faughni - Highly Fresh/IMG_20191111_061929_2.jpg	Highly Fresh	Not Fresh	19	20	0.6387
Rastrelliger Faughni - Not Fresh/IMG_20191019_060907.jpg	Not Fresh	Fresh	20	18	0.9279
Rastrelliger Faughni - Not Fresh/IMG_20191019_060811.jpg	Not Fresh	Fresh	20	18	0.6567
Rastrelliger Faughni - Not Fresh/IMG_20191018_055923.jpg	Not Fresh	Fresh	20	18	0.6146
Upeneus Moluccensis - Fresh/IMG_20191112_060445_2.jpg	Fresh	Not Fresh	21	23	0.8064
Upeneus Moluccensis - Fresh/IMG_20191016_063804.jpg	Fresh	Not Fresh	21	23	0.8463
Upeneus Moluccensis - Fresh/IMG_20191112_060029_1.jpg	Fresh	Not Fresh	21	23	0.2565
Upeneus Moluccensis - Fresh/IMG_20191112_055722.jpg	Fresh	Not Fresh	21	23	0.9387
Upeneus Moluccensis - Fresh/IMG_20191112_060519_2.jpg	Fresh	Highly Fresh	21	22	0.5530
Upeneus Moluccensis - Fresh/IMG_20191112_064136_2.jpg	Fresh	Highly Fresh	21	22	0.8991
Upeneus Moluccensis - Fresh/IMG_20191112_055940_2.jpg	Fresh	Highly Fresh	21	22	0.5077
Upeneus Moluccensis - Fresh/IMG_20191017_062725.jpg	Fresh	Not Fresh	21	23	0.8148

image_path	ground_truth	predicted	ground_truth_idx	predicted_idx	confidence
Upeneus Moluccensis - Fresh/IMG_20191112_055742.jpg	Fresh	Not Fresh	21	23	0.5838
Upeneus Moluccensis - Fresh/IMG_20191112_060253.jpg	Fresh	Highly Fresh	21	22	0.3720
Upeneus Moluccensis - Highly Fresh/IMG_20191111_060625.jpg	Highly Fresh	Not Fresh	22	23	0.7233
Upeneus Moluccensis - Highly Fresh/IMG_20191015_063144.jpg	Highly Fresh	Not Fresh	22	23	0.4521
Upeneus Moluccensis - Highly Fresh/IMG_20191110_061116_3.jpg	Highly Fresh	Rastrelliger Faughni - Highly Fresh	22	19	0.4158
Upeneus Moluccensis - Highly Fresh/IMG_20191110_061419_3.jpg	Highly Fresh	Fresh	22	21	0.5161
Upeneus Moluccensis - Not Fresh/IMG_20191019_055736.jpg	Not Fresh	Fresh	23	21	0.4873
Upeneus Moluccensis - Not Fresh/IMG_20191114_063444_1.jpg	Not Fresh	Fresh	23	21	0.6695
Upeneus Moluccensis - Not Fresh/IMG_20191018_054557.jpg	Not Fresh	Fresh	23	21	0.5841
Upeneus Moluccensis - Not Fresh/IMG_20191019_055647.jpg	Not Fresh	Fresh	23	21	0.5581