 e-ISSN: 3110-8652

Penggunaan Model *Pre-trained Convolutional Neural Network* untuk Klasifikasi Makanan *Seafood*

http://dx.doi.org/10.28932/jste.v2i2.13160

Received: 26 Agustus 2025 | *Revised*: 27 April 2026 | *Accepted*: 29 Juni 2026

Michael Kuswanto[✉#1], Hendra Bunyamin[*2]

[#]*Program Studi Teknik Informatika, Fakultas Teknologi dan Rekayasa Cerdas, Universitas Kristen Maranatha*

Bandung, Jawa Barat 40164, Indonesia

[1]2172037@maranatha.ac.id

[*]*Program Studi Teknik Informatika, Fakultas Teknologi dan Rekayasa Cerdas, Universitas Kristen Maranatha*

Bandung, Jawa Barat 40164, Indonesia

[2]hendra.bunyamin@it.maranatha.edu

[✉]Corresponding author: 2172037@maranatha.ac.id

How to cite this article:

M. Kuswanto, H. Bunyamin, "The Use of Pre-trained Convolutional Neural Network Models for Seafood Classification," *Journal of Smart Technology and Engineering*, vol. 2, no. 2, pp. 84-100, 2026. https://doi.org/10.28932/jts.v2i2.13160

Abstrak* — Alergi terhadap seafood seperti udang, kepiting, kerang-kerangan, dan ikan merupakan pemicu umum reaksi alergi yang dapat menimbulkan risiko kesehatan yang serius. Untuk mengatasi tantangan dalam mengidentifikasi makanan yang mengandung alergen, penelitian ini mengembangkan sistem klasifikasi gambar berbasis Jaringan Saraf Konvolusional (CNN) dengan menggunakan arsitektur MobileNetV2 dan EfficientNetV2-S. Data dikumpulkan melalui web scraping dan diproses terlebih dahulu dengan cara mengubah ukuran, normalisasi, dan augmentasi. Model-model tersebut dilatih menggunakan bobot yang telah dilatih sebelumnya dengan hiperparameter yang disesuaikan secara manual, termasuk strategi dropout, regularisasi, dan fine-tuning. Evaluasi dilakukan menggunakan akurasi, presisi, recall, dan F1-Score. Model dengan kinerja terbaik, EfficientNetV2-S, mencapai akurasi 97,5%, presisi 97,59%, recall 97,5%, dan F1-Score 97,5%, serta menunjukkan stabilitas yang lebih baik dalam menghindari overfitting dibandingkan dengan MobileNetV2. Hiperparameter optimal meliputi tingkat dropout sebesar 0,5, nilai regularisasi L1 dan L2 masing-masing sebesar 0,01, ukuran batch sebesar 32, serta lapisan yang dibekukan. Model yang telah dilatih dikonversi ke format TensorFlow Lite dan diintegrasikan ke dalam aplikasi seluler FoodLergic. Pengujian akhir pada gambar baru dan melalui antarmuka seluler menunjukkan prediksi yang konsisten dan akurat. Temuan ini menunjukkan bahwa sistem ini layak sebagai solusi awal untuk mendeteksi alergen makanan melalui gambar pada perangkat seluler.

Kata Kunci— **Alergi makanan; *Convolutional Neural Network; EfficientNetV2-S; MobileNetV2; TensorFlow Lite.***

The Use of Pre-trained Convolutional Neural Network Models for Seafood Classification

Abstract — ***Seafood allergies such as shrimp, crab, shellfish, and fish are common triggers of allergic reactions that can pose serious health risks. To address the challenge of identifying foods containing allergens, this study developed an image***

classification system based on Convolutional Neural Networks (CNN) using MobileNetV2 and EfficientNetV2-S architectures. Data was collected through Web scraping and preprocessed via resizing, normalization, and augmentation. The models were trained using pretrained weights with manually tuned hyperparameters, including dropout, regularization, and fine-tuning strategies. Evaluation was conducted using accuracy, precision, recall, and F1-Score. The best-performing model, EfficientNetV2-S, achieved an accuracy of 97,5%, precision 97,59%, recall 97,5%, and F1-score 97,5%, and showed greater stability in avoiding overfitting compared to MobileNetV2. The optimal hyperparameters included a dropout rate of 0.5, L1 and L2 regularization values of 0.01 each, a batch size of 32, and frozen layers. The trained model was converted into TensorFlow Lite format and integrated into the FoodLergic mobile application. Final testing on new images and through the mobile interface demonstrated consistent and accurate predictions. These findings suggest the system is feasible as an initial solution for detecting food allergens through images on mobile devices.

Keywords— ***Convolutional Neural Network; EfficientNetV2-S; food allergy; MobileNetV2; TensorFlow Lite.***

I. PENDAHULUAN

Alergi makanan semakin sering dijumpai dalam masyarakat, dengan reaksi yang bisa berkisar dari gejala ringan hingga kondisi yang mengancam nyawa, seperti anafilaksis. Identifikasi alergen dalam makanan menjadi tantangan utama, terutama pada makanan olahan atau hidangan khas tertentu. Studi di Yogyakarta dan Jawa menunjukkan prevalensi alergi makanan pada kelompok usia dewasa mencapai 33%, dengan makanan laut seperti udang, kepiting, kerang, dan ikan menjadi penyebab utama [1]. Hal ini menunjukkan pentingnya adanya solusi untuk mengidentifikasi alergen dengan cepat dan akurat.

Seiring dengan perkembangan teknologi, kecerdasan buatan (AI) dapat menjadi solusi efektif untuk masalah ini. *Convolutional Neural Network* (CNN), sebagai metode *deep learning* yang mampu mengenali pola visual, digunakan untuk mengidentifikasi makanan laut yang berpotensi menyebabkan alergi. Dalam penelitian ini, model CNN diterapkan untuk mengklasifikasikan makanan laut berdasarkan gambar, dan hasilnya diintegrasikan dalam aplikasi *FoodLergic* yang memungkinkan individu dengan alergi makanan untuk memverifikasi keamanan makanan yang mereka konsumsi hanya dengan menggunakan perangkat seluler.

Tujuan penelitian ini adalah untuk mengembangkan *model deep learning* berbasis *MobileNetV2* dan *EfficientNetV2* dalam mengidentifikasi makanan laut yang berpotensi menyebabkan alergi, serta untuk membandingkan kinerja kedua model dalam hal akurasi dan efisiensi. Pemilihan *MobileNetV2* didasarkan pada kemampuannya dalam menghasilkan performa klasifikasi yang baik dengan kebutuhan komputasi yang relatif rendah, sehingga menjadikannya salah satu arsitektur yang paling banyak digunakan pada aplikasi berbasis perangkat *mobile*. Sementara itu, *EfficientNetV2* dipilih karena merupakan arsitektur yang lebih baru dan dirancang untuk meningkatkan efisiensi pelatihan serta akurasi model melalui optimasi struktur jaringan yang lebih efektif. Perbandingan kedua arsitektur ini diharapkan dapat memberikan gambaran mengenai *trade-off* antara akurasi dan efisiensi komputasi dalam tugas klasifikasi citra makanan laut. Selain itu, penelitian ini bertujuan untuk menerapkan model terbaik dalam aplikasi *mobile* berbasis *Android* dengan menggunakan *TensorFlow Lite* sehingga dapat digunakan secara praktis sebagai alat bantu identifikasi makanan yang berpotensi memicu reaksi alergi..

Manfaat dari penelitian ini adalah memberikan solusi yang efisien dan praktis untuk mengidentifikasi alergen, sehingga dapat meningkatkan keamanan konsumsi makanan bagi individu dengan alergi. Aplikasi yang dihasilkan dapat membantu pengguna untuk secara cepat mengetahui kandungan alergen dalam makanan hanya dengan mengambil gambar, memberikan rasa aman bagi mereka yang memiliki alergi makanan dan memperkecil risiko reaksi alergi yang berbahaya.

II. KAJIAN TEORI

A. Machine Learning (ML)

Machine Learning (ML) adalah cabang dari kecerdasan buatan (AI) yang memungkinkan sistem untuk belajar secara otomatis dari data, tanpa perlu pemrograman eksplisit. Dalam ML, algoritma dirancang untuk menganalisis data, menemukan pola, dan membuat prediksi berdasarkan data tersebut. Secara umum, ML

dibagi menjadi tiga kategori utama: *supervised learning, unsupervised learning*, dan *reinforcement learning. Supervised learning* menggunakan data berlabel untuk membangun model, sementara *unsupervised learning* mencari pola dalam data tanpa label. *Reinforcement learning* berfokus pada pelatihan agen untuk membuat keputusan dalam lingkungan yang dinamis berdasarkan penghargaan atau hukuman. Tantangan utama dalam ML adalah kebutuhan akan data berkualitas tinggi serta pemilihan algoritma yang tepat untuk menyelesaikan masalah tertentu, karena algoritma yang salah dapat menghasilkan model yang kurang akurat [2].

B. Deep learning

Deep learning adalah sub-bidang dari *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan (deep neural networks) untuk mempelajari representasi data yang kompleks. *Deep learning* sangat efektif dalam menangani data yang sangat besar dan rumit, seperti gambar, suara, dan teks. Algoritma *deep learning* dapat secara otomatis mengekstraksi fitur dari data mentah melalui proses pelatihan intensif tanpa memerlukan rekayasa fitur manual. Model seperti *Convolutional Neural Networks* (CNN) digunakan untuk pengolahan gambar, sedangkan *Recurrent Neural Networks* (RNN) digunakan untuk data sekuensial seperti teks atau sinyal waktu [3] [4]. *Deep learning* mengandalkan data dalam jumlah besar dan sumber daya komputasi yang tinggi, serta optimasi algoritma seperti *Adam* dan *RMSprop* untuk meningkatkan kecepatan pelatihan dan stabilitas model.

C. Convolutional Neural Networks (CNN) dan Teknik Optimasi

Convolutional Neural Networks (CNN) adalah jenis jaringan saraf tiruan yang dirancang khusus untuk memproses data berbentuk grid, seperti gambar atau sinyal waktu. CNN terdiri dari beberapa lapisan utama, di antaranya:

1) Convolutional Layer: Lapisan ini menggunakan *filter* atau *kernel* untuk melakukan operasi konvolusi pada data *input*. Setiap filter bertugas untuk mengekstrak fitur spesifik dari gambar, seperti garis, pola tekstur, atau bentuk. Setiap operasi konvolusi menghasilkan peta fitur (*feature map*) yang merepresentasikan informasi penting dari data.

2) Pooling Layer: Lapisan ini bertugas untuk mengurangi dimensi data, meningkatkan efisiensi komputasi, dan membuat model lebih *robust* terhadap variasi kecil dalam data, seperti rotasi atau translasi. Teknik *pooling* yang umum digunakan adalah *max pooling*, yang memilih nilai maksimum dari suatu area kecil dalam peta fitur.

3) Fully Connected Layer: Pada lapisan ini, informasi yang telah diekstraksi dari lapisan sebelumnya digabungkan untuk menghasilkan prediksi akhir. Biasanya, fungsi *activation* seperti *softmax* digunakan untuk mengubah *output* menjadi probabilitas yang mewakili kelas yang diprediksi [4] [5].

Untuk meningkatkan performa dan efisiensi model CNN, berbagai teknik optimasi diterapkan selama proses pelatihan. Teknik-teknik ini bertujuan untuk mempercepat pelatihan, meningkatkan stabilitas model, serta mencegah *overfitting*, sehingga model dapat bekerja dengan baik pada data yang belum pernah dilihat sebelumnya. Beberapa teknik optimasi yang digunakan dalam CNN antara lain:

1) Batch Normalization: Teknik ini digunakan untuk menormalkan *input* di setiap lapisan, yang bertujuan untuk mempercepat proses pelatihan dan meningkatkan stabilitas model. Dengan menormalkan *input*, model dapat belajar lebih cepat dan lebih stabil selama pelatihan, serta mengurangi risiko *vanishing gradient.*

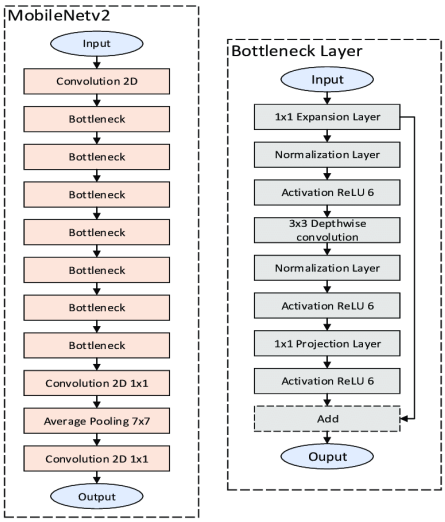
2) Dropout: Dropout menghapus beberapa *neuron* secara acak selama proses pelatihan. Teknik ini bertujuan untuk mencegah model terlalu bergantung pada neuron tertentu dan membantu mencegah *overfitting*, sehingga model dapat generalisasi dengan lebih baik.

3) Learning rate Scheduler: Teknik ini mengubah *learning rate* selama pelatihan untuk mempercepat konvergensi model. Dengan menurunkan *learning rate* setelah beberapa *epoch*, model dapat mencapai titik konvergensi lebih cepat dan lebih stabil, mencegah fluktuasi yang terlalu besar pada saat pelatihan.

Dengan menerapkan teknik-teknik optimasi ini, model CNN tidak hanya menjadi lebih efisien dalam hal komputasi, tetapi juga lebih handal dalam menghasilkan prediksi yang akurat, meskipun menghadapi data yang bervariasi dan kompleks [4] [5].

D. MobileNetV2

MobileNetV2 adalah arsitektur *deep learning* yang dirancang untuk perangkat dengan sumber daya terbatas, seperti perangkat seluler. Model ini menggunakan teknik *depthwise separable convolution*, yang memisahkan operasi konvolusi menjadi dua langkah: pertama, melakukan konvolusi pada setiap saluran (*depthwise*), dan kedua, menggabungkan hasilnya dengan konvolusi 1x1 (*pointwise*). Teknik ini mengurangi jumlah parameter dan meningkatkan efisiensi komputasi tanpa mengorbankan performa. *MobileNetV2* menggabungkan *bottleneck layers* untuk meningkatkan representasi fitur dengan parameter yang lebih sedikit, menjadikannya sangat efisien untuk aplikasi berbasis perangkat seluler [6].



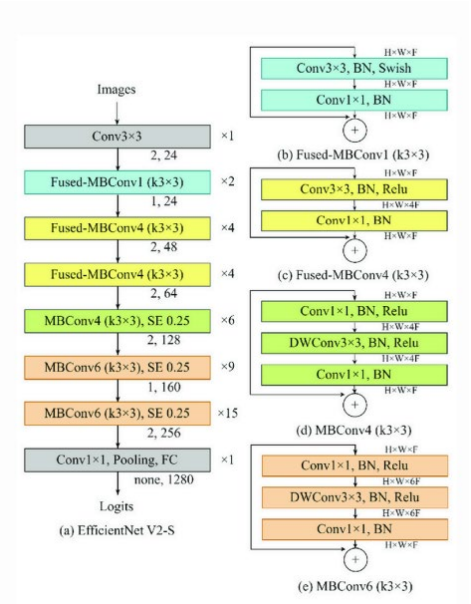
Gambar 1. Arsitektur *MobileNetV2* [7]

Pada Gambar 1, dapat dilihat Arsitektur *MobileNetV2*, yang terdiri dari lapisan *input*, konvolusi awal, beberapa blok *bottleneck*, serta lapisan klasifikasi akhir. Setiap komponen ini berfungsi untuk mengekstraksi dan mengubah fitur gambar menjadi representasi yang lebih efisien dan dapat digunakan untuk prediksi akhir. Gambar ini menunjukkan aliran data dari *input*

E. EfficientNetV2

EfficientNetV2 adalah model *deep learning* yang dikembangkan oleh *Google Research*, yang mengoptimalkan efisiensi pelatihan dan akurasi dengan menggunakan teknik *compound scaling*. *Compound scaling* mengoptimalkan tiga dimensi arsitektur model secara bersamaan: kedalaman (*depth*), lebar (*width*), dan resolusi *input*. Teknik ini memungkinkan model untuk mencapai performa terbaik dengan parameter yang lebih sedikit dibandingkan model lain tanpa mengorbankan akurasi. *EfficientNetV2* juga mengimplementasikan teknik pelatihan yang lebih efisien, sehingga lebih cepat dan lebih optimal dibandingkan model-model sebelumnya. Salah satu keunggulannya adalah pengurangan waktu pelatihan dan penggunaan memori yang lebih efisien [8].

 e-ISSN: 3110-8652



Gambar 2. Arsitektur *EfficientNetV2* [9]

Pada Gambar 2, dapat dilihat Arsitektur *EfficientNetV2*, yang memiliki beberapa lapisan utama seperti *Fused-MBConv1, Fused-MBConv4*, dan *MBConv4* dengan *Squeeze-and-Excitation* (SE). Gambar ini memberikan gambaran visual tentang bagaimana model ini menyusun lapisan-lapisan untuk mengekstraksi fitur dan mengoptimalkan representasi dalam *dataset*. Teknik *compound scaling* yang diterapkan di sini terlihat jelas pada bagaimana model ini mengatur kedalaman, lebar, dan resolusi *input* secara bersamaan, untuk mencapai keseimbangan optimal antara efisiensi dan akurasi komputasi.

F. Cross-validation

Cross-validation adalah teknik evaluasi yang digunakan untuk menilai performa model. Teknik ini membagi *dataset* menjadi beberapa bagian (*folds*) dan melatih model dengan menggunakan k bagian dan mengujinya dengan bagian lainnya. *K-fold Cross-validation* adalah salah satu jenis yang paling umum digunakan, di mana *dataset* dibagi menjadi k bagian yang sama besar, dan model dilatih k kali dengan setiap bagian digunakan sekali sebagai data uji. Teknik ini mencegah *overfitting* dan memberikan estimasi yang lebih akurat mengenai kemampuan model dalam menggeneralisasi pada data yang belum terlihat [5].

G. TensorFlow Framework

TensorFlow adalah *framework open-source* yang dikembangkan oleh Google untuk membangun dan melatih model ML dan *deep learning* . *TensorFlow* mendukung pelatihan terdistribusi, yang memungkinkan model dilatih secara paralel pada beberapa GPU atau TPU, mempercepat proses pelatihan. *TensorFlow* juga dilengkapi dengan *TensorFlow Lite* untuk pengembangan model pada perangkat *mobile*, serta alat visualisasi seperti *TensorBoard* untuk memantau metrik pelatihan secara real-time. *TensorFlow* memfasilitasi pengembangan model dengan API tingkat tinggi seperti Keras dan mendukung penggunaan model yang telah dilatih sebelumnya melalui *TensorFlow Hub [10] [11] [12] [13]*.

H. Augmentasi Data

Augmentasi data adalah teknik yang digunakan untuk memperbanyak variasi dalam data pelatihan dengan memodifikasi data asli. Teknik ini mencakup rotasi gambar, *flipping* (membalik gambar), *zooming*, dan perubahan kecerahan. Augmentasi membantu model untuk belajar dari variasi data yang lebih luas, mengurangi risiko *overfitting* dan meningkatkan kemampuan generalisasi model [6]. Augmentasi data juga dapat dilakukan secara dinamis selama pelatihan menggunakan *TensorFlow ImageDataGenerator*, yang memastikan bahwa model belajar dari variasi yang berbeda setiap *epoch*.

I. RegularisasiModel

Regularisasi adalah teknik yang digunakan untuk mencegah *overfitting* dengan menambahkan penalti pada parameter model yang kompleks. Teknik regulasi yang umum digunakan antara lain:

1) L1 Regularization (Lasso): Menambahkan penalti berdasarkan jumlah absolut nilai parameter, yang dapat menghapus parameter tertentu menjadi nol, menyederhanakan model.

2) L2 Regularization (Ridge): Menambahkan penalti berbasis kuadrat nilai parameter, menjaga bobot model tetap kecil namun tidak menghapusnya sepenuhnya.

3) Dropout: Mengabaikan sejumlah neuron secara acak selama pelatihan untuk meningkatkan generalisasi model [2] [3].

J. Scraping Gambar

Web scraping adalah teknik yang digunakan untuk mengambil data dari situs web secara otomatis. Dalam penelitian ini, *scraping* gambar digunakan untuk mengumpulkan *dataset* gambar makanan laut dengan memanfaatkan Selenium untuk mengunduh gambar secara otomatis dari halaman web yang memuat konten dinamis menggunakan *JavaScript* [14]. Teknik ini memungkinkan pengambilan gambar dari berbagai sumber seperti Google dan Bing, yang digunakan untuk pelatihan model klasifikasi gambar.

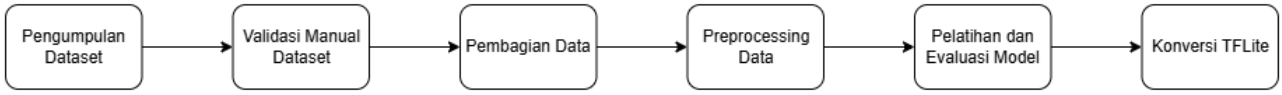
K. Validasi Data

Validasi data memastikan kualitas *dataset* yang digunakan dalam pelatihan model. Proses validasi meliputi pemeriksaan manual untuk memastikan gambar relevan dengan kategori yang diinginkan, serta pengecekan kualitas gambar seperti resolusi dan kejelasan. Setelah gambar diverifikasi, data diubah ke format RGB dan di-*resize* menjadi 224x224 *pixel* sesuai dengan standar *input* untuk model CNN, yang meningkatkan akurasi dan performa model [2] [5].

III. ANALISIS DAN RANCANGAN SISTEM

A. Metodologi Penelitian

Penelitian ini bertujuan untuk mengembangkan model *deep learning* berbasis *Convolutional Neural Network* (CNN) guna mengklasifikasikan makanan *seafood* yang berpotensi menyebabkan alergi, seperti udang, kepiting, kerang, dan ikan. Dua arsitektur model yang digunakan dalam penelitian ini adalah *MobileNetV2* dan *EfficientNetV2-S*, yang keduanya dioptimalkan untuk efisiensi komputasi dan akurasi.



Gambar 3. Diagram Metodologi Penelitian

Pendekatan yang digunakan dalam penelitian ini mengikuti beberapa tahapan utama. Setiap tahapan tersebut dirancang untuk memastikan efektivitas dan efisiensi dalam pengumpulan dan pemrosesan data, serta dalam pengembangan dan evaluasi model *deep learning* yang diusulkan. Menurut Gambar 3, tahapan utama yang diikuti dalam penelitian ini adalah sebagai berikut:

1) Pengumpulan Data: melalui teknik *Web scraping.*

2) Validasi Manual Dataset: untuk memastikan gambar yang relevan dan berkualitas.

3) Pembagian Data: dengan rasio 80:20 dan penggunaan *K-fold Cross-validation* untuk menghindari *overfitting.*

4) Preprocessing Data: yang mencakup resizing gambar, normalisasi, dan augmentasi data.

5) Pelatihan Model: dengan optimasi dan evaluasi menggunakan metrik seperti *accuracy*, *precision*, *recall*, *F1-Score*, dan *confusion matrix.*

6) Konversi ke TensorFlow Lite (TFLite): agar model dapat digunakan pada perangkat dengan sumber daya terbatas.

B. Pengumpulan Data

Dataset yang digunakan terdiri dari gambar makanan *seafood* yang dikumpulkan melalui teknik *Web scraping* dari sumber seperti *Google Images* dan *Bing Images*. Setiap kategori makanan (udang, kepiting, kerang, dan ikan) terdiri dari 300 gambar, dengan total *dataset* sebanyak 1.200 gambar. Validasi manual dilakukan untuk memastikan gambar yang digunakan relevan dan berkualitas tinggi.

C. Validasi Manual Dataset

Setelah pengumpulan data, dilakukan validasi manual untuk memastikan setiap gambar sesuai dengan label kategori dan bebas dari gangguan visual seperti *watermark*, *logo*, atau gambar yang buram. Proses ini sangat penting untuk memastikan kualitas *dataset* yang akan digunakan dalam pelatihan model.

D. Pembagian Data

Dataset dibagi menggunakan *train-test split* dengan rasio 80:20, di mana 80% digunakan untuk pelatihan dan 20% untuk pengujian. Data pelatihan kemudian dibagi lagi dengan menggunakan 5-*fold Cross-validation* untuk memastikan model tidak overfit dan dapat menggeneralisasi dengan baik pada data yang belum terlihat.

E. Preprocessing Data

Sebelum digunakan untuk pelatihan model, gambar-gambar dalam *dataset* perlu diproses melalui beberapa tahapan untuk memastikan data tersebut siap dan optimal untuk diproses oleh model CNN. Proses *preprocessing* bertujuan untuk meningkatkan kualitas data dan memastikan model dapat belajar secara efektif. Berikut adalah tahapan *preprocessing* yang dilakukan:

1) Resize Gambar: Semua gambar yang dikumpulkan dalam *dataset* diubah ukurannya menjadi 224x224 *pixel*, sesuai dengan *input* yang dibutuhkan oleh kedua model, yaitu *MobileNetV2* dan *EfficientNetV2-S*. Proses ini memastikan bahwa semua gambar memiliki ukuran yang konsisten, sehingga dapat diproses secara efisien oleh model.

2) Normalisasi: Setiap nilai *pixel* gambar dinormalisasi ke dalam skala [0, 1]. Proses normalisasi ini penting untuk mempercepat pelatihan model karena membantu dalam stabilitas gradien selama optimasi. Normalisasi juga menghindari dominasi fitur tertentu yang dapat mengganggu proses pelatihan.

3) Augmentasi Data: Augmentasi data adalah teknik untuk meningkatkan variasi dalam *dataset* dengan mentransformasikan gambar yang ada, tanpa menambah data baru secara fisik. Teknik ini mencakup rotasi gambar untuk mengenali objek dari berbagai arah, *flipping* untuk variasi orientasi, perubahan kecerahan untuk mensimulasikan pencahayaan yang berbeda, dan *zooming* untuk mengenali objek pada berbagai skala. Tujuan dari augmentasi ini adalah untuk meningkatkan keberagaman data pelatihan, membantu model belajar mengenali objek dalam kondisi yang lebih bervariasi.

4) Cross-validation: Setelah proses augmentasi, data dibagi menggunakan *K-fold Cross-validation*. Teknik ini membagi *dataset* menjadi beberapa bagian (*folds*) dan menggunakan setiap *fold* sebagai data uji sekali, sementara *fold* lainnya digunakan untuk pelatihan. *Cross-validation* membantu mengurangi risiko *overfitting* dengan memastikan model diuji pada berbagai *subset* data yang berbeda, memberikan gambaran yang lebih realistis tentang performa model pada data yang tidak terlihat sebelumnya.

F. Pelatihan Model

Proses pelatihan dilakukan dengan *optimizer Adam* dan *categorical cross-entropy* sebagai *loss function*. Parameter pelatihan termasuk *batch size* 32, *epoch*s 50, dan *learning rate* 0.0001. Metrik yang digunakan untuk mengevaluasi model adalah *accuracy*, *precision*, *recall*, dan *F1-Score*, yang memastikan model tidak hanya akurat tetapi juga stabil dalam menangani ketidakseimbangan kelas.

 e-ISSN: 3110-8652

G. Evaluasi Model

Setelah model dilatih menggunakan *dataset* yang telah diproses, evaluasi dilakukan untuk mengukur kinerja model secara objektif dan stabil. Evaluasi model sangat penting untuk mengetahui seberapa baik model dapat menggeneralisasi pada data baru. Berikut adalah metrik yang digunakan dalam evaluasi model:

1) Confusion matrix: Menyediakan informasi tentang distribusi prediksi model terhadap kelas yang benar, mencakup *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Metrik lainnya, seperti *precision*, *recall*, dan *F1-Score*, dihitung berdasarkan *confusion matrix* untuk memberikan gambaran yang lebih lengkap tentang performa model.

2) Akurasi: Mengukur persentase prediksi yang benar dibandingkan dengan total prediksi yang dilakukan. Memberikan gambaran umum tentang seberapa baik model memprediksi kelas yang benar.

3) Precision dan Recall: Precision mengukur seberapa banyak prediksi positif yang benar dibandingkan dengan total prediksi positif. *Recall* mengukur seberapa banyak sampel positif yang berhasil diidentifikasi dengan benar oleh model.

4) F1-Score: Rata-rata harmonis antara *precision* dan *recall*, digunakan untuk menilai keseimbangan antara keduanya, terutama ketika *dataset* tidak seimbang.

5) Loss & Accuracy Plot: Memvisualisasikan perubahan nilai *loss* dan *accuracy* selama pelatihan untuk memantau stabilitas model dan mendeteksi kemungkinan *overfitting* atau *underfitting*.

Evaluasi ini memberikan pemahaman mendalam tentang seberapa efektif model dalam mengenali pola pada data yang belum terlihat sebelumnya.

H. Konversi Model Menjadi TensorFlow Lite

Setelah model dilatih, ia dikonversi ke format *TensorFlow Lite* (TFLite) untuk implementasi pada perangkat *mobile*. Proses konversi mencakup optimasi menggunakan *quantization* untuk mengurangi ukuran model tanpa mengorbankan performa signifikan. Model TFLite diuji untuk memastikan bahwa hasil klasifikasi tetap akurat setelah konversi, memungkinkan penerapan pada perangkat dengan sumber daya terbatas.

IV. PENGUJIAN

A. Eksperimen Hyperparameter tuning

Proses *hyperparameter tuning* dilakukan untuk mengoptimalkan performa model dalam klasifikasi gambar *seafood* dengan mencari kombinasi nilai *dropout*, regularisasi L1 dan L2, *layer freeze*, dan *batch size* yang paling efektif. Eksperimen ini dilakukan secara terpisah untuk dua model yang diuji, yaitu *MobileNetV2* dan *EfficientNetV2-S*.

1) Eksperimen Hyperparameter tuning pada MobileNetV2: Pada eksperimen ini, dilakukan pengujian beberapa kombinasi hyperparameter untuk model *MobileNetV2*. Pengaturan yang diuji meliputi *dropout*, regularisasi L1 dan L2, *layer freeze*, dan *batch size*.

TABEL I
HYPERPARAMETER TUNING MOBILENETV2

Drop Out	L1	L2	Layer *Unfreeze*	*Batch size*	*MobileNetV2*				
					Accuracy	Precison	*Recall*	*F1-Score*	Keterangan
0.5	No	0.01	Yes	32	77,5 %	80 %	76 %	77 %	Overfit
0.75	No	0.01	Yes	32	76 %	78 %	75 %	76,5 %	Overfit
0.5	0.01	0.01	Yes	32	78 %	79,5 %	77,5 %	78 %	Overfit
0.75	0.01	0.01	Yes	32	76,5 %	79 %	76,5 %	77 %	Overfit
0.8	0.01	0.01	No	16	79,33 %	81,5 %	79,33 %	79.67 %	Overfit

 e-ISSN: 3110-8652

Drop Out	L1	L2	Layer *Unfreeze*	*Batch size*	*MobileNetV2*				
					Accuracy	Precison	*Recall*	*F1-Score*	Keterangan
0.8	0.1	0.1	No	16	80,33 %	83,45 %	80,33 %	80,66 %	Overfit
Setelah Perubahan *Dataset*									
0.8	0.1	0.1	No	16	**94,27 %**	**94,66 %**	**94,27 %**	**94,29 %**	**Underfit**
0.5	0.01	0.01	No	32	93,44 %	94.02 %	93,44 %	93,38 %	Overfit

Hasil eksperimen yang menunjukkan pengaturan terbaik untuk model *MobileNetV2* dapat dilihat dalam Tabel I, yang mencakup pengaturan *dropout* sebesar 0.8, L1 dan L2 *regularization* sebesar 0.1, *layer freeze* (*unfreeze = No*), dan *batch size* 16. Model dengan pengaturan ini menghasilkan akurasi: 94,27%, *precision*: 94,66%, *recall*: 94,27%, dan *F1-Score*: 94,29%. Meskipun performanya cukup baik, model *MobileNetV2* menunjukkan indikasi *underfitting*, yang tercermin dari perbedaan antara akurasi data pelatihan yang lebih rendah dan akurasi data validasi yang lebih tinggi.

2) Eksperimen Hyperparameter tuning pada EfficientNetV2-S: Untuk model *EfficientNetV2-S*, eksperimen dilakukan dengan pengaturan yang serupa, tetapi pengaturan terbaik ditemukan pada *dropout* 0.5, L1 dan L2 *regularization* sebesar 0.01, *layer freeze* (*unfreeze = No*), dan *batch size* 32.

TABEL II
HYPERPARAMETER TUNING EFFICIENTNETV2-S

Drop Out	L1	L2	Layer *Unfreeze*	*Batch size*	*EfficientNetV2-S*				
					Accuracy	*Precision*	*Recall*	*F1-Score*	Keterangan
0.5	*No*	0.01	Yes	32	78 %	81 %	75 %	78 %	Overfit
0.75	*No*	0.01	Yes	32	77,5 %	79,5 %	74,5 %	76,5 %	Overfit
0.5	0.01	0.01	Yes	32	78,5 %	80 %	77 %	78,5 %	Overfit
0.75	0.01	0.01	Yes	32	77,8 %	79,8 %	76 %	77,5 %	Overfit
0.8	0.01	0.01	*No*	16	81,5 %	83 %	81,5 %	81.83 %	Overfit
0.8	0.1	0.1	*No*	16	82,58 %	85,42 %	82,58 %	82,94 %	*No*rmal
Setelah Perubahan *Dataset*									
0.8	0.1	0.1	*No*	16	96,77 %	96,89 %	96,77 %	96,78 %	Underfit
0.5	0.01	0.01	*No*	32	**97,5 %**	**97,59 %**	**97,5 %**	**97,5 %**	***No*rmal**

Hasil eksperimen untuk *EfficientNetV2-S* dapat dilihat dalam Tabel II, yang menunjukkan hasil sebagai berikut akurasi: 97,5%, *precision*: 97,59%, *recall*: 97,5%, dan *F1-Score*: 97,5%. Model *EfficientNetV2-S* menunjukkan hasil yang sangat baik, bebas dari *overfitting* atau *underfitting*, dengan peningkatan stabilitas pada *dataset* yang digunakan.

TABEL III
PERBANDINGAN PERFORMA KEDUA MODEL

	Accuracy	*Precision*	*Recall*	*F1-Score*	Keterangan
MobileNetV2	**94,27 %**	**94,66 %**	**94,27 %**	**94,29 %**	**Underfit**
EfficientNetV2-S	**97,5 %**	**97,59 %**	**97,5 %**	**97,5 %**	***No*rmal**

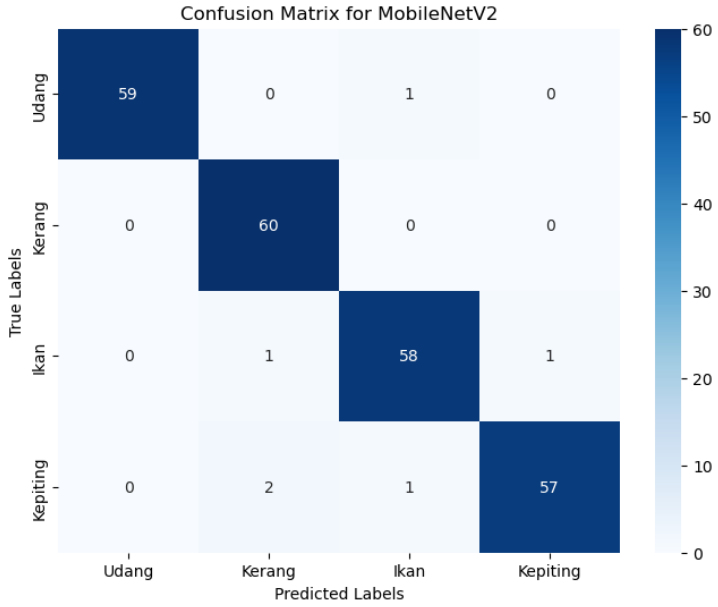
Setelah melakukan eksperimen *hyperparameter tuning* untuk kedua model, pada Tabel III terlihat perbandingan antara hasil eksperimen untuk *MobileNetV2* dan *EfficientNetV2-S* menunjukkan perbedaan yang signifikan dalam hal akurasinya. *MobileNetV2* menghasilkan akurat 94,27%, dengan beberapa indikasi *underfitting* pada *dataset* yang lebih besar dan lebih beragam. Di sisi lain, *EfficientNetV2-S* menunjukkan hasil yang jauh lebih unggul dengan akurat 97,5%, serta stabilitas performa yang lebih baik tanpa mengalami masalah *overfitting* atau *underfitting*. Dengan demikian, *EfficientNetV2-S* terbukti lebih unggul dibandingkan *MobileNetV2*, menjadikannya pilihan yang lebih optimal untuk aplikasi klasifikasi gambar *seafood* yang membutuhkan akurasi tinggi dan kemampuan generalisasi yang lebih baik.

B. Hasil Evaluasi Model

 e-ISSN: 3110-8652

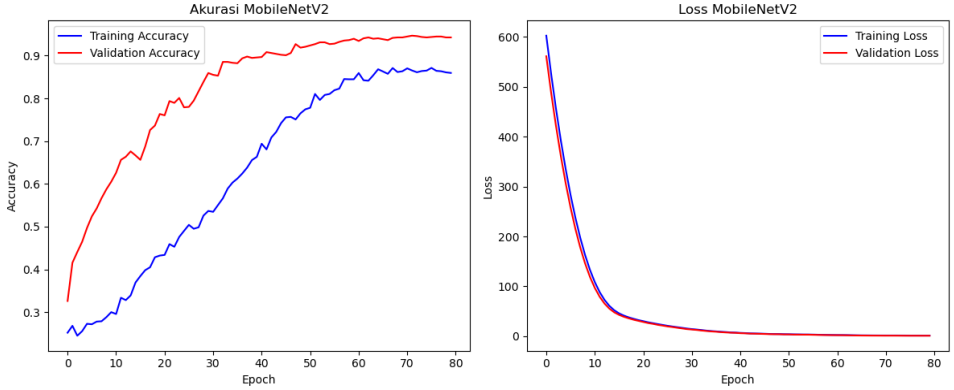
Pada bagian ini, dilakukan evaluasi terhadap performa dua model *deep learning* yang dikembangkan, yaitu *MobileNetV2* dan *EfficientNetV2-S*, dalam mengklasifikasikan gambar *seafood* yang berpotensi menyebabkan alergi. Evaluasi dilakukan dengan menggunakan beberapa metrik utama yang digunakan untuk menilai kualitas prediksi model, termasuk *Confusion matrix*, *Loss & Accuracy Plot*, dan *Classification report*. Melalui analisis ini, diharapkan dapat diperoleh gambaran yang jelas mengenai seberapa baik masing-masing model dalam mengenali dan mengklasifikasikan gambar *seafood* , serta bagaimana model-model tersebut berperforma dalam kondisi yang lebih bervariasi setelah perbaikan *dataset*.

1) Evaluasi Model MobileNetV2: Pada bagian ini, dilakukan evaluasi terhadap model *MobileNetV2* menggunakan beberapa metrik untuk mengukur kinerja model dalam mengklasifikasikan gambar *seafood* . Tiga metrik utama yang dianalisis adalah *Confusion matrix*, *Loss & Accuracy Plot*, dan *Classification report*.



Gambar 4. *Confusion matrix MobileNetV2*

Gambar 4 menunjukkan *confusion matrix* untuk *MobileNetV2*, yang menggambarkan distribusi prediksi model terhadap kelas yang benar. *Confusion matrix* ini memberikan informasi tentang jumlah true positive (TP), false positive (FP), true negative (TN), dan false negative (FN). Dari matrix ini, kita dapat menilai performa model dalam mengklasifikasikan gambar Udang, Kerang, Ikan, dan Kepiting, serta melihat tingkat kesalahan yang terjadi.



Gambar 5. *Accuracy & Loss Plot MobileNetV2*

Gambar 5 menunjukkan *Loss & Accuracy Plot* untuk *MobileNetV2*, yang menggambarkan bagaimana nilai *loss* dan *accuracy* berubah selama pelatihan. Grafik ini memberikan wawasan tentang apakah model

 e-ISSN: 3110-8652

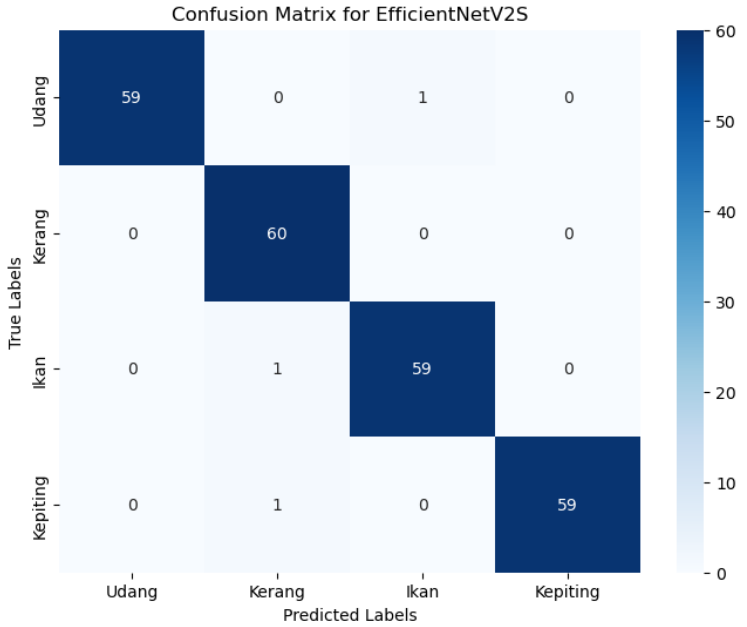
mengalami *underfitting* atau *overfitting*. Jika terdapat perbedaan besar antara akurasi pelatihan dan validasi, ini menunjukkan bahwa model mungkin tidak cukup terlatih atau gagal menggeneralisasi dengan baik. Grafik *loss* menunjukkan penurunan yang stabil, yang menunjukkan bahwa model belajar secara efisien.

TABEL IV
CLASSIFICATION REPORT MOBILENETV2

Classification report MobileNetV2				
	Precision	*Recall*	*F1-Score*	Support
Udang	100%	98%	99%	60
Kerang	95%	100%	98%	60
Ikan	97%	97%	97%	60
Kepiting	98%	95%	97%	60
Accuracy	97%			240
Macro Avg	98%	98%	97%	240
Weighted Avg	98%	97%	97%	240

Tabel IV menampilkan *classification report* untuk *MobileNetV2*, yang memberikan nilai *precision*, *recall*, *F1-Score*, dan support untuk setiap kelas. Metrik ini memberikan gambaran lebih rinci mengenai performa model dalam mengklasifikasikan gambar *seafood* . *Precision* menunjukkan seberapa banyak prediksi positif yang benar, sementara *recall* mengukur seberapa banyak kasus positif yang berhasil diidentifikasi oleh model. *F1-Score* adalah rata-rata harmonis antara *precision* dan *recall*, memberikan gambaran keseimbangan keduanya.

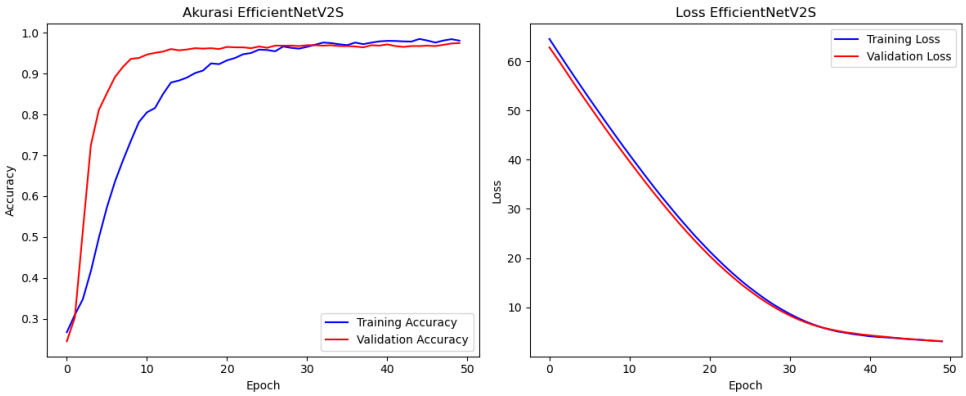
2) Evaluasi Model EfficientNetV2-S: Selanjutnya, evaluasi untuk *EfficientNetV2-S* dilakukan dengan menggunakan metrik yang sama untuk memberikan gambaran yang komprehensif mengenai kinerja model ini dalam klasifikasi gambar *seafood* .



Gambar 6. *Confusion matrix EfficientNetV2-S*

Gambar 6 menunjukkan *confusion matrix* untuk *EfficientNetV2-S*, yang memberikan gambaran yang lebih jelas tentang seberapa baik model ini mengklasifikasikan gambar *seafood* ke dalam setiap kelas. Dengan *confusion matrix* ini, kita dapat melihat jumlah prediksi yang benar (TP) dan kesalahan prediksi (FP, FN), serta bagaimana model menangani masing-masing kelas.

 e-ISSN: 3110-8652



Gambar 7. *Accuracy & Loss Plot EfficientNetV2-S*

Gambar 7 menunjukkan *Loss & Accuracy Plot* untuk *EfficientNetV2-S*, yang menggambarkan perubahan *loss* dan *accuracy* selama pelatihan. Grafik ini menunjukkan stabilitas model yang lebih baik dibandingkan dengan *MobileNetV2*. *Loss* dan *accuracy* pada data pelatihan dan validasi menunjukkan peningkatan yang konsisten, tanpa adanya indikasi *overfitting* atau *underfitting*. Grafik ini memperlihatkan bahwa *EfficientNetV2-S* bekerja lebih stabil selama pelatihan.

TABEL V
CLASSIFICATION REPORT EFFICEINTNETV2-S

Classification report EfficientNetV2-S				
	Precision	*Recall*	*F1-Score*	Support
Udang	100%	98%	99%	60
Kerang	97%	100%	98%	60
Ikan	98%	98%	98%	60
Kepiting	100%	98%	99%	60
Accuracy	99%			240
Macro Avg	99%	99%	99%	240
Weighted Avg	99%	99%	99%	240

Tabel V menampilkan *classification report* untuk *EfficientNetV2-S*, yang memberikan nilai *precision*, *recall*, *F1-Score*, dan support untuk setiap kelas. Hasil ini menunjukkan performa model dalam menangani gambar *seafood* dengan akurasi tinggi. *Precision* dan *recall* yang sangat tinggi pada masing-masing kelas menunjukkan bahwa *EfficientNetV2-S* memiliki kemampuan luar biasa dalam mengidentifikasi gambar *seafood* dengan akurat.

Dari hasil evaluasi yang dilakukan pada kedua model, dapat disimpulkan bahwa *EfficientNetV2-S* menunjukkan performa yang lebih unggul dibandingkan dengan *MobileNetV2*. *EfficientNetV2-S* menghasilkan nilai akurasi, *precision*, *recall*, dan *F1-Score* yang lebih tinggi pada semua metrik utama. Model ini berhasil menghindari *overfitting* atau *underfitting*, yang menjadikannya lebih stabil dan handal dalam mengklasifikasikan gambar *seafood* dengan tingkat akurasi yang tinggi. Sebaliknya, *MobileNetV2*, meskipun menunjukkan performa yang baik, masih menunjukkan indikasi *underfitting* yang terlihat dari perbedaan signifikan antara akurasi pada data pelatihan yang lebih rendah dan akurasi pada data validasi yang lebih tinggi. Secara keseluruhan, *EfficientNetV2-S* terbukti lebih baik dalam menangani kompleksitas data dan memberikan hasil yang lebih konsisten, menjadikannya pilihan yang lebih tepat untuk aplikasi klasifikasi gambar *seafood* .

C. Analisis Perbandingan Kesalahan

Setelah melakukan eksperimen, analisis dilakukan untuk membandingkan kesalahan prediksi antara model *MobileNetV2* dan *EfficientNetV2-S*. Meskipun kedua model menunjukkan peningkatan signifikan

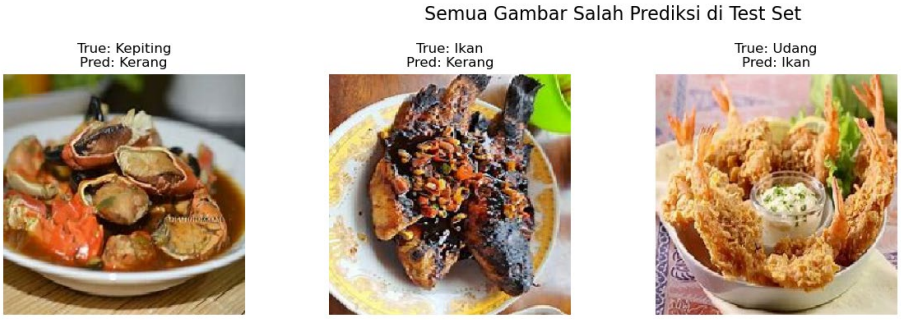
 e-ISSN: 3110-8652

setelah perbaikan *dataset*, terdapat perbedaan dalam jumlah kesalahan antara kedua model. *EfficientNetV2-S* mencatatkan hanya 3 kesalahan, sementara *MobileNetV2* mencatatkan 6 kesalahan.

1) Kesalahan Prediksi MobileNetV2: Pada Gambar 8, terlihat kesalahan prediksi yang dilakukan oleh *MobileNetV2*. Model ini cenderung mengklasifikasikan gambar Kepiting sebagai Kerang, dan beberapa gambar Udang salah diklasifikasikan sebagai Ikan. Kesalahan ini kemungkinan besar disebabkan oleh kemiripan visual antar kelas, terutama pada gambar-gambar yang memiliki fitur yang cukup mirip, seperti Kepiting yang memiliki bentuk cangkang yang mirip dengan Kerang dan gambar Udang yang berkelompok, membentuk pola yang mirip dengan Ikan.

Gambar 8. Mispredicted *MobileNetV2*

2) Kesalahan Prediksi EfficientNetV2-S: Pada Gambar 9, terlihat kesalahan prediksi yang dilakukan oleh *EfficientNetV2-S*. Meskipun jumlah kesalahan jauh lebih sedikit, masih terdapat beberapa prediksi yang keliru, seperti Ikan yang salah diklasifikasikan sebagai Kerang. Dalam hal ini, kemungkinan kesalahan disebabkan oleh adanya kesamaan visual atau ambiguitas dalam pengelompokan objek dalam gambar yang dapat membingungkan model.



Gambar 9. Mispredicted *EfficientNetV2-S*

Dari kedua gambar tersebut, terlihat bahwa meskipun *EfficientNetV2-S* lebih unggul dalam hal akurasi dan jumlah kesalahan yang lebih sedikit, kedua model masih menghadapi tantangan dalam mengklasifikasikan gambar dengan beberapa pola visual yang serupa atau objek yang berada dalam formasi yang dapat membingungkan model. Sebagai contoh, kesalahan antara Kepiting dan Kerang dapat dimengerti mengingat bahwa kedua objek tersebut memiliki ciri fisik yang mirip, terutama ketika sudut pengambilan gambar menyerupai satu sama lain. Demikian pula, kesalahan antara Udang dan Ikan mungkin lebih dipengaruhi oleh cara objek-objek tersebut berkelompok dalam gambar, menciptakan pola yang mirip secara visual.

Analisis lebih lanjut menunjukkan bahwa untuk mengatasi tantangan ini, pendekatan lanjutan seperti augmentasi sudut pandang, rotasi, dan segmentasi objek dalam gambar dapat digunakan untuk meningkatkan akurasi model dalam mengidentifikasi objek yang memiliki kemiripan visual atau pola yang tidak terduga.

 e-ISSN: 3110-8652

D. Hasil Prediksi Gambar Baru

Pada bagian ini, saya menguji kedua model (*MobileNetV2* dan *EfficientNetV2-S*) dengan 8 gambar baru yang terdiri dari 2 gambar untuk masing-masing kelas (Ikan, Kepiting, Kerang, dan Udang). Tujuan dari pengujian ini adalah untuk mengevaluasi bagaimana kedua model melakukan prediksi pada data yang tidak terlihat sebelumnya. Setiap model diminta untuk mengklasifikasikan gambar baru tersebut ke dalam salah satu dari empat kelas yang sudah didefinisikan: Ikan, Kepiting, Kerang, dan Udang.

TABEL VI
HASIL PREDIKSI GAMBAR BARU

Gambar	Prediksi	
	MobileNetV2	*EfficientNetV2-S*
	Ikan	Ikan
	Ikan	Ikan
	Kepiting	Kepiting
	Kepiting	Kepiting
	Kerang	Kerang
	Kerang	Kerang
	Ikan	Ikan
	Ikan	Ikan

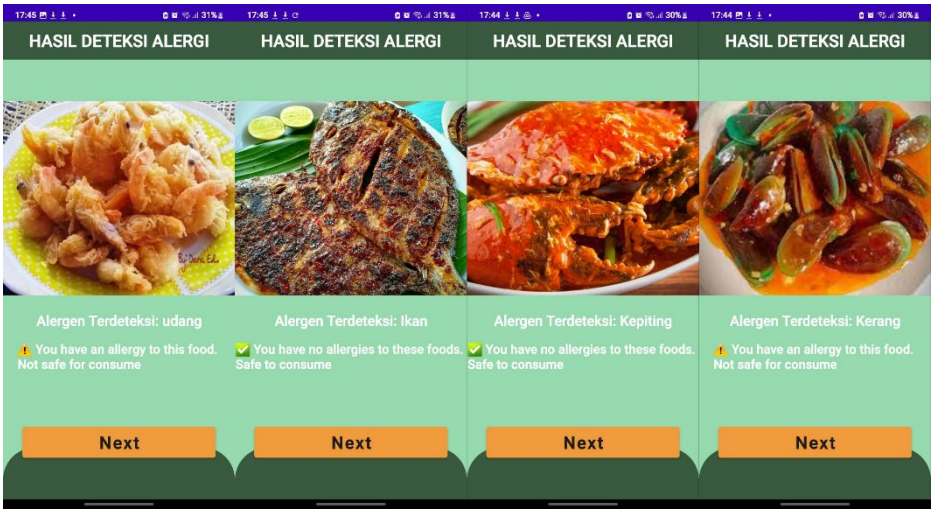
Berdasarkan Tabel VI, kedua model *MobileNetV2* dan *EfficientNetV2-S* menunjukkan performa yang sangat baik, dengan semua gambar Ikan, Kepiting, dan Kerang diprediksi dengan benar. Kedua model ini berhasil mengklasifikasikan gambar *seafood* secara akurat tanpa kesalahan prediksi.

EfficientNetV2-S dan *MobileNetV2* keduanya menunjukkan kestabilan dalam mengidentifikasi setiap kelas gambar, menunjukkan bahwa kedua model memiliki kemampuan yang baik dalam mengenali pola gambar *seafood* secara konsisten. Tabel di atas mencerminkan akurasi yang tinggi pada kedua model, yang dapat diandalkan dalam tugas klasifikasi gambar *seafood*.

 e-ISSN: 3110-8652

E. Hasil Prediksi Aplikasi Mobile

Setelah model terbaik, yaitu *EfficientNetV2-S*, dikonversi menjadi format *TensorFlow Lite* (TFLite), pengujian selanjutnya dilakukan pada aplikasi *mobile FoodLergic* untuk memastikan model dapat berjalan dengan baik di perangkat seluler. Model TFLite ini diintegrasikan ke dalam aplikasi, memungkinkan pengguna untuk melakukan klasifikasi gambar makanan secara langsung melalui kamera atau dengan mengunggah gambar dari galeri. Pengujian dilakukan dengan menggunakan gambar makanan dari kelas yang berfokus pada alergi, yaitu udang, kepiting, ikan, dan kerang. Gambar diambil dari kamera atau galeri ponsel, dan hasil prediksi menunjukkan bahwa model dapat mengklasifikasikan gambar dengan benar sesuai dengan kelas yang ditentukan.



Gambar 10. Prediksi Makanan dari Aplikasi *Mobile*

Gambar 10 menunjukkan hasil prediksi dari keempat gambar uji, yang diambil dari aplikasi. Model berhasil mengenali gambar udang, kerang, kepiting, dan ikan dengan akurasi tinggi, tanpa kesalahan klasifikasi. Hasil ini menunjukkan bahwa model tidak hanya efektif dalam pengklasifikasian gambar dalam kondisi pelatihan, tetapi juga mampu mempertahankan performanya setelah diintegrasikan ke dalam aplikasi *mobile*. Keberhasilan ini membuktikan bahwa model *EfficientNetV2-S* memiliki validitas yang baik dalam aplikasi dunia nyata, memberikan pengalaman pengguna yang efektif dan akurat dalam melakukan klasifikasi makanan berdasarkan gambar.

V. SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil penelitian yang telah dilaksanakan, penelitian ini berhasil mencapai tujuan yang telah ditetapkan dalam mengembangkan model *deep learning* untuk klasifikasi gambar *seafood* penyebab alergi. Penelitian ini menunjukkan efektivitas penggunaan model *MobileNetV2* dan *EfficientNetV2-S* untuk aplikasi *mobile*, serta memberikan *insight* mengenai *hyperparameter tuning* yang dapat mengoptimalkan kinerja model.

Pengembangan Model Deep learning untuk Klasifikasi Makanan Laut Penyebab Alergi: Penelitian ini berhasil mengembangkan model *deep learning* berbasis *MobileNetV2* dan *EfficientNetV2-S* untuk mengidentifikasi makanan laut penyebab alergi seperti udang, kepiting, kerang, dan ikan. Kedua model telah dievaluasi menggunakan metrik akurasi, *precision*, *recall*, dan *F1-Score*, serta berhasil diintegrasikan ke dalam aplikasi *mobile* menggunakan format *TensorFlow Lite*.

Hasil Hyperparameter tuning pada Model EfficientNetV2-S: Hyperparameter tuning secara manual menunjukkan bahwa kombinasi terbaik dicapai dengan pengaturan *dropout* 0.5, L1 dan L2 *regularization* sebesar 0.01, serta tanpa *unfreeze* pada layer pretrained dan *batch size* 32. Pengaturan ini diterapkan pada

EfficientNetV2-S dan berhasil menurunkan *loss* secara stabil pada data pelatihan dan validasi, serta menghindari *overfitting* dan *underfitting*.

Perbandingan Performa Model MobileNetV2 dan EfficientNetV2-S: Hasil evaluasi menunjukkan bahwa *EfficientNetV2-S* memiliki performa yang lebih baik dibandingkan *MobileNetV2*, dengan *accuracy* mencapai 97,5%, *precision* 97,59%, *recall* 97,5%, dan *F1-Score* 97,5%. Model ini juga menunjukkan kestabilan yang lebih baik dalam mencegah *overfitting* dan *underfitting*, menjadikannya pilihan yang lebih stabil dan efektif untuk aplikasi klasifikasi gambar. Hal ini menjadikan *EfficientNetV2-S* sebagai model yang lebih direkomendasikan untuk digunakan dalam implementasi aplikasi *mobile*.

Implementasi Model dalam Aplikasi Mobile menggunakan TensorFlow Lite: Model yang dikembangkan berhasil diterapkan ke dalam aplikasi *mobile* dalam format *TensorFlow Lite*, dengan proses konversi dan integrasi yang berjalan lancar. Hasil pengujian awal menunjukkan bahwa gambar yang diuji melalui aplikasi *mobile* dapat diklasifikasikan dengan benar, menandakan bahwa model dapat berjalan dengan baik di lingkungan *mobile* dan berfungsi sesuai dengan tujuan implementasi awal.

B. Hasil Evaluasi Model

Berdasarkan temuan dan hasil yang diperoleh dalam penelitian ini, terdapat beberapa rekomendasi untuk penelitian dan pengembangan selanjutnya. Saran-saran ini bertujuan untuk meningkatkan efisiensi, performa, dan kemampuan generalisasi model dalam berbagai kondisi, serta meningkatkan kualitas *dataset* dan proses pelatihan untuk mencapai hasil yang lebih baik. Berikut adalah beberapa saran yang dapat dipertimbangkan untuk penelitian mendatang:

1) Penggunaan Perangkat dengan Daya Komputasi Tinggi: Gunakan perangkat dengan daya komputasi tinggi (computing power), terutama pada saat pelatihan model dengan *dataset* besar atau saat mencoba arsitektur model yang lebih kompleks, agar proses berjalan lebih efisien dan optimal.

2) Eksperimen Hyperparameter tuning yang Lebih Sistematis: Lakukan eksperimen *hyperparameter tuning* yang lebih sistematis dan luas, seperti penggunaan grid search atau random search, agar kombinasi terbaik dapat ditemukan secara objektif dan menyeluruh.

3) Perluasan dan Pembaruan Dataset yang Efektif: Perluas dan perbarui *dataset* menggunakan teknik *Web scraping* yang baik, dengan memperhatikan kualitas dan validasi gambar hasil *scraping* agar *dataset* tetap relevan, bersih, dan konsisten terhadap label.

4) Penggunaan Model Deep learning yang Lebih Baru atau Lebih Ringan: Pertimbangkan penggunaan model *deep learning* yang lebih baru atau lebih ringan, jika di masa mendatang tersedia arsitektur yang menawarkan peningkatan akurasi, efisiensi, atau kecepatan inferensi pada perangkat *mobile*.

5) Pengujian di Berbagai Kondisi Pencahayaan dan Latar Belakan: Diperlukan pengujian lebih luas di berbagai kondisi pencahayaan dan latar belakang agar aplikasi dapat beradaptasi lebih baik terhadap penggunaan nyata di lingkungan beragam.

6) Peningkatan Kualitas Dataset dan Penyaringan Gambar yang Lebih Baik: Diperlukan peningkatan kualitas *dataset* dengan penyaringan gambar yang lebih baik, menggunakan teknik penyaringan otomatis yang memanfaatkan deteksi objek dan pemrosesan berbasis AI untuk menghindari pengunduhan gambar yang tidak relevan, seperti gambar ber*logo*, teks, atau kualitas rendah. Selain itu, algoritma scrolling dan pencarian gambar juga perlu dioptimalkan agar proses *scraping* menjadi lebih efisien.

7) Penerapan Teknik Augmentasi Data Otomatis: Teknik augmentasi data otomatis juga perlu diterapkan untuk memperkaya variasi gambar dalam *dataset* dan mengurangi risiko *overfitting*. Penggunaan model pembelajaran mesin atau *deep learning* untuk klasifikasi gambar otomatis dapat membantu gambar lebih akurat dikelompokkan sesuai kategori yang relevan, meningkatkan kualitas *dataset* secara keseluruhan.

DAFTAR PUSTAKA

[1] Z. Azizah, A. H. Falihah, B. A. B. Santoso, I. Puspitasari and M. N. A. Sahid, "Frekuensi Alergi Makanan Berdasarkan Survei pada Orang Dewasa di Wilayah Yogyakarta dan Jawa," 7 July 2023. [Online]. Available: https://journal.ugm.ac.id/majalahfarmaseutik/article/view/85546/39420.

[2] F. Chollet, *Deep learning with Python Second Edition*, Manning Publications Co., 2021.

[3] M. Elgendy, *Deep learning for Vision Systems*, Manning Publications Co., 2021.

[4] H. Asad, V. R. Shrimali and N. Singh, *The Computer Vision Workshop*, Packt Publishing Ltd., 2020.

[5] J. Brownlee, *Machine learning Mastery With Python*, Machine learning Mastery, 2019.

[6] V. Lakshmanan, M. Görner and R. Gillard, *Practical Machine learning for Computer Vision*, O'Reilly Media, Inc., 2021.

[7] A. Tragoudaras, P. Stoikos, K. Fanaras, A. Tziouvaras, G. Floros, G. Dimitriou, K. Kolomvatsos and G. Stamoulis, "Design Space Exploration of a Sparse MobileNetV2 Using High-Level Synthesis and Sparse Matrix Techniques on FPGAs," *MDPI,* 2022.

[8] M. Tan and Q. V. Le, "*EfficientNetV2: Smaller Models and Faster Training*," 23 June 2021. [Online]. Available: https://arxiv.org/abs/2104.00298. [Accessed 19 March 2025].

[9] T. Lee, Y. Na, B. G. Kim, S. Lee and Y. Choi, "Identification of Individual Hanwoo Cattle by Muzzle Pattern Images through *Deep learning* ," *MDPI,* 2023.

[10] DeepLearning.AI, "Convolutional Neural Networks in TensorFlow," 27 October 2024. [Online]. Available: https://www.coursera.org/programs/bangkit-2024-machine-learning-ftkc9/learn/convolutional-neural-networks-*TensorFlow*..

[11] DeepLearning.AI, "*Device-based Models with TensorFlow Lite*," 17 *No*vember 2024. [Online]. Available: https://www.coursera.org/programs/bangkit-2024-machine-learning-ftkc9/learn/device-based-models-*TensorFlow*.

[12] *TensorFlow*, "*TensorFlow API Documentation*," 8 *No*vember 2024. [Online]. Available: https://www.*TensorFlow*.org/.

[13] M. Moocarme, A. So and A. Maddalone, *The TensorFlow Workshop*, Packt Publishing Ltd., 2021.

[14] R. Mitchell, *Web scraping with Python*, Sebastopol: O'Reilly Media, 2018.