KLASIFIKASI IKAN CAKALANG DAN IKAN TONGKOL MENGGUNAKAN XCEPTION DAN MOBILENET

(*Fish Classification of Skipjack and Mackerel Tuna Classification Using Xception and MobileNet*)

Febrian Rizky Anugrah[1]*, Fitri Bimantoro[1], I Gede Pasek Suta Wijaya[1]
[1]Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Email: febriabi13@gmail.com, [bimo, gpsutawijaya]@unram.ac.id

Abstract

This study compares the performance of two deep learning architectures, Xception and MobileNet, for classifying skipjack and mackerel tuna, with a focus on accuracy and computational efficiency. MobileNet achieved an impressive accuracy of 97%, with precision, recall, and F1-score all at 97%, and demonstrated a faster prediction time of 0.06 seconds, making it well-suited for real-time applications. In contrast, Xception achieved an accuracy of 93%, with a precision of 94%, recall of 93%, and an F1-score of 93%. However, its prediction time was slower at 0.13 seconds, indicating a higher computational complexity. Although Xception delivered substantial accuracy, MobileNet outperformed it in terms of efficiency, suggesting that MobileNet is better suited for applications with limited resources or time constraints. The results indicated that MobileNet's lightweight architecture makes it ideal for mobile or embedded systems. At the same time, Xception's more complex structure may be advantageous for tasks that require higher precision in image processing. This research makes a significant contribution to the development of deep learning-based methods for fish species classification, offering improvements in both accuracy and speed.

Keywords: *Fish Classification, Xception, MobileNet, Convolutional Neural Network, Deep Learning.*

**Correspondence Author*

1. PENDAHULUAN

Industri pengolahan perikanan merupakan usaha yang mengolah hasil perikanan atau organisme akuatik, baik dari hasil budidaya maupun tangkapan, dengan tujuan komersial atau industri [1]. Sektor ini menjadi salah satu sektor vital dalam perekonomian Indonesia, terutama di wilayah pesisir. Diperkirakan terdapat sekitar 3.200 spesies ikan di Indonesia, dengan distribusi sekitar 51% di perairan laut, 48% di perairan tawar, dan 1% bergerak antara perairan laut dan perairan tawar [2]. Ikan cakalang (*Katsuwonus pelamis*) dan ikan tongkol (*Euthynnus affinis*) adalah dua spesies yang memiliki nilai ekonomi tinggi dan menjadi sumber protein utama bagi masyarakat. Kedua jenis ikan ini banyak ditemui di pasar tradisional hingga supermarket modern. Meskipun di pasaran ikan tongkol dan ikan cakalang dapat dibedakan berdasarkan ukuran dan sisik, tantangan signifikan tetap ada dalam membedakan kedua spesies tersebut secara visual, terutama bagi konsumen yang kurang berpengalaman. Kesamaan bentuk dan warna tubuh ikan cakalang dan tongkol sering kali menyebabkan kebingungan dan potensi kesalahan identifikasi [3].

Permasalahan klasifikasi ikan cakalang dan tongkol menjadi penting dalam konteks otomatisasi, di mana perbedaan visual seperti ukuran dan sisik tidak selalu cukup jelas dalam pengolahan citra. Tantangan utama dalam pengolahan citra terletak pada kemiripan bentuk dan pola tekstur ikan tersebut, yang sering membuat model klasifikasi kesulitan dalam membedakan antara keduanya secara akurat. Oleh karena itu, pendekatan klasifikasi berbasis fitur visual yang lebih rinci dan mendalam diperlukan untuk meningkatkan akurasi klasifikasi otomatis dengan memanfaatkan *transfer learning* dari model terlatih yang tersedia di pustaka Keras *Applications*. Yang mana klasifikasi citra tersebut sudah pernah digunakan dalam klasifikasi berbagai spesiesikan, seperti ikan karang [4], ikan laut [5], ikan mujair, dan ikan nila [6].

Salah satu tujuan utama dari penelitian ini adalah membandingkan secara mendalam performa dua arsitektur *deep learning*, yaitu *Xception* dan *MobileNet*, dalam klasifikasi ikan cakalang dan tongkol. Kedua arsitektur ini dipilih karena *Xception* dikenal memiliki akurasi yang tinggi meskipun dengan kompleksitas yang lebih besar, sementara *MobileNet* menawarkan

efisiensi komputasi yang baik dan cocok untuk aplikasi dengan sumber daya terbatas.

Pemilihan arsitektur *Xception* dan *MobileNet* didukung oleh berbagai penelitian sebelumnya yang menunjukkan efektivitas keduanya dalam klasifikasi berbasis citra, termasuk klasifikasi ikan. Tonapa, Manembu, dan Kambey [7] dalam penelitian mereka mengenai ikan cakalang dan tongkol menggunakan CNN dengan arsitektur VGG16 memperoleh *accuracy*, *precision*, dan *recall* sebesar 93%, dan f1 *score* 92%, sedangkan *ResNet50* mencapai *accuracy*, *precision*, *recall*, dan f1 *score* sebesar 95%. Sementara itu, studi oleh Diah Mitha Aprilla et al [8] menunjukkan bahwa VGG16 memperoleh *accuracy* 70.8%, *precision* 61%, *recall* 62%, dan f1 *score* 82%; *Xception* mendapatkan *accuracy* 88.3%, *precision* 82%, *recall* 85%, dan f1 *score* 82%; *ResNet50* mendapatkan *accuracy* 5.8%, *precision* 1%, *recall* 7%, dan f1 *score* 2%; *MobileNet* memperoleh *accuracy* 96.6%, *precision* 94%, *recall* 96%, dan f1 *score* 94%; serta *EfficientNetB0* mendapatkan *accuracy* 3.3%, *precision* 0%, *recall* 3%, dan f1 *score* 3.3%. Perbedaan signifikan dalam performa CNN terlihat dari setiap penelitian yang dilakukan.

Identifikasi yang akurat dari ikan cakalang dan tongkol penting tidak hanya untuk tujuan perdagangan dan konsumsi, tetapi juga untuk kepentingan penelitian biologi kelautan dan pengelolaan sumber daya ikan. Kesalahan dalam klasifikasi dapat mengakibatkan penilaian stok yang tidak akurat, yang pada akhirnya mempengaruhi pengelolaan dan keberlanjutan sumber daya perikanan. Penggunaan *Convolutional Neural Networks* (CNN) telah terbukti berhasil dalam klasifikasi objek termasuk ikan, dan penelitian ini berfokus pada penerapan arsitektur *Xception* dan *MobileNet* untuk meningkatkan akurasi dan efisiensi klasifikasi.

Penelitian ini membandingkan arsitektur *Xception* dan *MobileNet* dari segi akurasi, efisiensi komputasi, dan waktu pelatihan untuk mengidentifikasi apakah keduanya mampu memberikan hasil yang sebanding atau lebih baik dibandingkan arsitektur seperti VGG16 dan ResNet50. Dengan menonjolkan kelebihan dan kekurangan masing-masing, penelitian ini mengevaluasi sejauh mana *Xception* dan *MobileNet* bersaing dalam klasifikasi citra, terutama pada dataset yang terbatas.

Selain itu, penelitian ini menampilkan kebaruan dengan menggunakan arsitektur *Xception* dan *MobileNet* dengan arsitektur sebelumnya seperti VGG16 dan ResNet50, untuk klasifikasi ikan cakalang dan tongkol. Pendekatan ini juga dilengkapi dengan strategi augmentasi data yang spesifik pada data train untuk meningkatkan pada model terhadap variasi tampilan, yang belum dimaksimalkan dalam penelitian terdahulu. Selain itu, penelitian ini memperkenalkan analisis performa yang lebih mendalam, tidak hanya mengukur akurasi, precision, recall, dan F1 score, tetapi juga mengevaluasi efisiensi komputasi dan waktu pelatihan setiap arsitektur, memberikan kontribusi signifikan terhadap pemilihan model yang optimal dan aplikasi yang efisien di perangkat mobile atau lingkungan dengan sumber daya terbatas.

Oleh karena itu, dengan adanya penelitian ini diharapkan dapat memberikan kontribusi penting dalam pengelolaan sumber daya ikan, serta membantu mengurangi kesalahan klasifikasi yang dapat berdampak pada keberlanjutan sektor perikanan. juga membuka peluang untuk aplikasi klasifikasi otomatis yang lebih luas dalam bidang biologi kelautan dan perdagangan ikan.

2. TINJAUAN PUSTAKA

2.1. Penelitian terkait

Beberapa penelitian terkait menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur *MobileNet* dan *Xception* yang dilakukan pada ikan. Penelitian-penelitian ini akan dijadikan sebagai rujukan ketika pelaksanaan penelitian ini.

Salah satu penelitian yang menggunakan arsitektur *MobileNet* bertujuan untuk mengidentifikasi jenis ikan dengan menggunakan *dataset* yang terdiri dari 3960 gambar dari 468 spesies ikan, menghasilkan akurasi sebesar 89% ketika menggunakan 10 kelas [9]. Penelitian ini menunjukkan bahwa *MobileNet* efektif dalam melakukan identifikasi ikan.

Penelitian lain membandingkan arsitektur *ResNet50*, *Xception*, *InceptionV3*, VGG19, VGG16, dan *MobileNetV3Small* dalam menganalisis spesies ikan *Brachysoma Rastrelliger* dan *R*. *Kanagurta*. Dengan menggunakan *dataset* yang terdiri dari 517 gambar (267 gambar dari *Brachysoma Rastrelliger* dan 250 gambar dari *R. Kanagurta*), hasil terbaik dicapai oleh arsitektur *Xception* dengan akurasi 83,4% [10].

Penelitian lain menggunakan arsitektur CNN seperti VGG16, *InceptionV3*, *DenseNet121*, *MobileNet*, dan *Xception*, dengan teknik *transfer learning* dan *fine-tuning* untuk meningkatkan performa klasifikasi. *Dataset* terdiri dari 338 gambar dari 8 spesies ikan yang ditemukan di pesisir timur Ceará, Brasil. Studi ini menguji model CNN dengan dua skenario, dan hasil terbaik diperoleh oleh arsitektur *Xception* dengan akurasi 86%, sementara *MobileNet* mencapai akurasi 85% [11].

Penelitian-penelitian tersebut menunjukkan bahwa setiap arsitektur memiliki kelebihan dan kekurangannya masing-masing. *Xception* menawarkan akurasi tinggi namun dengan kompleksitas lebih besar, sementara *MobileNet* memberikan efisiensi komputasi yang cocok untuk aplikasi dengan sumber daya terbatas. Perbandingan kedua arsitektur ini diharapkan dapat mengidentifikasi arsitektur yang paling tepat untuk analisis spesifik ikan cakalang dan tongkol.

2.2. Citra

Citra adalah gambaran visual dari suatu objek yang didapatkan dari perangkat seperti kamera, *scanner* atau sensor lainnya. Citra ini berupa gambar dua dimensi (2D) yang terdiri dari piksel [3]. Dalam citra menampilkan suatu visual dari suatu objek, tetapi juga menyimpan informasi penting yang dapat dianalisis lebih lanjut.

2.3. *Deep Learning*

Deep learning, juga dikenal sebagai *deep structured learning*, *hierarchical learning*, atau *deep neural networks*, adalah metode pembelajaran yang memanfaatkan transformasi non-linear berlapis. *Deep learning* dapat dilihat sebagai gabungan antara *machine learning* dan jaringan saraf buatan (*artificial neural network*) [12]. *Deep Learning* (DL), yang mulai dikenal luas sejak tahun 2006, menggunakan pendekatan arsitektur pembelajaran yang mendalam atau pembelajaran hierarkis. Pembelajaran dalam konteks ini merupakan sebuah prosedur yang melibatkan proses estimasi parameter-parameter model, sehingga model yang dikembangkan (algoritma) mampu menyelesaikan tugas atau permasalahan tertentu [13].

2.4. *Convolutional Neural Network* (CNN)

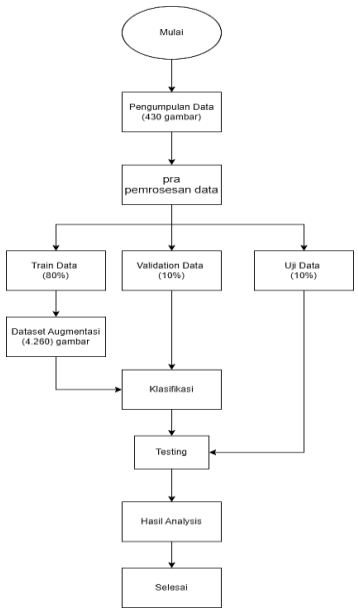
Convolutional Neural Network (CNN) adalah salah satu metode dalam *Deep Learning*. Metode ini merupakan pengembangan dari *Multi-Layer Perceptron* (MLP) yang memungkinkan pengolahan data dua dimensi seperti citra atau suara [14], [15]. CNN memiliki kemampuan yang lebih baik dalam memahami detail gambar karena arsitekturnya meniru cara otak manusia memproses informasi visual [16]. CNN digunakan untuk tugas klasifikasi dengan pendekatan pembelajaran yang diawasi (*supervised learning*). Dalam pendekatan ini, CNN memanfaatkan *dataset* yang telah diberi label atau kelas sebagai panduan untuk melatih modelnya, sehingga dapat mengenali pola dan fitur yang relevan untuk membedakan antara berbagai kategori data.

Arsitektur CNN terdiri dari dua bagian utama, yaitu *feature learning* dan *classification*. Dalam CNN terdapat empat lapisan utama, yaitu lapisan konvolusi (*Convolution Layer*), lapisan aktivasi (*Activation Layer*), lapisan *pooling* (*Pooling Layer*), dan lapisan *fully-connected* (*Fully-Connected Layer*) [17]. Lapisan-lapisan ini sering diulang atau disusun dalam beberapa lapisan sesuai kebutuhan arsitektur yang dibuat untuk mengekstraksi fitur-fitur dari data input. Setelah fitur diekstraksi, bagian *classification* mengambil alih dengan menggunakan lapisan *flatten*, *fully connected*, dan *softmax* untuk mengubah fitur-fitur yang diekstraksi menjadi probabilitas kelas yang memungkinkan model untuk melakukan klasifikasi akhir [15]. Pendekatan yang sistematis dan berlapis ini menjadikan CNN sangat efektif dalam menangani tugas-tugas pengenalan dan klasifikasi pada data citra dan suara.

2.5. *Transfer learning*

Transfer learning adalah teknik dalam *machine learning* yang memanfaatkan model yang telah dilatih sebelumnya untuk menyelesaikan tugas baru yang berkaitan, sehingga memerlukan lebih sedikit data dan waktu pelatihan ulang [18]. Teknik ini sering diterapkan ketika *dataset* yang tersedia bersifat terbatas. Proses *transfer learning* melibatkan pembekuan beberapa lapisan konvolusional awal, sementara hanya lapisan terakhir yang dilatih untuk klasifikasi. Lapisan yang dibekukan berfungsi sebagai fitur ekstraktor untuk tugas baru [19].

3. METODE PENELITIAN



Gambar 1. Alur Penelitian

Dalam penelitian ini, terdapat tahapan-tahapan kegiatan yang dilakukan pada alur penelitian, yaitu sebagai berikut.

3.1. Pengumpulan Data

Pengumpulan gambar dalam penelitian ini dilakukan dengan memperhatikan morfologi ikan yang sudah terverifikasi [20], menggunakan gambar tampak samping yang diperoleh dari internet (github) dan juga dari jurnal patokan. Namun, informasi tentang spesifikasi kamera yang digunakan atau kondisi pencahayaan saat pengambilan gambar tidak tersedia dalam dokumentasi *dataset* tersebut.

Gambar-gambar tersebut kemudian dikelompokkan ke dalam dua folder, masing-masing untuk ikan cakalang dan ikan tongkol. Proses ini memastikan bahwa *dataset* terstruktur dengan baik dan siap untuk analisis lebih lanjut guna mendukung *Accuracy* dan efisiensi dalam klasifikasi ikan.

TABEL I. CONTOH GAMBAR DATASET

Cakalang	Tongkol
	
	
	
	
	

Ikan tongkol merupakan spesies pelagis yang hidup di perairan pantai, memiliki ciri khas berupa sirip punggung yang berjauhan, dengan jumlah 10 hingga 12 duri pada sirip punggung pertama. Tubuhnya berwarna hitam kebiruan pada bagian atas dan memiliki area corselet yang polos, serta terdapat lebih dari 15 baris vertikal berwarna gelap di belakang sirip punggung pertama. Ikan ini memiliki lunas besar yang diapit oleh dua lunas kecil, serta lidah dengan dua guratan memanjang. Distribusi ikan tongkol bersifat circumglobal, dengan panjang tubuh dapat mencapai 50 cm [20].

Sementara itu, ikan cakalang merupakan spesies pelagis yang ditemukan di perairan pantai hingga samudera, memiliki lunas besar yang diapit oleh dua lunas kecil. Giginya kecil berbentuk kerucut, dan lidahnya juga memiliki dua guratan memanjang. Kedua sirip punggung ikan cakalang berdekatan, dengan warna tubuh hitam keunguan pada bagian atas dan keperakan di bagian bawah yang dilengkapi 4 hingga 6 garis gelap. Distribusinya juga circumglobal, dengan panjang tubuh dapat mencapai 110 cm [20].

3.2. Pra Pemrosesan Data

Pra-pemrosesan data merupakan tahap krusial dalam penelitian ini untuk meningkatkan kualitas data yang digunakan dalam pelatihan model pembelajaran mesin. Langkah ini mencakup penyesuaian ukuran gambar ikan menjadi 224x224 piksel, guna memastikan keseragaman kualitas dan ukuran, mengurangi beban komputasi, serta menjaga konsistensi selama pelatihan [21]. Pemilihan resolusi ini optimal karena mampu menyeimbangkan detail citra dan efisiensi komputasi, mendukung performa model dalam klasifikasi gambar secara efektif.

3.3. Augmentasi Data

Pada tahap ini, peneliti melakukan augmentasi *dataset* dengan tujuan untuk meningkatkan jumlah *dataset* serta membantu model mengenali setiap kelas dengan data yang lebih variatif. Augmentasi data dalam penelitian ini hanya diterapkan pada *data train* karena tujuannya adalah untuk memperluas variasi dan keragaman contoh yang dilihat oleh model selama proses pelatihan, sehingga model menjadi lebih *robust* dan mampu mengenali pola dengan lebih baik dalam berbagai kondisi. Data validasi dan data uji, di sisi lain, harus tetap representatif dari data nyata tanpa modifikasi. Ini penting agar data validasi dan data uji benar-benar mencerminkan performa model dalam kondisi aslinya dan memastikan evaluasi model yang adil dan akurat, tanpa pengaruh dari modifikasi yang tidak seharusnya terjadi.

Adapun augmentasi yang diterapkan meliputi teknik *rescale*, *rotation_range*, *width_shift_range*, *height_shift_range*, *shear_range*, *zoom_range*, *horizontal_flip*, dan *fill_mode*. Penggunaan parameter augmentasi tersebut bertujuan untuk meningkatkan variasi data dan mencegah *overfitting* pada model klasifikasi citra. Dengan menerapkan teknik-teknik tersebut, jumlah data awal yang semula sebanyak 430 gambar meningkat menjadi 4.260 gambar, sehingga

model dapat dilatih dengan lebih baik dan memiliki kemampuan generalisasi yang lebih tinggi.

3.4. *Classification*

Penelitian ini menggunakan dua arsitektur *Deep Learning*, yaitu *Xception* dan *MobileNet*, untuk klasifikasi spesies ikan cakalang dan tongkol. Arsitektur *Xception* adalah pengembangan dari *Inception* yang menggunakan *depthwise separable convolutions*, yaitu teknik pemisahan antara konvolusi spasial dan konvolusi kanal, untuk mengoptimalkan proses ekstraksi fitur dan meningkatkan efisiensi pada *dataset* gambar kompleks. Tahapan klasifikasi dalam *Xception* melibatkan blok-blok konvolusi yang terpisah, diikuti dengan lapisan *pooling* dan *fully connected* untuk prediksi akhir. Sementara itu, arsitektur *MobileNet* dirancang dengan fokus pada kecepatan dan efisiensi komputasi. *MobileNet* juga menggunakan *depthwise separable convolutions* untuk mengurangi jumlah parameter dan beban komputasi, sehingga cocok untuk aplikasi pada perangkat *mobile* dengan keterbatasan sumber daya. Dalam tahapan klasifikasi, *MobileNet* menggunakan lapisan konvolusi yang diikuti dengan lapisan *batch normalization* dan aktivasi *ReLU* untuk menghasilkan keluaran prediksi yang akurat namun efisien.

4. HASIL DAN PEMBAHASAN

Dalam pelatihan model ini, digunakan beberapa parameter penting, termasuk *batch_size* = 32, epoch = 200, ukuran citra gambar 224 x 224 piksel, serta pembagian data latih, data validasi dan data uji masing-masing sebesar 80%, 10% dan 10%. Parameter-parameter ini dioptimalkan untuk mencapai kinerja yang maksimal dalam pengenalan pola pada *dataset* yang digunakan.

Dalam penelitian ini, penulis akan melakukan uji coba untuk membandingkan hasil antara menggunakan augmentasi *dataset* dan tidak menggunakan augmentasi *dataset*. Peneliti juga akan menerapkan beberapa metode augmentasi *dataset* yang berbeda.

4.1. *Dataset* Augmentasi

TABEL II. NILAI AUGMENTASI

Parameter	Nilai
Rescale	1./255
Rotation range	15
Width shift range	0.2
Height shift range	0.2
Shear range	0.2
Zoom range	0.2

Dalam proses augmentasi gambar, beberapa parameter digunakan untuk meningkatkan variasi data. *Rescale*= 1./255 dilakukan untuk menormalkan *pixel* pada gambar dari nilai 0-255 menjadi antara 0 dan 1, yang membantu dalam konvergensi model. *Rotation range* sebesar 15 dipilih karena perubahan sudut yang moderat sudah cukup untuk merepresentasikan variasi rotasi alami dari objek tanpa terlalu mengubah struktur dasar gambar. *Width shift range* dan *height shift range* sebesar 0.2 memberikan fleksibilitas terhadap pergeseran posisi objek, sementara *shear range* dan *zoom range* masing-masing memungkinkan perubahan geometri dan ukuran objek. Nilai-nilai ini seimbang untuk menghasilkan variasi gambar yang realistis tanpa merusak struktur asli objek.

TABEL III. DATASET YANG SEBELUM DAN SESUDAH AUGMENTASI

Dataset	Sebelum	Sesudah
Cakalang		
Tongkol		

Dataset yang diaugmentasi mengalami perubahan karena transformasi yang diterapkan pada gambar-gambar aslinya. Sebelum augmentasi, *dataset* hanya berisi gambar-gambar asli tanpa variasi tambahan. Setelah augmentasi, *dataset* berubah karena setiap gambar asli dimodifikasi melalui berbagai teknik, seperti *rescaling*, rotasi, translasi, *shear*, dan *zoom*. Transformasi ini menciptakan versi baru dari gambar yang memperkenalkan variasi dalam bentuk, posisi, ukuran, dan orientasi objek. Akibatnya, *dataset* yang dihasilkan memiliki lebih banyak variasi visual dibandingkan *dataset* asli, yang membantu model belajar dari data yang lebih beragam.

4.2. *Dataset Prepossessing*

Dataset yang dikumpulkan akan diproses untuk memastikan struktur data yang rapi dan siap digunakan dalam pelatihan model. Proses ini mencakup penyesuaian dan optimalisasi data untuk

meningkatkan *Accuracy* dan efisiensi model dalam mengenali pola. Hasil penelitian menunjukkan bahwa penggunaan resolusi citra 224x224 piksel secara konsisten memberikan performa optimal dalam klasifikasi, dengan keseimbangan yang baik antara *Accuracy* dan efisiensi komputasi, serta kemampuan untuk mempertahankan detail penting dalam gambar yang mendukung klasifikasi yang efektif.

TABEL IV. DATASET YANG SEBELUM DAN SESUDAH PREPOSSESSING

Dataset	Sebelum	Sesudah
Cakalang		
Tongkol		

4.3. Pembuatan Model

Pembangunan model pada penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) untuk melakukan klasifikasi citra dan akan menggunakan *library TensorFlow* untuk inisialisasi dan konfigurasi model. Berikut ini adalah konfigurasi *hyperparameter* yang digunakan dalam pembangunan model.

4.3.1. *MobileNet*

TABEL V. ARSITEKTUR MODEL ARSITEKTUR *MOBILENET*

Layer Type	***Description***	***Output Shape***
Input	*Input shape*	(224, 224, 3)
GlobalMaxPooling2D	Operasi penyatuan maksimum diterapkan di seluruh *input*	(1280)
Dense	128 unit, aktivasi: ReLU	(128)
Dense	128 unit, aktivasi: ReLU	(128)
Dense	2 unit, aktivasi: *Sigmoid* (Lapisan keluaran untuk klasifikasi biner	(2)

TOTAL PARAMS: 3,228,864 (12.32 MB)
TRAINABLE PARAMS: 3,206,976 (12.23 MB)
NON-TRAINABLE PARAMS: 21, 888 (85.50 KB)

Pada bagian klasifikasi, terdapat lapisan *GlobalMaxPooling2D* dan tiga lapisan *fully connected*. Lapisan *GlobalMaxPooling2D* mengambil nilai maksimum dari setiap dimensi *feature map* yang dihasilkan oleh lapisan konvolusi terakhir, menghasilkan vektor 1 dimensi yang kemudian digunakan sebagai input untuk lapisan *fully connected*. Lapisan *fully connected* pertama terdiri dari 128 neuron dengan fungsi aktivasi ReLU, diikuti oleh lapisan *fully connected* kedua yang juga memiliki 128 neuron dengan fungsi aktivasi ReLU. Lapisan *fully connected* ketiga, yang merupakan lapisan *output*, memiliki 2 neuron untuk tujuan klasifikasi dan menggunakan fungsi aktivasi *sigmoid*.

Model ini memiliki total 3.228.864 parameter, yang berukuran 12,32 MB, terdiri dari 3.206.976 parameter yang dapat dilatih dan 21.888 parameter yang tidak dapat dilatih. Parameter yang dapat dilatih mencakup bobot dan bias yang diperbarui selama pelatihan, sementara parameter yang tidak dapat dilatih umumnya terkait dengan layer dengan parameter tetap, seperti statistik *batch normalization*. Total parameter ini memberikan gambaran tentang kompleksitas model dan kebutuhan memori untuk pelatihan serta inferensi.

4.3.2. *Xception*

TABEL VI. ARSITEKTUR MODEL ARSITEKTUR *XCEPTION*

Layer Type	***Description***	***Output Shape***
Input	*Input shape*	(224, 224, 3)
GlobalMaxPooling2D	Operasi penyatuan maksimum diterapkan di seluruh *input*	(1024)
Dense	128 unit, aktivasi: ReLU	(128)
Dense	128 unit, aktivasi: ReLU	(128)
Dense	2 unit, aktivasi: *Sigmoid* (Lapisan keluaran untuk klasifikasi biner	(2)

TOTAL PARAMS: 20,861,480 (79.58 MB)
TRAINABLE PARAMS: 20,806,952 (79.37 MB)
NON-TRAINABLE PARAMS: 54,528 (213.00 KB)

Pada bagian klasifikasi, terdapat lapisan *GlobalMaxPooling2D* dan tiga lapisan *fully connected*. Lapisan *GlobalMaxPooling2D* mengambil nilai maksimum dari setiap dimensi *feature map* yang dihasilkan oleh lapisan konvolusi terakhir, menghasilkan vektor 1 dimensi yang kemudian

digunakan sebagai input untuk lapisan *fully connected*. Lapisan *fully connected* pertama terdiri dari 128 neuron dengan fungsi aktivasi ReLU, diikuti oleh lapisan *fully connected* kedua yang juga memiliki 128 neuron dengan fungsi aktivasi ReLU. Lapisan *fully connected* ketiga, yang merupakan lapisan *output*, memiliki 2 neuron untuk tujuan klasifikasi dan menggunakan fungsi aktivasi *sigmoid*.

Model ini memiliki total 20.861.480 parameter, yang berukuran 79.58 MB, terdiri dari 20,806,952 parameter yang dapat dilatih dan 54,528 parameter yang tidak dapat dilatih. Parameter yang dapat dilatih mencakup bobot dan bias yang diperbarui selama pelatihan, sementara parameter yang tidak dapat dilatih umumnya terkait dengan layer dengan parameter tetap, seperti statistik *batch normalization*. Total parameter ini memberikan gambaran tentang kompleksitas model dan kebutuhan memori untuk pelatihan serta inferensi.

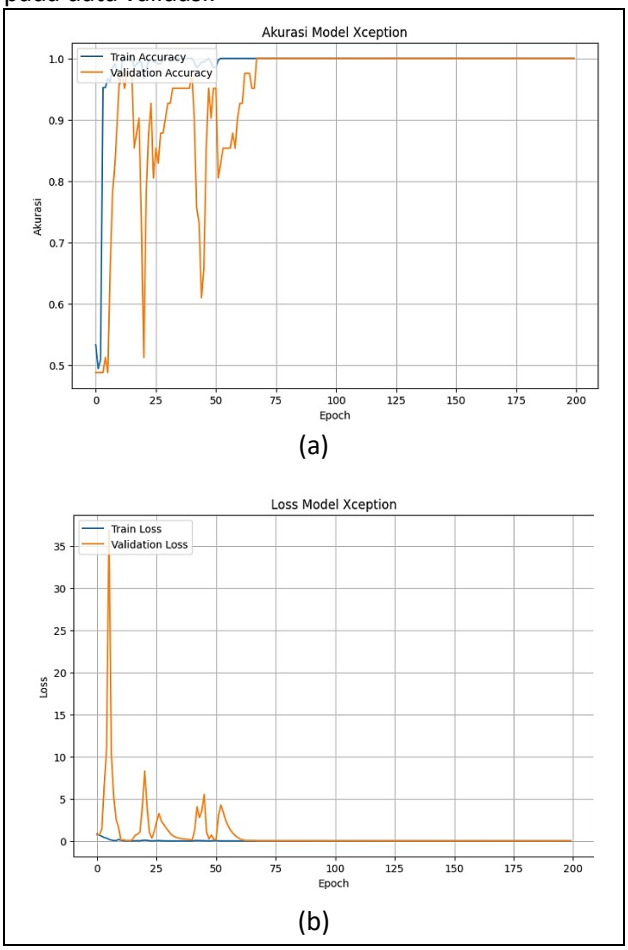
4.4. Hasil Pembahasan dan Analisis

Sebelum menampilkan grafik *Accuracy* dan *loss*, model dilatih menggunakan data yang dibagi menjadi 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk uji. Pada tahap pelatihan, model *Xception* dan *MobileNet* dilatih untuk mengenali pola dari data pelatihan, kemudian dievaluasi pada data validasi guna menguji kemampuan generalisasi model. Model *Xception* dan *MobileNet* dilatih selama 200 epoch dan dievaluasi menggunakan metrik *Accuracy* dan *loss* untuk mengukur kinerja dan kesalahan model.

Hasil evaluasi divisualisasikan dalam grafik untuk menunjukkan stabilitas kinerja. Data uji sebesar 10% digunakan untuk mengevaluasi performa model berdasarkan *Accuracy*, loss, dan efektivitas terhadap data yang belum pernah dihadapi sebelumnya untuk kemudian digunakan dalam menilai kemampuan model dalam generalisasi dan menilai keberhasilan proses pelatihan.

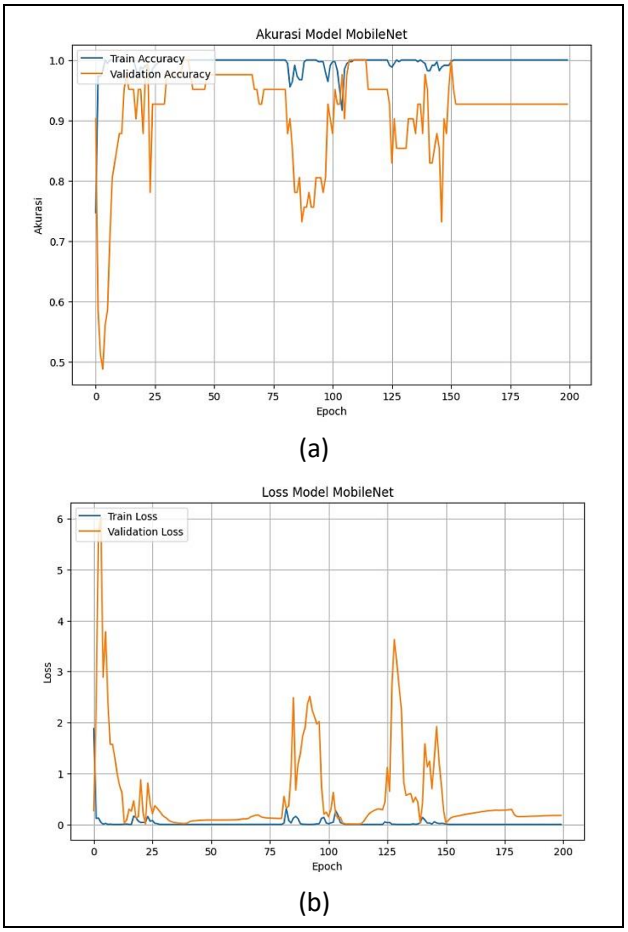
4.4.1. Tanpa Augmentasi

Pada Gambar 2, hasil pengujian *Xception* tanpa augmentasi menunjukkan peningkatan cepat di awal, dengan *Accuracy* pelatihan mencapai 0.9318, menandakan model belajar baik pada data latih. Namun, terdapat fluktuasi pada *Accuracy* validasi, menunjukkan potensi *overfitting* karena adanya perbedaan yang mencolok antara *Accuracy* pelatihan dan *validasi*. Sedangkan pada nilai *Loss* pelatihan konsisten menurun hingga 0.2386, namun *loss* validasi lebih bervariasi dengan beberapa lonjakan signifikan, menunjukkan model kesulitan menjaga kinerja stabil pada data validasi. Secara keseluruhan, model belajar baik pada data latih, namun memerlukan regularisasi atau lebih banyak data untuk meningkatkan kinerja pada data validasi.



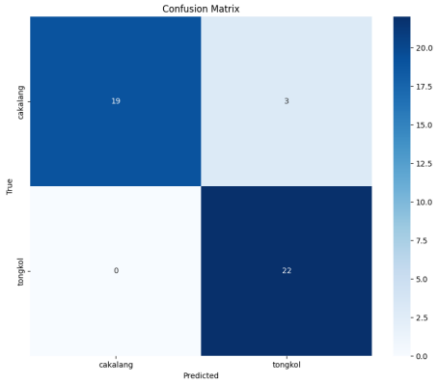
Gambar 2. (a) Hasil Pengujian Akurasi Model *Xception* Tanpa Augmentasi dan (b) Hasil Pengujian Loss Model *Xception* Tanpa Augmentasi

Hasil pengujian *MobileNet* tanpa augmentasi pada Gambar 3 menunjukkan *Accuracy* pelatihan mencapai 0.9772, menandakan model belajar baik pada data latih. Namun, terdapat fluktuasi pada *Accuracy* validasi, dengan beberapa penurunan signifikan yang mungkin mengindikasikan *overfitting*. *Loss* pada pelatihan turun stabil hingga 0.1400, namun *loss* pada data validasi menunjukkan lonjakan besar di beberapa epoch, menandakan tantangan dalam menjaga performa pada data validasi. Secara keseluruhan, meskipun *MobileNet* belajar baik pada data latih, diperlukan peningkatan seperti regularisasi atau augmentasi untuk stabilitas pada data validasi.

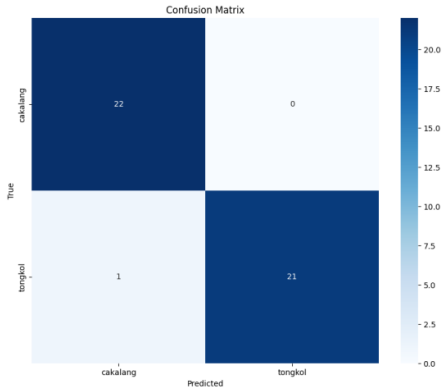


Gambar 3. (a) Hasil Pengujian Akurasi Model *MobileNet* Tanpa Augmentasi dan (b) Hasil Pengujian Loss Model *MobileNet* Tanpa Augmentasi

Untuk mengetahui performa model dalam memprediksi ikan cakalang dan ikan tongkol dilakukan evaluasi menggunakan *confusion matrix* dengan hasil seperti yang ditunjukkan pada gambar 4 untuk *Xception* dan gambar 5 untuk *MobileNet*.



Gambar 4. Hasil Pengujian *Confusion matrix Xception* Tanpa Augmentasi



Gambar 5. Hasil Pengujian *Confusion matrix MobileNet* Tanpa Augmentasi

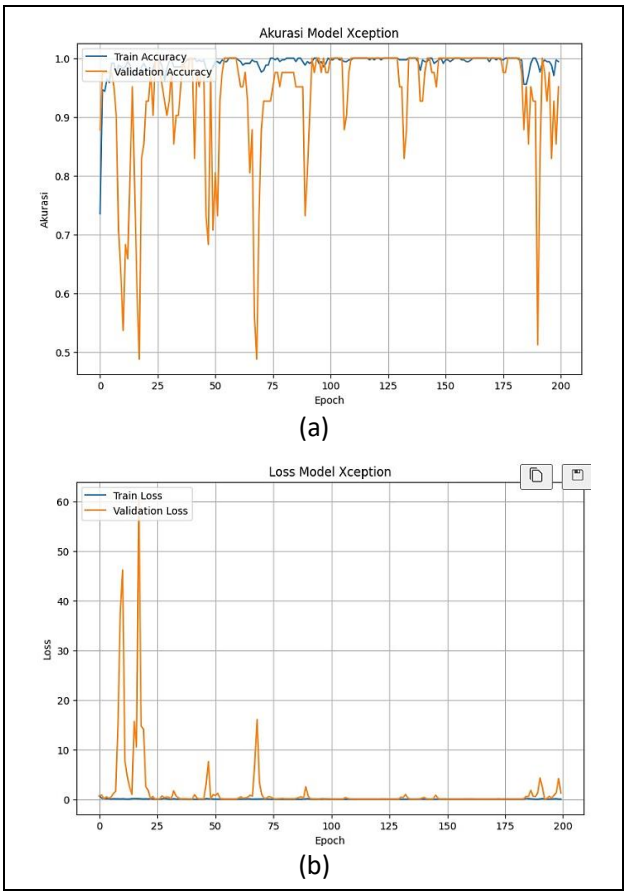
Selanjutnya, ditampilkan tabel yang berisi kolom *accuracy, precision, recall*, F1-*score*, dan waktu prediksi untuk membantu dalam menginterpretasikan sejauh mana model mampu mengklasifikasikan citra ikan cakalang dan ikan tongkol dengan benar. Setelah melalui pelatihan dan pengujian.

TABEL VII. HASIL ARSITEKTUR *XCEPTION* DAN *MOBILENET* TANPA AUGMENTASI

Metode	*Accuracy*	*Precision*	*Recall*	F1-*Score*	Waktu Prediksi
Xception	93%	94%	93%	93%	0.13s
MobileNet	97%	97%	97%	97%	0.06s

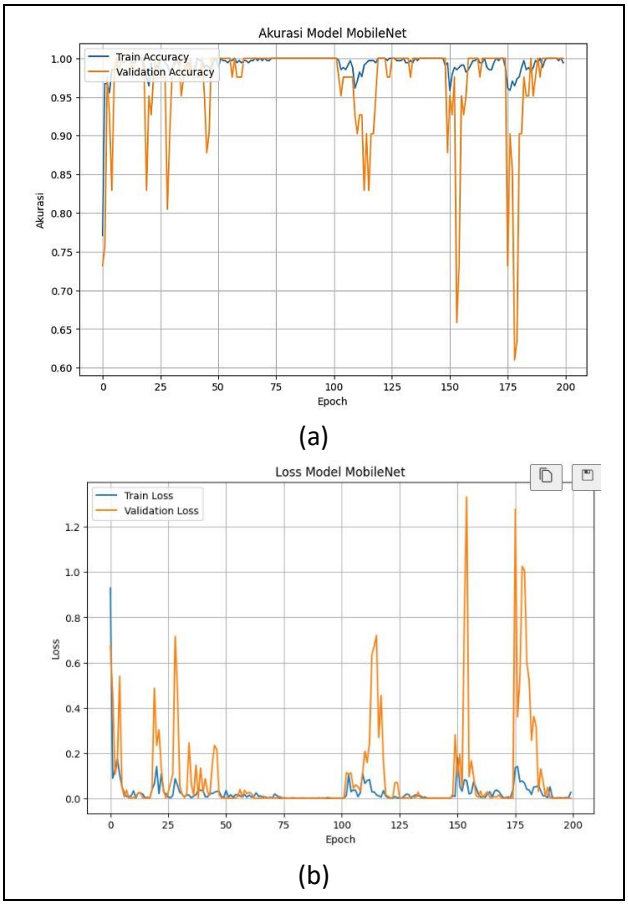
Hasil pengujian pada Tabel VII. Hasil Arsitektur *Xception* Dan *MobileNet* Tanpa Augmentasi menunjukkan bahwa model *Xception* tanpa augmentasi data menunjukkan kinerja yang jauh lebih unggul dibandingkan model *MobileNet*. *Xception* memperoleh nilai yang sangat tinggi pada metrik performa, dengan *Accuracy* 95%, *Precision* 96%, *recall* 95%, dan F1-*score* 95%, sedangkan *MobileNet* mencatat nilai yang lebih rendah, yaitu 98% untuk *Accuracy*, 98% untuk *Precision*, 98% untuk *recall*, dan 98% untuk F1-*score*. Selain *MobileNet* juga menunjukkan lebih baik dalam waktu pelatihan, dengan durasi hanya 134 menit 1,4 detik, sedangkan *Xception* memerlukan waktu 560 menit 53,4 detik. Selain itu, waktu prediksi untuk model *Xception* tercatat sebesar 0.13 detik, sedangkan *MobileNet* membutuhkan waktu jauh lebih sedikit, yaitu 0.06 detik. *MobileNet* unggul dalam efisiensi waktu dikarenakan model sangat dipengaruhi oleh base model yang dipakai. *Xception* dengan 20.861.480 parameter sedangkan *MobileNet* dengan total 3.228.864 parameter. Base model yang lebih ringan dapat mempercepat waktu pelatihan secara signifikan, terutama ketika sumber daya komputasi terbatas.

4.4.2. Augmentasi



Gambar 6. (a) Hasil Pengujian Akurasi Model *Xception* Menggunakan Augmentasi dan (b) Hasil Pengujian Loss Model *Xception* Menggunakan Augmentasi

Pada Gambar 6. Hasil Pengujian *Xception* Menggunakan Augmentasi, menggambarkan hasil pengujian model *Xception*. Grafik *Accuracy* di sisi kiri menampilkan kinerja model dalam membedakan antara data pelatihan dan data validasi. *Accuracy* pelatihan terlihat cukup stabil setelah beberapa epoch, sedangkan *Accuracy* validasi menunjukkan fluktuasi yang cukup besar. Ini mengindikasikan adanya kesulitan dalam generalisasi model terhadap data validasi, yang dapat menunjukkan potensi *overfitting*. Pada grafik loss di sebelah kanan, terlihat bahwa loss validasi memiliki fluktuasi lebih besar dibandingkan dengan loss pelatihan, terutama pada fase awal pelatihan. Fluktuasi pada nilai loss validasi ini juga menguatkan indikasi bahwa model mengalami kesulitan dalam menjaga kinerja yang seimbang antara data pelatihan dan data validasi.

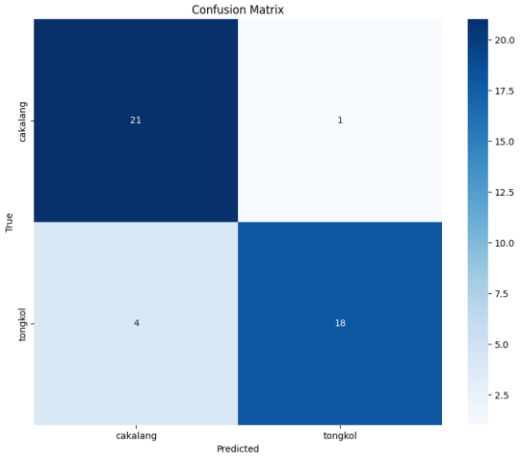


Gambar 7. (a) Hasil Pengujian Akurasi Model *MobileNet* Menggunakan Augmentasi dan (b) Hasil Pengujian Loss Model *MobileNet* Menggunakan Augmentasi

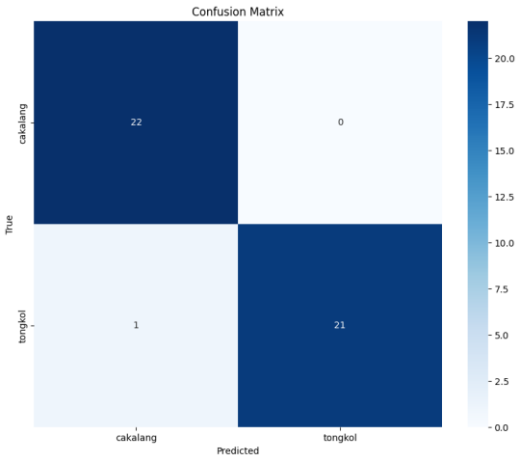
Grafik pada gambar 7. Hasil Pengujian *MobileNet* Menggunakan Augmentasi menunjukkan hasil pengujian model *MobileNet* dengan menggunakan data pelatihan sebesar 80% dan data validasi sebesar 10%. Pada grafik *Accuracy* di sebelah kiri, *Accuracy* pelatihan terlihat stabil mendekati 1 setelah beberapa epoch. Namun, *Accuracy* validasi memperlihatkan fluktuasi yang signifikan pada beberapa epoch, yang menunjukkan tantangan dalam generalisasi model terhadap data validasi. Grafik loss di sebelah kanan mengungkapkan tren serupa, di mana loss validasi cenderung lebih bervariasi dibandingkan dengan loss pelatihan, terutama di awal dan pertengahan pelatihan, yang menandakan adanya potensi *overfitting*.

Pengujian menghasilkan kinerja model yang optimal, dengan nilai *Accuracy*, precision, recall, dan F1-score masing-masing mencapai 100%. Hasil ini menunjukkan bahwa *MobileNet* berhasil memberikan performa tinggi dalam klasifikasi, meskipun tetap menghadapi fluktuasi dalam validasi.

Untuk mengevaluasi performa model dalam memprediksi ikan cakalang dan ikan tongkol, dilakukan analisis menggunakan *confusion matrix*, yang hasilnya dapat dilihat pada gambar 8 untuk model *Xception* dan gambar 9 untuk model *MobileNet*.



Gambar 8. Hasil Pengujian *Confusion matrix Xception* Menggunakan Augmentasi



Gambar 9. Hasil Pengujian *Confusion matrix MobileNet* Menggunakan Augmentasi

Kemudian, ditampilkan sebuah tabel yang mencakup kolom *accuracy*, *precision*, *recall*, F1-*score*, serta waktu prediksi untuk mempermudah dalam memahami kemampuan model dalam mengklasifikasikan gambar ikan cakalang dan ikan tongkol secara akurat. Hasil ini merupakan output akhir setelah model melewati tahapan pelatihan dan pengujian secara menyeluruh.

TABEL VIII. HASIL ARSITEKTUR *XCEPTION* DAN *MOBILENET* DENGAN AUGMENTASI

Metode	*Accuracy*	*Precision*	*Recall*	F1-*Score*	Waktu Prediksi
Xception	88%	89%	88%	88%	0.13s
MobileNet	100%	100%	100%	100%	0.09s

Hasil pengujian pada Tabel VIII. Hasil Arsitektur *Xception* Dan *MobileNet* Dengan Augmentasi dari metrik performa di mana *MobileNet* mencapai *Accuracy*, *Precision*, *recall*, dan F1-*score* sempurna dengan nilai 100%, sementara model *Xception* sedikit lebih rendah dengan nilai 98% pada semua metrik. Dengan jumlah parameter yang lebih sedikit, *MobileNet* juga unggul dalam efisiensi waktu pelatihan, hanya membutuhkan 123 menit 29,3 detik, jauh lebih singkat dibandingkan *Xception* yang memerlukan 515 menit 29,7 detik. Selain itu, waktu prediksi untuk model *Xception* tercatat sebesar 0.30 detik, sedangkan *MobileNet* membutuhkan waktu jauh lebih sedikit, yaitu 0.09 detik. Keunggulan *MobileNet* mencakup efisiensi waktu pelatihan dan waktu prediksi, arsitektur yang lebih ringan, dan penggunaan memori yang lebih hemat, menjadikannya pilihan yang lebih efisien dan optimal terutama dalam situasi dengan keterbatasan sumber daya komputasi, tanpa mengorbankan *Accuracy* dan kinerja model. Model dasar yang lebih ringan dapat secara drastis mempercepat proses pelatihan, terutama saat sumber daya komputasi terbatas.

Pada Gambar 4, Gambar 5, Gambar 8 dan Gambar 9 menunjukkan performa dari masing-masing metode dimana pada Gambar 4 dan Gambar 8 menunjukkan hasil dari metode *Xception* memprediksi data uji. Gambar 4 melakukan prediksi dengan benar 19 untuk ikan cakalang dan 22 untuk ikan tongkol. Sedangkan untuk Gambar 8 dengan metode *Xception* menggunakan Augmentasi mendapatkan hasil prediksi dengan benar 21 untuk ikan cakalang dan 18 untuk ikan tongkol. Ini menunjukkan bahwa penerapan augmentasi memberikan peningkatan performa dalam mengenali ikan cakalang, tetapi tidak secara signifikan meningkatkan prediksi untuk ikan tongkol.

Pada Gambar 5 dan Gambar 9 menunjukkan hasil dari metode *MobileNet* memprediksi data uji. Gambar 5 *MobileNet* tanpa augmentasi melakukan prediksi dengan benar 22 untuk ikan cakalang sedangkan 21 untuk ikan tongkol. Sedangkan untuk Gambar 9 *MobileNet* menggunakan Augmentasi mendapatkan hasil prediksi benar 22 untuk ikan cakalang dan 22 untuk ikan tongkol. Ini menunjukkan bahwa performa dari Augmentasi yang digunakan untuk *MobileNet* mendapatkan hasil yang sempurna.

Untuk metode *Xception* menunjukkan bahwa augmentasi memberikan sedikit peningkatan dalam mengenali ikan cakalang, tetapi tidak signifikan untuk ikan tongkol. Sementara itu untuk metode *MobileNet* menunjukkan performa yang lebih baik, di mana penggunaan augmentasi menghasilkan prediksi yang

sempurna untuk kedua jenis ikan. Hal ini menunjukkan bahwa augmentasi berkontribusi positif terhadap akurasi model *MobileNet*, meningkatkan kinerjanya secara keseluruhan.

Pada Gambar 2, Gambar 3, Gambar 6, dan Gambar 7 menunjukkan bahwa terjadinya *overfitting* yang dimana terjadi fluktuasi yang ekstrem pada grafik accuracy dan grafik loss nya. *Overfitting* adalah ketika algoritma peramalan terlalu akurat menggambarkan data pelatihan, sehingga model hanya "menghafal" pola yang ada tanpa benar-benar "mempelajari" pola tersebut. Akibatnya, model tidak dapat menggeneralisasi dengan baik pada data baru, karena terlalu bergantung pada informasi dari data yang telah dilatih [22]. Pada percobaan ini *overfitting* terjadi karena pelatihan yang banyak dan dataset yang terbatas. Ini ditunjukkan pada Tabel IX Rangkuman Hasil Pelatihan Model.

TABEL IX. RANGKUMAN HASIL PELATIHAN MODEL

Scenario	Epoch	*Accuracy*	*Loss Accuracy*	*Val Accuracy*	*Val Loss*
Xception Augmentasi	129	100%	0.0073%	100%	0.0001%
MobileNet Augmentasi	197	100%	0.14%	100%	0.0006%
Xception Tanpa Augmentasi	189	100%	0.0002%	100%	0.0065%
MobileNet Tanpa Augmentasi	113	100%	0.0009%	100%	0.0050%

Secara keseluruhan setelah melalui pelatihan model dengan menerapkan augmentasi data dan tanpa augmentasi data hasil menunjukkan bahwa pelatihan dengan kedua *scenario* tersebut tidak menunjukkan perbedaan hasil yang signifikan ini ditunjukkan pada nilai *accuracy* sebesar 100%. Dari hasil pelatihan pada tabel di atas, *overfitting* dapat diidentifikasi melalui perbedaan antara nilai *loss* pada data pelatihan dan *val_loss* pada data validasi. Meskipun *accuracy* pada semua skenario mencapai 100%, adanya selisih yang signifikan antara *loss* dan *val_loss* menunjukkan bahwa model memiliki performa yang sangat baik pada data pelatihan, tetapi tidak dapat menggeneralisasi dengan baik pada data validasi.

Pada skenario *Xception* Augmentasi, *loss* sebesar 0.0073% sangat rendah, tetapi *val_loss* jauh lebih kecil yaitu 0.0001%. Pada skenario *MobileNet* Augmentasi, *loss* sebesar 0.14% lebih besar daripada *val_loss* yang hanya 0.0006%. Sementara itu, pada skenario tanpa augmentasi, baik untuk *Xception* maupun *MobileNet*, perbedaan antara *loss* dan *val_loss* juga terlihat, terutama pada skenario *Xception* Tanpa Augmentasi di mana loss sangat kecil (0.0002%), tetapi *val_loss* lebih tinggi (0.0065%).

Perbedaan ini menunjukkan bahwa model sangat terfokus pada data pelatihan dan tidak dapat mempertahankan performa serupa pada data validasi, yang merupakan indikasi terjadinya *overfitting*.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa hal yang dapat disimpulkan oleh penulis, antara lain sebagai berikut:

1. Penelitian ini menunjukkan bahwa *MobileNet* lebih efisien dalam waktu pelatihan dan inferensi dibandingkan *Xception*, terutama untuk aplikasi dengan sumber daya komputasi terbatas.
2. *MobileNet* memiliki akurasi, presisi, *recall*, dan F1-*score* yang lebih tinggi, serta waktu prediksi yang lebih cepat.
3. *MobileNet* lebih optimal untuk klasifikasi ikan cakalang dan ikan tongkol, meskipun *Xception* memiliki kompleksitas lebih tinggi.
4. Keunggulan *Xception* dalam *dataset* kompleks menunjukkan potensinya untuk klasifikasi yang memerlukan akurasi tinggi, meski dengan biaya komputasi yang lebih besar.
5. Secara keseluruhan, penelitian ini berkontribusi pada pengembangan metode klasifikasi berbasis *deep learning* untuk pengelolaan sumber daya perikanan dan eksplorasi model lain untuk data citra dan lingkungan komputasi.

5.2. Saran

Saran yang dapat digunakan untuk penelitian selanjutnya, yaitu untuk mengoptimalkan model dengan memilih arsitektur yang lebih ringan guna menekan biaya komputasi dan menerapkan teknik regularisasi tambahan guna mencegah *overfitting*. Eksplorasi arsitektur lain, seperti yang memiliki efisiensi tinggi juga dapat dipertimbangkan untuk meningkatkan akurasi. Selain itu, penerapan teknik augmentasi data yang lebih variatif juga penting untuk memperkuat kemampuan generalisasi model terhadap data citra ikan.

DAFTAR PUSTAKA

[1] S. Riyanto and H. Mardiansjah, "Pengembangan Industri Pengolahan Perikanan dalam Pengembangan Ekonomi Lokal," *Jurnal Litbang*, vol. 14, no. 2, pp. 107–118, 2018.

[2] M. Samdani, I. W. Restu, and R. Ekawaty, "Inventarisasi Ikan Ekonomis Penting pada Musim Barat di PPI Kedonganan, Bali," *Journal of Marine and Aquatic Sciences*, vol. 7, no. 1, p. 10, Feb. 2021.

[3] L. D. Sukarman, G. F. Laxmi, and F. Fatimah, "Identifikasi Ikan Air Tawar dengan Metode Color Moment Feature," *Seminar Nasional Teknologi Informasi Universitas Ibn Khaldun Bogor 2018*, p. 375, 2018.

[4] I. Ariawan, W. Aprizal Arifin, A. Armelita Rosalia, and N. Tufailah, "Klasifikasi Tiga Genus Ikan Karang Menggunakan Convolution Neural Network," *Jurnal Ilmu dan Teknologi Kelautan Tropis*, vol. 14, pp. 205–216, Aug. 2022.

[5] I. Anugrah, A. Cendekia Siregar, and B. C. Octariadi, "Perbandingan Model Arsitektur CNN Dengan Metode Transfer Learning Untuk Klasifikasi Spesies Ikan Laut," *Jurnal Ilmiah Komputer*, vol. 20, no. 1, pp. 444–453, Jan. 2024.

[6] Cakra, S. Syarif, H. Gani, A. Patombongi, and Muh. A. Islah, "Analisis Kesegaran Ikan Mujair dan Ikan Nila dengan Metode Convolutional Neural Network," *Jurnal Sistem Informasi dan Teknik Komputer*, vol. 7, no. 2, pp. 74–79, 2022.

[7] W. A. Tonapa, P. D. K. Manembu, and F. D. Kambey, "Fish Classification of Skipjack and Mackerel Tuna Using Convolutional Neural Network," *Jurnal Teknik Informatika*, vol. 19, no. 01, pp. 31–36, Mar. 2024.

[8] D. M. Aprilla, F. Bimantoro, and I. G. P. Suta Wijaya, "The Palmprint Recognition Using Xception, VGG16, ResNet50, MobileNet, and EfficientNetB0 Architecture," *Jurnal Media Informatika Budidarma*, vol. 8, no. 2, p. 1065, Apr. 2024.

[9] Herlambang Duwi Prasetyo, Pandu Ananto Hogantara, and Ika Nurlaili Isnainiyah, "MobileNets-V1 Architecture for Web Based Fish Image Classification," *Journal of Computing and Applied Informatics*, vol. 5, no. 2, pp. 60–70, Jul. 2021.

[10] R. Jongjaraunsuk, W. Taparhudee, S. Sirisuay, M. Kaewnern, V. Dulyapurk, and S. Janekitkarn, "Transfer Learning Model Application for Rastrelliger brachysoma and R. kanagurta Image Classification Using Smartphone-Captured Images," *Fishes*, vol. 9, no. 3, Mar. 2024.

[11] R. S. Monteiro *et al.*, "Fish Recognition Model for Fraud Prevention using Convolutional Neural Networks.".

[12] P. Adi Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) pada Ekspresi Manusia," *Jurnal Algor*, vol. 2, no. 1, pp. 12–21, 2020.

[13] M. Haris, T. Pustaka, M. H. Diponegoro, S. Kusumawardani, and I. Hidayah, "Tinjauan Pustaka Sistematis: Implementasi Metode Deep Learning pada Prediksi Kinerja Murid," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 10, no. 2, pp. 131–138, 2021.

[14] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *Jurnal Sistem & Teknologi Informasi Indonesia*, vol. 3, no. 2, pp. 49–56, Aug. 2018.

[15] A. Yusuf, R. Cahya Wihandika, and C. Dewi, "Klasifikasi Emosi Berdasarkan Ciri Wajah Menggunakan Convolutional Neural Network," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 11, Nov. 2019.

[16] Anhar and R. A. Putra, "Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN)," *Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 11, no. 2, pp. 466–478, Apr. 2023.

[17] A. Hibatullah and I. Maliki, "Penerapan Metode Convolutional Neural Network pada Pengenalan Pola Citra Sandi Rumput," Computer Science, Information & General Works, Universitas Komputer Indonesia, 2019.

[18] G. Eka Okta Putra, K. Queena Fredlina, and I. Nyoman Yudi Anggara Wijaya, "Implementasi Transfer Learning Menggunakan Convolutional Neural Network dalam Klasifikasi Penyu," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 1, pp. 1077–1082, 2024.

[19] A. Eka *et al.*, "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning," 2023.

[20] W. T. White *et al.*, "Market Fishes of Indonesia," 2013.

[21] N. Khasanah, "Komparasi Arsitektur ResNet50 dan VGG16 untuk Klasifikasi Citra Tanda Tangan," *Jurnal Sistem Informasi*, vol. 14, no. 1, 2022.

[22] W. A. Firmansyach, U. Hayati, and Y. A. Wijaya, "Analisa Terjadinya Overfitting dan Underfitting pada Algoritma Naive Bayes dan Decision Tree dengan Teknik Cross Validation," 2023.