

MODEL CORALNET, INCEPTIONV3, DAN MOBILENETV2 UNTUK KLASIFIKASI KONDISI TERUMBU KARANG

Ulfa Hidayati^{1*}, Felix Windriyareksa Hardyan², Faizah Rizki Auliawati³, Ali Akbar Rafsanjani⁴
 Fanka Arie Reza⁵, Mario Mora Siregar⁶

^{1,2,3,6}Prodi Informatika Universitas Gunadarma; Jl. Margonda Raya No. 100, Depok; 7863819

⁴Prodi Sistem Komputer Universitas Gunadarma; Jl. Margonda Raya No. 100, Depok; 7863819

⁵Prodi Sistem Informasi Universitas Gunadarma; Jl. Margonda Raya No. 100, Depok; 7863819

Keywords:

Deep Learning;
 Convolutional Neural
 Network (CNN);
 CoralNet; InceptionV3;
 MobileNetV2.

Correspondent Email:

ulfahidayati21@gmail.com

Abstrak. Indonesia memiliki keanekaragaman hayati laut tertinggi di dunia dengan sekitar 569 spesies dan 83 genus karang batu (*stony coral*) yang mencakup 69% spesies dan 76% genus dari total karang batu global. Namun, terumbu karang sangat rentan terhadap perubahan iklim. Faktor lingkungan dapat merusak fotosistem II, menyebabkan karang kehilangan alga simbiotiknya dan tampak memutih. Penerapan model *Support Vector Machine* (SVM) dan *Bag of Features* (BoF) dilakukan sebagai solusi AI untuk mengklasifikasikan kondisi terumbu karang. Namun, pendekatan ini bergantung pada ekstraksi fitur manual dan memerlukan alur kerja yang lebih kompleks dibandingkan metode *end-to-end deep learning* seperti CoralNet, InceptionV3, dan MobileNetV2. Penelitian ini bertujuan untuk analisis perbandingan performa dari ketiga model tersebut dalam mengklasifikasikan kondisi terumbu dalam kelas *healthy*, *bleached*, dan *dead*. MobileNetV2 memberikan hasil terbaik dengan akurasi pelatihan 97%, validasi 88%, dan pengujian 89% untuk klasifikasi terumbu karang.



JITET is licensed under
 a Creative Commons
 Attribution-NonCommercial
 4.0 International License

Abstract. Indonesia hosts the highest marine biodiversity in the world, with approximately 569 species and 83 genus of stony corals, accounting for 69% of global coral species and 76% of genera. However, coral reefs are highly vulnerable to climate change. Environmental factors can damage photosystem II, causing corals to lose their symbiotic algae and resulting in coral bleaching. The application of Support Vector Machine (SVM) and Bag of Features (BoF) models has been explored as AI-based solutions for classifying coral reef conditions. However, these approaches rely on manual feature extraction and involve a more complex workflow compared to end-to-end deep learning methods such as CoralNet, InceptionV3, and MobileNetV2. This study aims to analyze and compare the performance of these three models in classifying coral conditions into three categories: healthy, bleached, and dead. Among the models, MobileNetV2 achieved the best performance, with training accuracy of 97%, validation accuracy of 88%, and testing accuracy of 89% for coral reef classification.

1. PENDAHULUAN

Sebagai bagian dari wilayah segitiga terumbu karang (*coral triangle*), Indonesia

memiliki keanekaragaman hayati laut tertinggi di dunia. Tercatat ada sekitar 569 spesies dan 83 genus karang batu (*stony coral*), yang masing-

masing mewakili sekitar 69% spesies dan 76% genus dari total karang batu di seluruh dunia. Dalam upaya melestarikan keanekaragaman hayati dan habitat laut, pemerintah telah menetapkan 22,68 juta hektar kawasan konservasi laut hingga pertengahan tahun 2019, yang setara dengan sekitar 6% dari total wilayah laut Indonesia [1].

Terumbu karang sangat sensitif terhadap perubahan iklim, terutama karena hubungan simbiosis antara karang pembentuk terumbu (*stony reef-building corals*) dan *zooxanthellae* (*Symbiodinium spp.*). Fotosintesis alga ini menyediakan hingga 90% energi bagi karang. Stres lingkungan, seperti suhu laut yang tinggi dan penyinaran berlebih, dapat merusak fotosistem II (*photosystem II*). Kerusakan ini menyebabkan karang mengeluarkan alganya, sehingga tampak ‘memutih’ [2].

Pemutihan dapat berkisar dari ringan hingga parah. Karang yang mengalami pemutihan ringan bisa pulih, tetapi pemutihan parah dapat menyebabkan kematian massal. Dalam jangka panjang, pemutihan mengurangi rugositas karang, tutupan karang, keanekaragaman hayati, dan kualitas habitat bagi spesies lain. Perubahan iklim diperkirakan akan meningkatkan frekuensi dan intensitas pemutihan di masa depan [2].

Demi menghadapi tantangan yang ditimbulkan oleh kerusakan dan pemutihan terumbu karang, penerapan teknologi kecerdasan buatan atau *Artificial Intelligence* (AI) menawarkan solusi yang efektif untuk melakukan klasifikasi kondisi terumbu karang secara otomatis. Penerapan *Support Vector Machine* (SVM) dan *Bag of Features* (BoF) telah terbukti efektif dalam klasifikasi kondisi terumbu karang. Dalam pendekatan ini, gambar terumbu karang dipecah menjadi *patch* kecil, dan fitur-fitur visual diekstraksi menggunakan deskriptor manual seperti *Local Binary Patterns* (LBP) dan *Gray Level Co-occurrence Matrix* (GLCM), serta model *Convolutional Neural Networks* (CNN) seperti CoralNet [3].

Fitur-fitur ini kemudian digabungkan menjadi vektor fitur tunggal yang disebut BoF dan dikelompokkan dengan algoritma *K-Means Clustering* untuk membentuk *Visual Vocabulary* (VV). Vektor VV ini digunakan sebagai input oleh SVM untuk mengklasifikasikan kondisi terumbu karang yang sehat, memutih, atau mati. Pendekatan

BoF dan SVM efektif dalam menangkap pola tekstur, warna, dan spasial dari citra terumbu karang. Namun, meskipun memiliki kinerja yang baik, pendekatan ini bergantung pada ekstraksi fitur manual dan memerlukan alur kerja yang lebih kompleks dibandingkan dengan metode *end-to-end deep learning*, yang dapat mengotomatisasi seluruh proses [3].

Pendekatan *end-to-end deep learning* dan *transfer learning* memungkinkan model *pre-trained* memanfaatkan pengetahuan yang diperoleh dari pelatihan awal (*pre-training*) pada dataset besar seperti ImageNet. Teknik ini mempercepat proses pelatihan dan meningkatkan akurasi, terutama pada dataset terbatas. Melalui *fine-tuning*, lapisan model *pre-trained* disesuaikan dengan data spesifik, seperti kondisi terumbu karang, untuk memastikan hasil yang optimal [4]. Berbeda dengan metode tradisional seperti BoF dan SVM yang membutuhkan proses ekstraksi fitur manual dan klasifikasi secara terpisah, pendekatan *end-to-end deep learning* ini dirancang untuk memproses data secara langsung, mengurangi kompleksitas alur kerja sekaligus meningkatkan kinerja.

CoralNet, InceptionV3, dan MobileNetV2 merupakan model berbasis *Convolutional Neural Network* (CNN) yang mendukung pendekatan *end-to-end deep learning*. CoralNet dirancang khusus untuk klasifikasi kondisi terumbu karang [3]. InceptionV3 unggul dalam menangani pola visual kompleks dengan efisiensi komputasi yang tinggi [5]. MobileNetV2 menawarkan performa cepat dan efisien, terutama pada perangkat keras dengan sumber daya terbatas [6]. Keunggulan tersebut menjadikan ketiga model ini layak untuk diterapkan dalam klasifikasi kondisi terumbu karang secara akurat dan efisien.

Penelitian ini bertujuan untuk: (1) Melatih, memvalidasi, dan menguji model CNN CoralNet, InceptionV3, dan MobileNetV2 dalam mengklasifikasikan kondisi terumbu karang. (2) Melakukan *comparative analysis* pada performa dari ketiga model CNN dalam mengklasifikasikan kondisi terumbu karang.

2. TINJAUAN PUSTAKA

2.1. Deep Learning

Deep Learning adalah salah satu pendekatan dalam *Artificial Intelligence* (AI) yang memungkinkan komputer berkembang melalui

pengalaman dan data. Model khas *deep learning* adalah *feedforward deep network* atau *multilayer perceptron* (MLP), yaitu fungsi matematis yang memetakan data masukan ke keluaran dengan menggabungkan sejumlah fungsi sederhana. Setiap penerapan fungsi ini menghasilkan representasi baru dari data masukan. Pendekatan ini memberikan dua perspektif, yaitu membantu mempelajari representasi yang sesuai untuk data dan memungkinkan komputer mempelajari program *multi-layer*. Setiap lapisan representasi dianggap sebagai kondisi memori komputer setelah menjalankan serangkaian instruksi secara paralel. Jaringan yang lebih dalam mampu menjalankan instruksi secara berurutan, di mana hasil dari langkah sebelumnya digunakan pada langkah berikutnya [7].

2.2. Convolutional Neural Network (CNN)

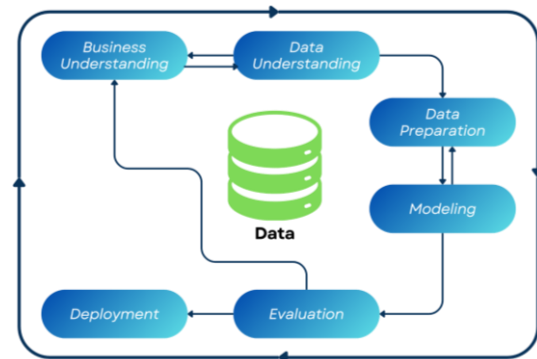
Convolutional Neural Network (CNN) merupakan salah satu model pada jaringan *deep learning* yang dirancang untuk mendeteksi dan mengklasifikasikan fitur dalam gambar, khususnya dalam aplikasi *computer vision*. Struktur dan arsitektur CNN terinspirasi oleh cara kerja korteks visual pada otak manusia, yang menyerupai koneksi antar neuron. Secara umum, CNN terdiri dari enam komponen utama, yaitu input *image*, *convolution layer*, *pooling layer*, *activation function*, *fully connected layer*, dan *output layer* [8].

2.3. CRISP-DM

CRISP-DM (*Cross Industry Standard Process for Data Mining*) adalah suatu pendekatan yang menyusun standar pemrosesan *data mining* dengan tujuan agar data dapat diproses melalui langkah-langkah yang terorganisir, jelas, dan efisien.

Pendekatan ini terbagi dalam enam tahap, yakni:

1. *Business Understanding*,
2. *Data Understanding*,
3. *Data Preparation*,
4. *Modeling*,
5. *Evaluation*, dan
6. *Deployment* [9].



Gambar 1. Alur Proses Pengembangan Proyek dengan Metode CRISP-DM

2.4. Confusion Matrix

Salah satu metrik atau alat yang digunakan untuk mengevaluasi performa klasifikasi adalah *Confusion Matrix*. Matriks ini menyajikan tabel yang membandingkan hasil prediksi model dengan hasil yang sebenarnya dari data dengan mengukur empat metrik sebagai berikut:

- *True Positive* (TP): Kondisi saat model memprediksi hasil sebagai nilai positif, yaitu hasil prediksi sesuai dengan kategori atau kelas yang sebenarnya.
- *False Positive* (FP): Kondisi ketika model memprediksi hasil sebagai nilai positif, tetapi nilai sebenarnya negatif. *False Positive* menunjukkan bahwa model salah mengidentifikasi data yang seharusnya bukan bagian dari kategori yang dicari sebagai bagian dari kategori tersebut.
- *False Negative* (FN): Kondisi ketika model memprediksi hasil sebagai nilai negatif, padahal nilai sebenarnya positif. Artinya, model gagal mengenali data yang sebenarnya masuk dalam kategori yang dicari.
- *True Negative* (TN): Kondisi saat model memprediksi hasil sebagai nilai negatif, dan nilai sebenarnya juga negatif. Nilai negatif menunjukkan bahwa hasil tidak sesuai dengan kategori yang sedang dianalisis atau dicari.

Rumus menghitung metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score* digunakan untuk mengukur performa model dalam melakukan prediksi dengan benar dalam konteks dataset yang diberikan [10].

3. METODE PENELITIAN

3.1. Business Understanding

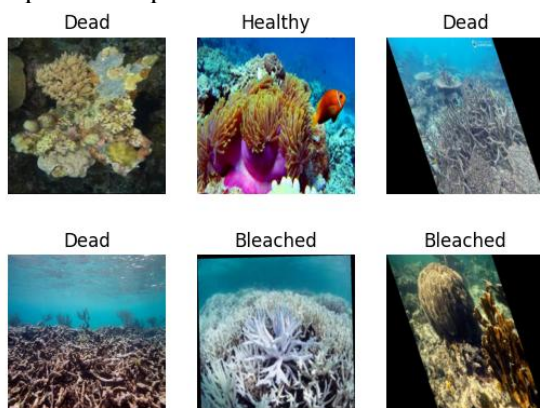
Tahap pertama pada CRISP-DM disebut dengan *business understanding*. Fokus pada tahap *business understanding* adalah memahami tujuan dan persyaratan yang dibutuhkan secara menyeluruh.

Tujuan dari penelitian ini adalah untuk menganalisis dan membandingkan performa dari tiga algoritma *deep learning* untuk melakukan klasifikasi gambar kondisi terumbu karang. Analisis akan dilakukan untuk mengetahui metode dengan performa terbaik dalam pengklasifikasian. Kondisi yang akan diklasifikasi antara lain:

- Terumbu karang sehat (*healthy coral*),
- Terumbu karang memutih (*bleached coral*),
- Terumbu karang mati (*dead coral*).

3.2. Data Understanding

Dalam penelitian ini, *dataset* yang digunakan adalah *dataset* 'BHD Corals' yang dapat diunduh secara terbuka dan gratis pada situs Kaggle. *Dataset* ini memiliki 3 kelas yaitu, *healthy* (terumbu karang sehat), *bleached* (terumbu karang memutih), dan *dead* (terumbu karang mati). Jumlah data pada *dataset* 'BHD Corals' terdiri dari 1250 gambar dengan ukuran 227×227 piksel, warna RGB, dan format PNG. Gambar pada *dataset* ini telah melalui proses augmentasi menggunakan rotasi gambar dan *shearing* [11]. Sampel gambar yang digunakan dapat dilihat pada Gambar 2.



Gambar 2. Sampel acak gambar dari *dataset* 'BHD Corals'

3.3. Data Preparation

Tahap *data preparation* merupakan langkah krusial dalam proses analisis data. Kualitas data

memiliki peran besar dalam menentukan keberhasilan pengembangan model. Sehingga *Dataset* 'BHD Corals' perlu diolah terlebih dahulu sebelum dapat digunakan pada tahapan *modeling*. Terdapat dua tahapan pengolahan data yaitu *data splitting* (pembagian data) dan *data augmentation* (augmentasi data).

3.3.1. Data Splitting

Pada tahapan ini, *dataset* dibagi menjadi data latih, data validasi, dan data uji [12]. Pembagian ini dilakukan dengan rasio 80:10:10, yaitu 80% data latih, 10% data validasi, dan 10% data uji.

3.3.2. Data Augmentation

Data yang sudah dibagi kemudian akan dilakukan proses augmentasi untuk memperbanyak jumlah data dan mencegah terjadinya *overfitting* pada model [4]. Augmentasi yang digunakan adalah sebagai berikut:

- Rescale* untuk normalisasi data,
- Width shift* untuk menggeser gambar searah dengan lebar gambar,
- Height shift* untuk menggeser gambar searah dengan tinggi gambar,
- Zoom* untuk memperbesar gambar,
- Horizontal flip* untuk membalikan gambar secara horizontal.

3.4. Modeling

Modeling pada penelitian ini dilakukan dengan menggunakan bahasa pemrograman Python, library TensorFlow, dan Keras. Tiga model *deep learning* yang digunakan adalah CoralNet, InceptionV3, dan MobileNetV2 yang merupakan model *Convolutional Neural Network* (CNN).

Model CNN memiliki tiga komponen penting yaitu lapisan konvolusi untuk ekstraksi *feature map*, lapisan *max-pooling* untuk mengurangi ukuran *feature map*, dan lapisan *fully-connected* untuk melakukan tugas klasifikasi.

3.4.1. CoralNet

CoralNet adalah model CNN yang dirancang untuk klasifikasi kondisi terumbu karang dengan menggunakan kombinasi *Support Vector Machine* (SVM) dan *Bag of Features* (BoF). Namun, dalam penelitian ini, CoralNet akan disesuaikan untuk penerapan *end-to-end deep learning* guna menyederhanakan alur kerja dan

meningkatkan efisiensi. Model ini memiliki 13 lapisan yang terdiri dari, tiga lapisan *convolutional* (Conv2D), dua lapisan *max-pooling* (MaxPooling2D), dua lapisan *dense*, satu lapisan *flatten*, dan tiga lapisan aktivasi. Ukuran lapisan input ditetapkan sebesar $227 \times 227 \times 3$. Dua lapisan aktivasi menggunakan fungsi aktivasi *rectified linear unit* (ReLU), sementara lapisan aktivasi terakhir menggunakan fungsi aktivasi *softmax* yang digunakan untuk klasifikasi. Fitur diekstraksi menggunakan lapisan *convolutional* dan *max-pooling*. Klasifikasi dilakukan dengan lapisan *fully-connected* [3]. Model CoralNet pada penelitian ini disederhanakan seperti pada Tabel 1.

Tabel 1. Arsitektur CoralNet

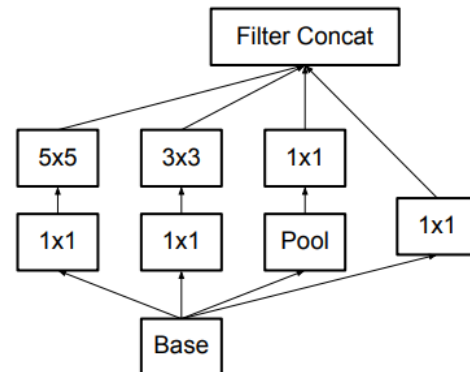
Layer	Output Shape	Param#
Conv2D	(None, 225, 225, 32)	896
MaxPooling 2D	(None, 112, 112, 32)	0
Conv2D	(None, 110, 110, 64)	18,496
MaxPooling 2D	(None, 55, 55, 64)	0
Conv2D	(None, 53, 53, 64)	36,928
Flatten	(None, 179776)	0
Dense	(None, 128)	23,011,456
Dense	(None, 3)	387

Arsitektur CoralNet pada Tabel 1 memiliki 23,068,163 total parameter.

3.4.2. InceptionV3

InceptionV3 adalah model *pre-trained* yang telah dilatih menggunakan *dataset* ImageNet 2012. Salah satu keunggulan model InceptionV3 adalah adanya penerapan *inception module* sebagai komponen utama pada arsitektur model. *Inception module* memungkinkan operasi konvolusi paralel dengan ukuran filter yang berbeda, biasanya termasuk konvolusi 1×1 , 3×3 , dan 5×5 , serta lapisan *max pooling*. Jalur paralel ini kemudian dikombinasikan untuk membentuk

output modul. Hal ini memungkinkan model menangkap fitur lokal dan global secara bersamaan, meningkatkan kemampuannya dalam mengenali pola yang kompleks [5].



Gambar 3. Inception Module [5]

Pada penelitian ini, model InceptionV3 digunakan sebagai *feature extractor* untuk memanfaatkan kemampuan ekstraksi fitur yang kuat dari InceptionV3. Lapisan *fully-connected* baru ditambahkan untuk melakukan klasifikasi. Lapisan *fully-connected* baru ini terdiri dari satu lapisan *input*, satu lapisan *flatten*, dua lapisan *dense*, dan satu lapisan *dropout*. Model InceptionV3 dan lapisan *fully-connected* yang digunakan dapat dilihat pada Tabel 2.

Tabel 2. Gabungan Arsitektur InceptionV3 dan Lapisan *Fully-Connected* Baru

Layer	Output Shape	Param#
InputLayer	(None, 227, 227, 3)	0
inception_v3	(None, 5, 5, 2048)	21,802,784
Flatten	(None, 51200)	0
Dense	(None, 512)	26,214,912
Dropout	(None, 512)	0
Dense	(None, 3)	1,539

Arsitektur InceptionV3 pada Tabel 2 memiliki 48,019,235 total parameter, dengan 26,216,451 *trainable parameter* dan 21,802,784 *non-trainable parameter*.

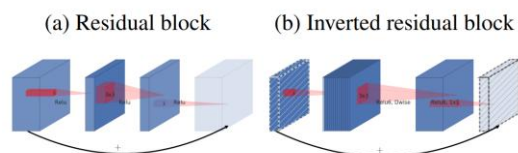
3.4.3. MobileNetV2

MobileNetV2 adalah model *pre-trained* yang dikembangkan oleh Google pada tahun

2019 dan merupakan pengembangan dari model versi sebelumnya yaitu MobileNetV1. Pada konvolusi standar, proses *filtering* dan penggabungan input dilakukan dalam satu langkah untuk menghasilkan *output* baru. MobileNetV2 menerapkan *depthwise separable convolution* yaitu bentuk konvolusi terfaktorisasi yang memisahkan konvolusi standar menjadi *depthwise convolution* dan konvolusi 1×1 yang disebut *pointwise convolution*. Pada MobileNet, *depthwise convolution* menerapkan satu filter pada setiap *channel* input. Kemudian, *pointwise convolution* menerapkan konvolusi 1×1 untuk menggabungkan keluaran dari *depthwise convolution*. Faktorisasi ini secara signifikan mengurangi beban komputasi dan ukuran model [13]. selain *depthwise separable convolution*, MobileNetV2 juga menggunakan *linear bottleneck* dan *Inverted residual block* [6].

Linear bottleneck berfungsi untuk menjaga informasi penting. Teknik ini memastikan bahwa transformasi non-linear, seperti ReLU, tidak menghapus terlalu banyak informasi dari data yang diproses. Caranya adalah dengan menggunakan lapisan linear yang mempertahankan struktur data pada ruang berdimensi rendah, sehingga *neural network* tidak kehilangan detail penting [6].

Inverted residual block bekerja dengan cara memulai dari lapisan berdimensi rendah (*bottleneck*), diikuti oleh ekspansi ke dimensi yang lebih tinggi melalui konvolusi *depthwise* dan *pointwise*, sebelum kembali ke dimensi semula. Kombinasi *linear bottleneck* dan *inverted residual* memungkinkan pengurangan kompleksitas komputasi secara signifikan tanpa mengorbankan akurasi [6].



Gambar 4. *Inverted Residual Block* [6]

Sama seperti model InceptionV3, pada penelitian ini, model MobileNetV2 digunakan sebagai *feature extractor*. Lapisan *fully-connected* baru ditambahkan untuk melakukan klasifikasi, yang terdiri dari satu

lapisan *input*, satu lapisan *flatten*, dua lapisan *dense*, dan satu lapisan *dropout*. Model MobileNetV2 dan lapisan *fully-connected* layer yang digunakan dapat dilihat pada Tabel 3.

Tabel 3. Gabungan Arsitektur MobileNetV2 dan Lapisan *Fully-Connected* Baru

Layer	Output Shape	Param#
InputLayer	(None, 227, 227, 3)	0
mobilenetv2_1.00_224	(None, 8, 8, 1280)	2,257,984
Flatten	(None, 81920)	0
Dense	(None, 512)	41,943,552
Dropout	(None, 512)	0
Dense	(None, 3)	1,539

Arsitektur MobileNetV2 pada Tabel 3 memiliki 44,203,075 *total parameter*, dengan sebanyak 41,945,091 *trainable parameter* dan 2,257,984 *non-trainable parameter*.

3.4.4. *Hyperparameter Pelatihan*

Ketiga model dilatih menggunakan beberapa *hyperparameter* penting yang dirancang untuk mengoptimalkan performa klasifikasi model. Pelatihan ketiga model menggunakan *hyperparameter* yang sama untuk memastikan perbandingan performa dapat diukur secara adil. Tabel 4 merupakan *hyperparameter* pelatihan yang digunakan dalam penelitian ini.

Tabel 4. *Hyperparameter Pelatihan Model*

Parameter	Value
<i>Loss Function</i>	<i>Categorical Cross-Entropy</i>
<i>Optimizer</i>	Adam dengan <i>learning rate</i> 0.0001
<i>Epoch</i>	20
<i>Batch Size</i>	64

Loss function merupakan parameter yang berfungsi untuk mengevaluasi kualitas prediksi model. *Categorical cross-entropy* digunakan saat data memiliki jumlah kelas lebih dari dua dan data bersifat kategorik [14].

Optimizer function adalah algoritma matematis yang menggunakan kalkulus (turunan dan turunan parsial) untuk menentukan seberapa besar perubahan yang diperlukan pada *loss function* agar mencapai nilai minimum. *Adaptive Moment Estimation (Adam)* adalah teknik optimasi yang menghitung *adaptive learning rate* untuk setiap parameter, memadukan momentum dan varians dari *gradient loss* untuk memperbarui bobot parameter, sehingga meningkatkan stabilitas dan efisiensi pelatihan [14].

Epoch adalah *hyperparameter* yang menentukan berapa kali algoritma pembelajaran akan memproses seluruh *dataset* pelatihan. Satu *epoch* berarti setiap sampel dalam *dataset* pelatihan telah digunakan untuk memperbarui parameter internal model. Satu epoch terdiri dari satu atau lebih *batches*, di mana data dibagi menjadi subset untuk efisiensi komputasi [14].

Sedangkan *Batch size* merupakan *hyperparameter* yang menentukan jumlah sampel yang diproses sebelum pembaruan model dilakukan. Nilai *batch size* harus lebih besar dari atau sama dengan satu, dan lebih kecil dari atau sama dengan jumlah sampel dalam *dataset* pelatihan. Ukuran *batch* yang umum digunakan antara lain 32, 64, dan 128 sampel [15].

4. HASIL DAN PEMBAHASAN

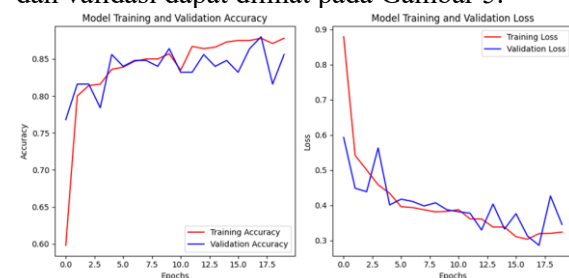
Dataset yang telah melalui proses di tahap sebelumnya akan dilanjutkan pada pengukuran dan analisis perbandingan performa dari ketiga model menggunakan metrik evaluasi dan visualisasi.

4.1. Model CoralNet

Hasil pelatihan dan validasi model selama 20 epoch menunjukkan peningkatan *accuracy* dan penurunan *loss* yang konsisten, meskipun terdapat ketidakkonsistenan di beberapa titik. Pada epoch pertama, *training accuracy* dimulai dengan 59,8%, dan *validation accuracy* mencapai 76,8%. Setelah 20 epoch, *training accuracy* meningkat menjadi 87,8%, sementara *validation accuracy* mencapai 85,6%. Peningkatan ini menunjukkan bahwa model mampu mempertahankan kemampuan generalisasi terhadap data validasi sambil belajar pola dari data pelatihan.

Sebaliknya, *training loss* turun secara signifikan dari 0,8787 pada epoch pertama menjadi 0,3236 pada epoch terakhir, menunjukkan bahwa model berhasil mengurangi kesalahan pada data pelatihan. *Validation loss* juga menunjukkan penurunan yang signifikan, dari 0,5925 pada epoch awal menjadi 0,3454 pada epoch terakhir. Namun terdapat ketidakkonsistenan pada beberapa epoch, seperti pada epoch ke-19 *validation loss* meningkat menjadi 0,4263. Hal ini dapat menunjukkan kemungkinan *overfitting*, terutama jika perbedaan antara *training loss* dan *validation loss* terus meningkat.

Secara keseluruhan, model menunjukkan performa yang baik, dengan peningkatan yang konsisten pada *accuracy* dan penurunan pada *loss*. Meski ada ketidakkonsistenan dan indikasi *overfitting* pada beberapa titik, model mampu menunjukkan kemampuan generalisasi yang cukup baik pada data validasi. Hasil pelatihan dan validasi dapat dilihat pada Gambar 5.

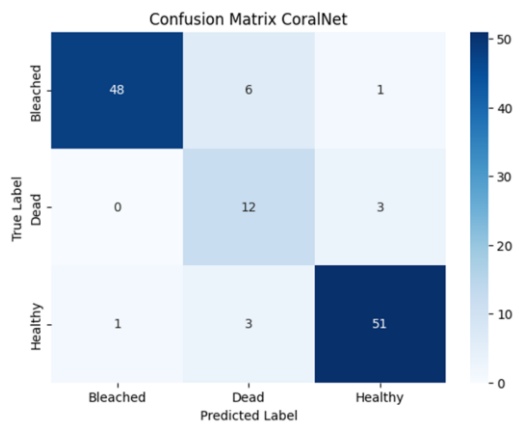


Gambar 5. *Training and Validation* pada Model CoralNet.

Evaluasi model pada data uji menunjukkan akurasi 88,8% dan *loss* 0.3501 pada data uji. Hal ini menunjukkan bahwa data uji memiliki performa yang baik untuk mengklasifikasikan gambar. Meskipun akurasi tinggi, terdapat fluktuasi dalam *loss* yang menandakan ruang untuk peningkatan lebih lanjut.

Berdasarkan Gambar 6, hasil *confusion matrix* menunjukkan bahwa model berhasil mengidentifikasi kategori *Bleached* dan *Healthy*, dengan *True Positives (TP)* masing-masing 48 dan 51. Akan tetapi, model mengalami kesulitan dalam mengklasifikasikan kategori *Dead*, dengan 3 *False Negatives (FN)* dan 6 *False Positives (FP)*. Hal ini menunjukkan bahwa model lebih sulit mengidentifikasi gambar *Dead*. Model berhasil dalam mengidentifikasi kategori *Bleached* dan

Healthy, tetapi perlu peningkatan pada kategori *Dead* untuk mengurangi kesalahan klasifikasi.



Gambar 6. *Confusion Matrix CoralNet*.

Hasil uji prediksi model CoralNet menunjukkan tiga gambar dari dataset uji, lengkap dengan label sebenarnya dan label yang diprediksi oleh model, yang semuanya teridentifikasi sebagai “*Bleached*”. Model berhasil mengenali kondisi terumbu karang yang mengalami *bleached* dengan akurasi tinggi, yakni 99,80%, 94,24%, dan 91,87%. Hal ini membuktikan kepercayaan model yang tinggi terhadap prediksinya, menunjukkan kemampuan CoralNet yang konsisten dan akurat dalam mengklasifikasikan kondisi terumbu karang, serta efektivitasnya dalam mendeteksi data baru. Uji hasil prediksi model CoralNet dapat dilihat pada Gambar 7.



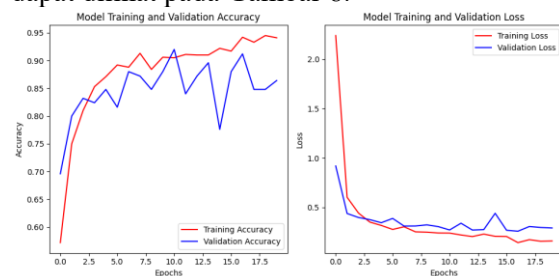
Gambar 7. Uji Hasil Prediksi Model *CoralNet*.

4.2. Model InceptionV3

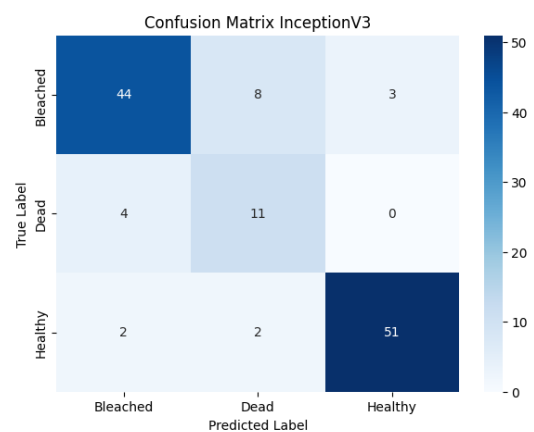
Hasil pelatihan dan validasi, model menunjukkan peningkatan yang signifikan dalam *training accuracy* dan *validation accuracy* selama 20 epoch. Pada epoch pertama, *training accuracy* berada pada 57,2%, sementara *validation accuracy* mencapai 69,6%. Seiring berjalannya waktu, *training accuracy* meningkat secara konsisten hingga mencapai 94,1% pada epoch terakhir, dengan *validation accuracy* mencapai 86,4%.

Meskipun terdapat fluktuasi pada *validation accuracy*, tren keseluruhan menunjukkan peningkatan kemampuan model dalam mengklasifikasikan data dengan baik.

Sebaliknya, *training loss* menunjukkan penurunan signifikan dari 2,2381 pada epoch pertama menjadi 0,1612 pada epoch terakhir, menunjukkan bahwa model mampu mempelajari pola dari data pelatihan secara efektif. Namun, *validation loss* menunjukkan beberapa fluktuasi, khususnya pada epoch ke-15 hingga ke-18, yang dapat menjadi indikasi potensi *overfitting*. Secara keseluruhan, performa model dinilai baik, dengan *validation accuracy* yang tinggi dan *loss* yang relatif stabil dapat dilihat pada Gambar 8.



Gambar 8. *Training and Validation* pada Model *InceptionV3*.



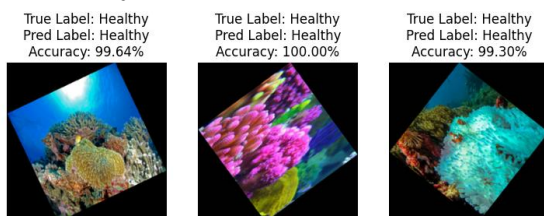
Gambar 9. *Confusion Matrix InceptionV3*.

Evaluasi model pada data uji menunjukkan akurasi 84,8% dan *loss* 0.3577 pada data uji, menunjukkan performa yang cukup baik. Meskipun akurasi relatif tinggi, masih ada ruang untuk perbaikan terutama dalam mengurangi *loss*.

Hasil *confusion matrix* pada Gambar 9 menunjukkan bahwa model berhasil mengidentifikasi kategori *Bleached*, *Dead*, dan *Healthy*. Model berhasil mengidentifikasi 44 gambar *Bleached*, 11 gambar *Dead*, dan 51

gambar *Healthy* dengan benar. Namun, terdapat beberapa kesalahan klasifikasi, seperti 8 gambar *Dead* yang diprediksi sebagai *Bleached* dan 4 gambar *Bleached* yang diprediksi sebagai *Dead*. Meskipun akurasi model cukup baik, terdapat kebingungan antara kategori *Bleached* dan *Dead*, yang menunjukkan perlu adanya perbaikan dalam membedakan kedua kategori tersebut.

Hasil uji prediksi model InceptionV3 menunjukkan tiga gambar dari dataset uji, lengkap dengan label sebenarnya dan label yang diprediksi oleh model, yang semuanya teridentifikasi sebagai "*Healthy*". Pada gambar pertama, model mengidentifikasi sebagai *Healthy* dengan akurasi 99,64%, mencerminkan hasil yang sangat akurat. Gambar kedua mencapai akurasi sempurna sebesar 100%, menunjukkan performa optimal dari model. Gambar ketiga juga diprediksi sebagai *Healthy* dengan akurasi 99,30%, meskipun sedikit lebih rendah dibandingkan gambar pertama. Secara keseluruhan, model mampu mengklasifikasikan semua gambar sebagai *Healthy* dengan tingkat akurasi yang konsisten tinggi dapat dilihat pada Gambar 10.



Gambar 10. Uji Hasil Prediksi Model InceptionV3.

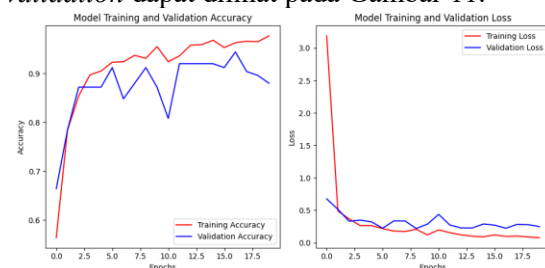
4.3. Model MobileNetV2

Hasil pelatihan selama 20 epoch, model menunjukkan peningkatan yang signifikan dalam *accuracy* dan *loss*, baik pada data pelatihan maupun validasi. Pada epoch pertama, *training accuracy* dimulai dengan 56,4% dan *validation accuracy* mencapai 66,4%. Setelah pelatihan selesai, *training accuracy* meningkat pesat hingga 97,7%, sementara *validation accuracy* mencapai 88,0%.

Training loss juga menurun secara signifikan dari 3,1880 di epoch pertama menjadi 0,0781 di epoch terakhir. *Validation loss*, meskipun mengalami fluktuasi pada beberapa epoch, menunjukkan tren penurunan

keseluruhan, dari 0,6755 di awal pelatihan menjadi 0,2485 di akhir.

Namun, terdapat indikasi *overfitting* karena adanya perbedaan signifikan antara *training accuracy* yang sangat tinggi (97,7%) dan *validation accuracy* yang lebih rendah (88,0%). Fluktuasi pada *validation loss*, khususnya pada epoch ke-7 dan ke-11, semakin menguatkan indikasi ini. Secara keseluruhan, model berhasil belajar dengan baik dari data pelatihan dan menunjukkan kemampuan generalisasi yang memadai pada data validasi. Hasil *training* dan *validation* dapat dilihat pada Gambar 11.



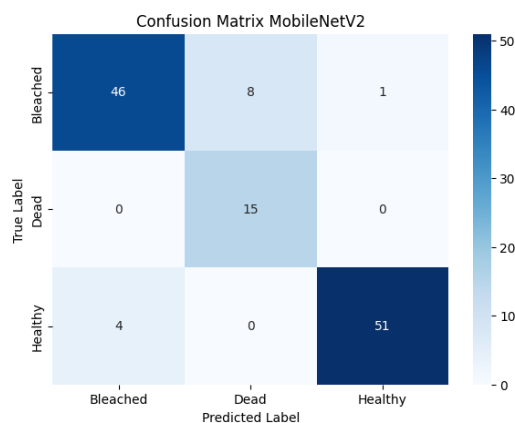
Gambar 11. Training and Validation pada Model MobileNetV2.

Evaluasi pada data uji menunjukkan bahwa model MobileNetV2 mencapai akurasi 89,6% dan loss 0,2043, mencerminkan performa yang sangat baik. Meskipun akurasi tinggi, masih ada beberapa area yang perlu diperbaiki, terutama terkait dengan kesalahan klasifikasi.

Hasil *confusion matrix* menunjukkan bahwa model berhasil mengidentifikasi kategori *Bleached*, *Dead*, dan *Healthy*. Model berhasil mengidentifikasi kategori *Dead* dengan sempurna (15 TP dan 0 FP, FN), sementara untuk *Healthy* dan *Bleached*, model menunjukkan performa yang sangat baik dengan jumlah *True Positives* (TP) yang tinggi dan *False Positives* (FP) yang rendah. Namun, ada beberapa *False Positives* seperti 8 data *Dead* yang salah diklasifikasikan sebagai *Bleached* dan 4 data *Bleached* yang diprediksi sebagai *Healthy*.

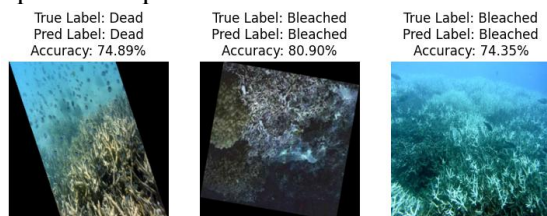
Secara keseluruhan, model MobileNetV2 menunjukkan performa yang sangat baik, dengan akurasi tinggi dan kesalahan minimal. Namun, perlu ada perbaikan dalam membedakan kategori *Bleached* dan *Dead*, serta *Bleached* dan *Healthy*, yang bisa dicapai dengan penambahan data pelatihan yang lebih beragam atau teknik augmentasi. Hasil *confusion matrix* dapat dilihat pada Gambar 13.

Hasil uji prediksi menunjukkan bahwa model berhasil mengklasifikasikan kondisi terumbu karang dengan cukup efektif, meskipun terdapat variasi akurasi antar gambar. Gambar pertama diprediksi sebagai *Dead* dengan akurasi 74,89%, mengindikasikan kemampuan model dalam mengidentifikasi karang yang mati. Pada gambar kedua, model mengklasifikasikan kondisi sebagai *Bleached* dengan akurasi 80,90%, menunjukkan keandalannya dalam mengenali karang yang terbleaching. Gambar ketiga juga diprediksi sebagai *Bleached* dengan akurasi 74,35%, meskipun sedikit lebih rendah, tetap mempertahankan konsistensi dalam klasifikasi.



Gambar 12. *Confusion Matrix MobileNetV2*.

Secara keseluruhan, model mampu membedakan kategori *Dead* dan *Bleached* dengan baik. Dengan akurasi yang cukup tinggi, model ini menunjukkan potensi yang baik dalam mengidentifikasi kondisi terumbu karang. Hasil uji prediksi model MobileNetV2 dapat dilihat pada Gambar 13.



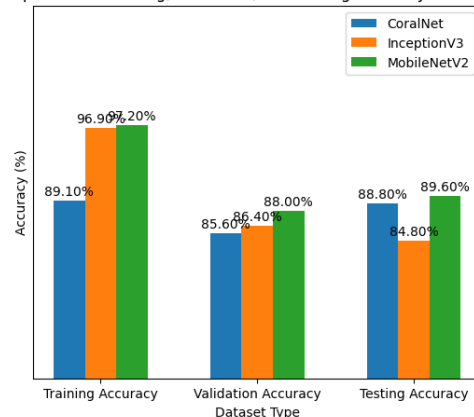
Gambar 13. Uji Hasil Prediksi Model *MobileNetV2*.

4.4. Perbandingan Performa 3 Model

Perbandingan menunjukkan bahwa MobileNetV2 memiliki akurasi pelatihan tertinggi (97,00%), mencerminkan kemampuannya yang luar biasa dalam

mengenali pola pada data pelatihan. Di sisi lain, model ini juga unggul pada akurasi validasi (88,00%), menunjukkan kemampuan generalisasi yang sangat baik. Pada akurasi pengujian, MobileNetV2 kembali mencatat hasil terbaik (89,00%), menjadikannya pilihan yang unggul secara keseluruhan.

Comparison of Training, Validation, and Testing Accuracy for 3 Models



Gambar 14. Perbandingan Performa 3 Model.

Sementara itu, CoralNet menunjukkan akurasi pelatihan sebesar 89,00%, validasi 85,00%, dan pengujian 88,00%. Meskipun memiliki performa yang stabil, hasilnya tidak setara dengan MobileNetV2. Di sisi lain, InceptionV3 unggul pada akurasi pelatihan dengan nilai 96,00%, namun validasi (86,00%) dan pengujian (84,00%) menunjukkan potensi overfitting yang signifikan, mengurangi kemampuan generalisasi model ini.

Secara keseluruhan, MobileNetV2 menawarkan keseimbangan yang lebih baik dalam hal akurasi pelatihan, validasi, dan pengujian, menjadikannya pilihan yang ideal untuk tugas klasifikasi koral. Pemilihan model optimal bergantung pada prioritas pengguna, apakah lebih mengutamakan kemampuan generalisasi atau performa spesifik pada data pelatihan. Perbandingan performa ketiga model dapat dilihat pada Gambar 14.

5. KESIMPULAN

Dari hasil evaluasi terhadap tiga model *computer vision* yang memiliki performa berbeda dalam mengklasifikasikan kondisi terumbu karang untuk kategori *Healthy*, *Bleached* dan *Dead*, berikut adalah kesimpulan yang dapat diambil:

- a. CoralNet memiliki akurasi pelatihan sebesar 89%, validasi 85%, dan pengujian 88%. Namun, model ini kurang efisien untuk digunakan dalam aplikasi yang kompleks.
- b. InceptionV3 menunjukkan performa tinggi pada pelatihan dengan akurasi 96%, tetapi mengalami *overfitting* yang menyebabkan validasi (86%) dan pengujian (84%) menjadi kurang stabil.
- c. MobileNetV2 memberikan hasil terbaik dengan akurasi pelatihan 97%, validasi 88%, dan pengujian 89%. Model ini unggul dalam efisiensi dan konsistensi, menjadikannya pilihan terbaik untuk klasifikasi terumbu karang.

Model MobileNetV2 dapat dikembangkan dengan mengimplementasikan teknologi *Internet of Things* (IoT) untuk klasifikasi kondisi secara *real time* dan identifikasi penyakit pada terumbu karang untuk pelestarian. Penelitian ini diharapkan juga dapat menjadi acuan dalam membantu penelitian terkait di masa yang akan datang.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Hadi, T. A., Abrar M., Giyanto., Prayudha B., Johan O., Budiyanto A., Dzumalek A. R., Alifatri L. O., Sulha S. dan Suharsono, “*The status of Indonesian coral reef 2019*,” Jakarta: research Center for Oceanography, 2020, p. 86
- [2] Heron, S.F., Maynard, J.A., Van Hooijdonk, R. and Eakin, C.M., “Warming trends and bleaching stress of the world’s coral reefs 1985–2012,” *Scientific reports*, 6(1), 2016, p.38402.
- [3] Jamil, S., Rahman, M., Haider, A., “Bag of features (BoF) based deep learning framework for bleached corals detection,” *Big Data and Cognitive Computing*, 5(4), 2021, p.53.
- [4] Aggarwal, C.C., “*Neural networks and deep learning*,” Cham: springer, 2018.
- [5] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., “Rethinking the inception architecture for computer vision,” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826.
- [6] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., “*MobileNetV2: Inverted residuals and linear bottlenecks*,” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510-4520.
- [7] Bengio, Y., Goodfellow, I., & Courville, A., “*Deep learning* (Vol. 1)”. Cambridge, MA, USA: MIT press, 2017, pp. 5-6.
- [8] M. S. Shahab, Achmad Junaidi, and A. N. Sihananto, “Klasifikasi Citra Plankton Dengan Algoritma Hibrida *Convolutional Neural Network* dan *Extreme Learning Machine*,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3S1, Oct. 2024, doi: <https://doi.org/10.23960/jitet.v12i3s1.5219>
- [9] M. F. Harby, Eka Dyar Wahyuni, and N. C. Wibowo, “Rekomendasi Strategi Penjualan Bundling Di Cafe Sz Point Menggunakan Algoritma *Frequent Pattern Growth*,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Aug. 2024, doi: <https://doi.org/10.23960/jitet.v12i3.4867>.
- [10] Zilvi Azus Sriyanti, Dhian Satria Yudha Kartika, and A. Rezha, “Implementasi Model BERT Pada Analisis Sentimen Pengguna Twitter Terhadap Aksi Boikot Produk Israel,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Aug. 2024, doi: <https://doi.org/10.23960/jitet.v12i3.4743>.
- [11] BHD Corals. Available online: <https://www.kaggle.com/sonainjamil/bhd-corals> (diakses pada 13 Oktober 2024).
- [12] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. “MobileNets: efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 126, 2017.
- [13] Montesinos López, O.A., Montesinos López, A., Crossa, J., “*Multivariate statistical machine learning methods for genomic prediction*,” Cham: Springer Nature, 2022.