

**PERBANDINGAN MOBILENETV2 DAN DENSENET121 PADA
KLASIFIKASI CITRA TERUMBU KARANG
MENGUNAKAN *CONVOLUTIONAL NEURAL NETWORK***

LAPORAN TUGAS AKHIR

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



DISUSUN OLEH :

AHMAD AGUNG UTOMO

NIM 32602100138

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
JANUARI 2025**

***COMPARISON OF MOBILENETV2 AND DENSENET121 IN
CORAL REEF IMAGE CLASSIFICATION USING
CONVOLUTIONAL NEURAL NETWORK***

FINAL PROJECT

This report is prepared to fulfill one of the requirements to obtain a Bachelor's Degree (S1) in the Informatics Engineering Study Program, Faculty of Industrial Technology, Sultan Agung Islamic University, Semarang



***MAJORING OF INFORMATICS ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG
JANUARY 2025***

**LEMBAR PENGESAHAN
TUGAS AKHIR**

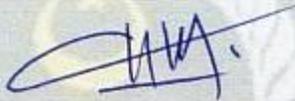
**PERBANDINGAN MOBILENETV2 DAN DENSENET121 PADA
KLASIFIKASI CITRA TERUMBU KARANG MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK**

**AHMAD AGUNG UTOMO
NIM 32602100138**

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 18 Februari 2025

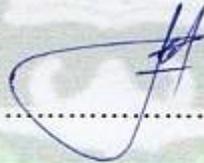
TIM PENGUJI UJIAN SARJANA :

Dedy Kurniadi, ST., M.Kom
NIDN.0622058802
(Ketua Penguji)



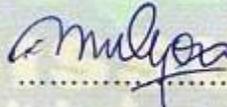
24 - 02 - 2025

**Sam Farisa Chaerul Haviana,
ST., M.Kom**
NIDN.0628028602
(Anggota Penguji)



25 - 02 - 2025

Ir. Sri Mulyono, M.Eng
NIDN.0626066601
(Pembimbing)



24 - 02 - 2025

Semarang, 25 Februari 2025

Mengetahui,

Kaprodi Teknik Informatika
Universitas Islam Sultan Agung



Moch Taufik, S.T., M.IT
NIDN 0622037502



SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Ahmad Agung Utomo

NIM : 32602100138

Judul Tugas Akhir : Perbandingan MobileNetV2 Dan DenseNet121 Pada Klasifikasi Citra Terumbu Karang Menggunakan *Convolutional Neural Network*.

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 25 Februari 2025

Yang Menyatakan,



Ahmad Agung Utomo

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Ahmad Agung Utomo

NIM : 32602100138

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Alamat Asal : Semarang

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Perbandingan MobileNetV2 Dan DenseNet121 Pada Klasifikasi Citra Terumbu Karang Menggunakan *Convolutional Neural Network*

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, 25 Februari 2025



Ahmad Agung Utomo

KATA PENGANTAR

Puji syukur penulis panjatkan kepada ALLAH SWT, yang telah memberikan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Perbandingan MobileNetV2 dan DenseNet121 Pada Klasifikasi Citra Terumbu Karang menggunakan *Convolutional Neural Network*” ini dengan baik. Dengan penuh rasa hormat, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT, atas segala rahmat dan karunia-Nya.
2. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, SH., M.Hum yang mengizinkan penulis menimba ilmu di kampus ini.
3. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T
4. Dosen pembimbing penulis Bapak Ir. Sri Mulyono, M.Eng yang telah memberikan arahan, bimbingan, dan saran yang sangat berarti dalam proses penelitian hingga penulisan laporan ini.
5. Dosen-dosen Program Studi Teknik Informatika, Fakultas Teknologi Industri UNISSULA yang telah membekali penulis dengan ilmu dan pengetahuan selama masa studi.
6. Orang tua penulis, Bapak Sumartono dan Ibu Nurmarsih yang selalu memberikan doa, dukungan, dan motivasi selama proses penyelesaian tugas akhir ini.
7. Teman-teman dan rekan seperjuangan, yang telah memberikan moral, motivasi, serta semangat dalam proses penyelesaian tugas akhir ini.
8. Dan kepada seluruh pihak yang tidak dapat saya sebutkan satu persatu.

Semarang, 19 Januari 2025

Penulis

DAFTAR ISI

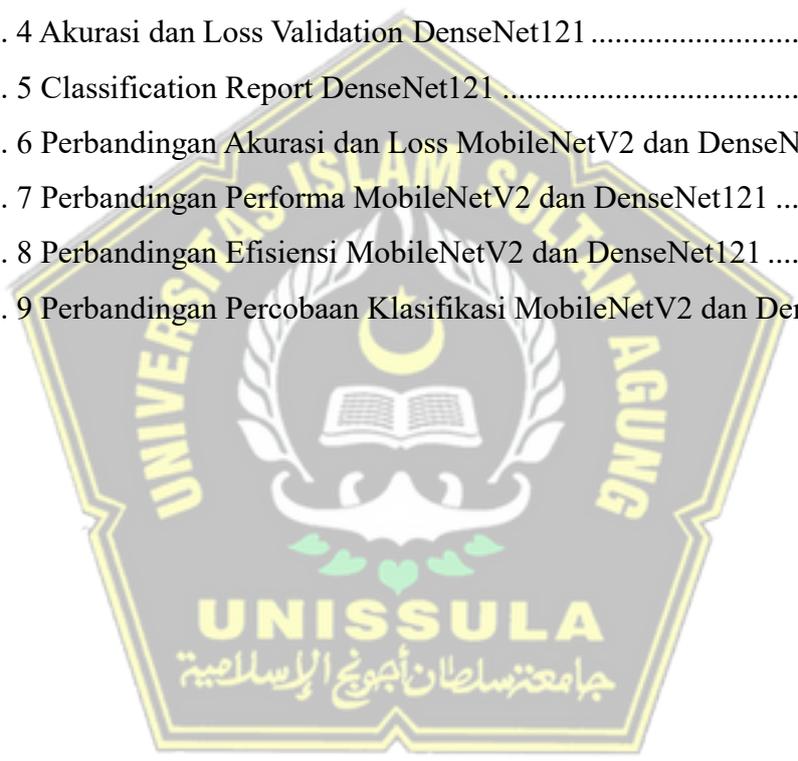
HALAMAN JUDUL	i
LEMBAR PENGESAHAN TUGAS AKHIR	ii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
DAFTAR LAMPIRAN	x
ABSTRAK	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Pembatasan Masalah	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	7
2.1 Tinjauan Pustaka.....	7
2.2 Dasar Teori	8
2.2.1 Pemutihan Terumbu Karang dan Klasifikasinya.....	8
2.2.2 <i>Convolutional Neural Network</i> (CNN).....	10
2.2.3 MobileNetV2.....	11
2.2.4 DenseNet121	12
2.2.5 Perbandingan Arsitektur MobileNetV2 dan DenseNet121	13
BAB III METODE PENELITIAN	14
BAB IV HASIL DAN ANALISIS PENELITIAN.....	21
4.1 HASIL	21
4.1.1 Pengumpulan Data	21
4.1.2 <i>Preprocessing</i>	22

4.1.3	<i>Split Data</i>	22
4.1.4	Pelatihan Model	23
4.1.5	Pengujian Model	24
4.2	Analisa	33
4.2.1	Evaluasi Model	33
4.2.2	Perbandingan MobileNetV2 dan DenseNet121	34
4.2.3	Hasil Implementasi Model	36
BAB V KESIMPULAN DAN SARAN		38
5.1	Kesimpulan	38
5.2	Saran	38
DAFTAR PUSTAKA		39
HALAMAN LAMPIRAN		42



DAFTAR TABEL

Tabel 2. 1 Penyebab Kerusakan Terumbu Karang (Sulfa dkk., 2024)	9
Tabel 2. 2 Perbandingan Karakteristik Arsitektur MobileNetV2 dan DenseNet121	13
Tabel 4. 1 Sampel Dataset Citra Terumbu Karang	21
Tabel 4. 2 Akurasi dan Loss Validation MobileNetV2.....	26
Tabel 4. 3 Classification Report MobileNetV2	27
Tabel 4. 4 Akurasi dan Loss Validation DenseNet121	30
Tabel 4. 5 Classification Report DenseNet121	31
Tabel 4. 6 Perbandingan Akurasi dan Loss MobileNetV2 dan DenseNet121	33
Tabel 4. 7 Perbandingan Performa MobileNetV2 dan DenseNet121	34
Tabel 4. 8 Perbandingan Efisiensi MobileNetV2 dan DenseNet121	35
Tabel 4. 9 Perbandingan Percobaan Klasifikasi MobileNetV2 dan DenseNet121	38



DAFTAR GAMBAR

Gambar 2. 1	Arsitektur CNN (ANHAR & PUTRA, 2023)	11
Gambar 2. 2	Arsitektur MobileNetV2 (W. Wang dkk., 2020)	12
Gambar 2. 3	Arsitektur DenseNet121 (Saputra dkk., 2023)	13
Gambar 3. 1	<i>Flowchart</i> Alur Perancangan Sistem.....	15
Gambar 3. 2	<i>Flowchart</i> Data <i>Preprocessing</i>	16
Gambar 4. 1	Kurva <i>Training</i> dan <i>Validation</i> Akurasi MobileNetV2	25
Gambar 4. 2	Kurva <i>Training</i> dan <i>Validation Loss</i> MobileNetV2	25
Gambar 4. 3	<i>Confusion Matrix</i> MobileNetV2	28
Gambar 4. 4	Kurva <i>Training</i> dan <i>Validation</i> Akurasi DenseNet121.....	29
Gambar 4. 5	Kurva <i>Training</i> dan <i>Validation Loss</i> DenseNet121	29
Gambar 4. 6	<i>Confusion Matrix</i> DenseNet121.....	32
Gambar 4. 7	Tampilan <i>UI</i> Klasifikasi	36
Gambar 4. 8	Tampilan Hasil Klasifikasi	37
Gambar 4. 9	Tampilan Hasil Klasifikasi	37



DAFTAR LAMPIRAN



ABSTRAK

Terumbu karang adalah ekosistem laut yang mengandung keragaman biologi yang besar dan berperan besar dalam melestarikan keselarasan ekologi perairan. Namun, saat ini terumbu karang sedang mengalami fenomena pemutihan (*bleaching*), sehingga Klasifikasi terumbu karang merupakan langkah penting dalam mendukung upaya pelestarian ekosistem laut. Riset ini berfokus menelaah perbedaan antara dua arsitektur *Convolutional Neural Network* (CNN), yaitu MobileNetV2 dan DenseNet121, dalam proses klasifikasi gambar terumbu karang. Dataset yang digunakan adalah citra terumbu karang (*coral*) yang diperoleh dari Kaggle yang berjudul "*Bleached and Unbleached Corals Classification*". Dataset tersebut kemudian diproses melalui beberapa tahap, yaitu perubahan ukuran citra, pembagian data, pelatihan model, dan pengujian model MobileNetV2 dan DenseNet121. Setelah itu, dilakukan evaluasi terhadap kedua model menggunakan matriks kurva akurasi dan *loss*, akurasi dan *loss*, *classification report*, *confusion matrix*, waktu pelatihan dan penggunaan memori. Implementasi model menggunakan kedua model MobileNetV2 dan DenseNet121, dan bagian terakhir adalah analisis hasil. Riset ini memberikan bukti bahwa MobileNetV2 mempunyai performa yang lebih tinggi dibandingkan dengan DenseNet121. Hasil akurasi dari MobileNetV2 sebesar 92,54% dengan nilai *loss* sebesar 0,1280. Sedangkan hasil akurasi dari DenseNet121 sebesar 88,06% dengan nilai *loss* sebesar 0,3691. MobileNetV2 lebih unggul dengan waktu pelahitan yang lebih cepat, sehingga meningkatkan efisiensi. Selain itu, penggunaan memori dari MobileNetV2 lebih rendah jika dibandingkan dengan DenseNet121. Dengan demikian, Penelitian ini menyimpulkan bahwa arsitektur MobileNetV2 lebih unggul dalam melakukan klasifikasi terumbu karang, baik yang mengalami pemutihan (*bleached*) maupun yang tidak mengalami pemutihan (*unbleached*).

Kata kunci: *Convolutional Neural Network*, MobileNetV2, DenseNet121, klasifikasi.

ABSTRACT

Coral reefs are marine ecosystems that contain great biological diversity and play a major role in preserving the ecological harmony of the waters. However, currently coral reefs are experiencing a bleaching phenomenon, so coral reef classification is an important step in supporting efforts to preserve marine ecosystems. This research focuses on examining the differences between two Convolutional Neural Network (CNN) architectures, namely MobileNetV2 and DenseNet121, in the process of classifying coral reef images. The dataset used is coral reef images obtained from Kaggle entitled "Bleached and Unbleached Corals Classification". The dataset is then processed through several stages, namely image size changes, data sharing, model training, and testing of the MobileNetV2 and DenseNet121 models. After that, an evaluation of both models was carried out using the accuracy and loss curve matrix, accuracy and loss, classification report, confusion matrix, training time and memory usage. The model implementation uses both MobileNetV2 and DenseNet121 models, and the last part is the analysis of the

results. This research provides evidence that MobileNetV2 has a higher performance compared to DenseNet121. The accuracy results of MobileNetV2 are 92.54% with a loss value of 0.1280. While the accuracy results of DenseNet121 are 88.06% with a loss value of 0.3691. MobileNetV2 is superior with a faster training time, thus increasing efficiency. In addition, the memory usage of MobileNetV2 is lower when compared to DenseNet121. Thus, this study concludes that the MobileNetV2 architecture is superior in classifying coral reefs, both bleached and unbleached.

Keywords: Convolutional Neural Network, MobileNetV2, DenseNet121, classification.



BAB I

PENDAHULUAN

1.1 Latar Belakang

Terumbu karang adalah jaringan ekologi dengan beragamnya spesies hayati yang tinggi dan berfungsi penting dalam menjaga keseimbangan ekosistem perairan. Kesehatan dan produktivitas terumbu karang memiliki peran yang penting dalam berbagai aspek, seperti keberlanjutan ekosistem laut, pengelolaan sumber daya perikanan, pengembangan sektor wisata, serta pelestarian keanekaragaman hayati. Untuk mendukung pengelolaan terumbu karang secara berkelanjutan, diperlukan pemahaman yang mendalam mengenai status terumbu karang di suatu kawasan (Permatasari dkk., 2023). Kondisi terumbu karang di dunia mengalami penurunan kualitas akibat fenomena terumbu karang yang mengalami pemutihan, yang terjadi ketika suhu laut meningkat secara drastis. Peningkatan suhu air laut memicu stres pada karang dan menyebabkan kerusakan pada fotosintesis alga simbiotik (*zooxanthellae*), yang menyediakan nutrisi bagi karang. Ketika karang kehilangan alga ini, jaringan karang menjadi transparan dan tampak putih karena struktur kerangkanya terlihat jelas. Pemutihan ini, jika berlangsung dalam jangka waktu lama tanpa pemulihan suhu, dapat mengakibatkan kematian karang (Sully dkk., 2019).

Mengingat pentingnya terumbu karang dalam menjaga ekosistem laut dan mendukung keanekaragaman hayati, pemantauan dan pengklasifikasian kondisi terumbu karang yang efektif menjadi sangat penting. Penelitian lain juga menunjukkan penerapan *Convolutional Neural Network* (CNN) untuk mengklasifikasikan citra karang yang sehat dan yang mengalami *bleaching*, menggunakan pendekatan yang mengedepankan pengolahan citra secara otomatis, sehingga memungkinkan pemantauan yang lebih cepat dan akurat dalam konteks konservasi karang (S. Wang dkk., 2024).

MobileNet dipilih karena efisiensinya dalam pemrosesan citra pada perangkat dengan kinerja terbatas, berkat arsitekturnya yang optimal. Di sisi

lain, DenseNet-121 dipilih karena menggunakan struktur yang menghubungkan setiap lapisan ke setiap lapisan berikutnya, mengoptimalkan pembelajaran fitur dan meningkatkan efisiensi penggunaan parameter (Hadi, Rachmawanto, & Ali, 2024).

Riset terdahulu yang memanfaatkan CNN untuk klasifikasi terumbu karang telah menunjukkan hasil yang menjanjikan, namun masih terdapat peluang untuk mengeksplorasi performa arsitektur yang berbeda dalam konteks efisiensi dan akurasi. Dengan demikian, riset ini dimaksudkan untuk menganalisis perbedaan antara MobileNetV2 dan DenseNet121 dalam hal akurasi dan efisiensi komputasi dalam tugas klasifikasi *coral bleached* dan *unbleached*.

Pemilihan arsitektur MobileNetV2 dan DenseNet121 pada klasifikasi terumbu karang *bleached* dan *unbleached* adalah karena kedua model ini memiliki keunggulan dalam hal efisiensi komputasi dan kinerja yang optimal, sehingga sangat cocok untuk diimplementasikan pada perangkat dengan kemampuan komputasi terbatas (*edge devices*), seperti *website*, aplikasi *mobile*, dan *Internet Of things*.

Dengan membandingkan kedua arsitektur ini, dapat diperoleh pemahaman yang lebih baik mengenai *trade-off* antara akurasi, kompleksitas model, dan kecepatan inferensi. Analisis ini penting untuk menentukan model mana yang paling efektif dan praktis digunakan dalam sistem klasifikasi terumbu karang, khususnya pada *platform* berbasis *website* dan *mobile*, di mana keterbatasan komputasi dan kebutuhan akan respons waktu nyata menjadi pertimbangan utama.

1.2 Perumusan Masalah

Berdasarkan uraian latar belakang yang telah disampaikan, masalah utama yang dirumuskan dalam penelitian ini adalah:

1. Bagaimana performa MobileNetV2 dan DenseNet121 dalam melakukan klasifikasi terumbu karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*) berdasarkan citra terumbu karang?
2. Bagaimana efisiensi komputasi (waktu pelatihan dan ukuran model/penggunaan memori) MobileNetV2 dibandingkan dengan DenseNet121 pada tugas klasifikasi citra terumbu karang?

3. Arsitektur mana yang lebih unggul dalam hal *trade-off* antara akurasi dan efisiensi komputasi untuk klasifikasi terumbu karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*)?

1.3 Pembatasan Masalah

Proposal Penelitian: Perbandingan MobileNetV2 dan Densenet121 pada Klasifikasi Citra Terumbu Karang Menggunakan *Convolutional Neural Network* memiliki beberapa batasan, antara lain:

1. Penelitian ini dibatasi pada citra terumbu karang yang sudah tersedia dalam dataset publik yang bersumber dari Kaggle. Citra yang digunakan hanya yang menunjukkan kondisi terumbu karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*), tanpa memperhitungkan kondisi peralihan atau jenis kerusakan lainnya pada terumbu karang.
2. Penelitian ini hanya fokus pada dua kelas: karang yang mengalami pemutihan (*bleached*) dan karang yang tidak mengalami pemutihan (*unbleached*). Tidak mencakup klasifikasi lebih rinci seperti spesies karang, tingkat pemutihan, atau kerusakan lainnya.
3. Penelitian dibatasi pada perbandingan dua arsitektur *Convolutional Neural Network* (CNN), yaitu MobileNetV2 dan DenseNet121. Arsitektur *Convolutional Neural Network* (CNN) lainnya tidak akan dibahas atau dibandingkan dalam penelitian ini.
4. Implementasi dan pengujian model *Convolutional Neural Network* (CNN) dilakukan pada perangkat keras dan perangkat lunak tertentu yang mendukung deep learning, seperti GPU dan *framework deep learning* yang hanya menggunakan TensorFlow. Perangkat lunak yang digunakan mencakup Anaconda Navigator, Jupyter Notebook, dan Python. Optimalisasi performa mungkin terbatas pada spesifikasi perangkat keras yang digunakan.
5. Evaluasi performa dibatasi pada metrik-metrik tertentu, seperti akurasi, *precision*, *recall*, *F1-score*, serta waktu pelatihan dan ukuran model. Aspek lain seperti *latency* pada implementasi *real-time* atau performa pada perangkat *mobile* tidak dibahas dalam penelitian ini.

1.4 Tujuan

Proposal Penelitian: Perbandingan MobileNetV2 dan Densenet121 pada Klasifikasi Citra Terumbu Karang Menggunakan *Convolutional Neural Network* memiliki beberapa tujuan, antara lain:

1. Mengevaluasi Performa MobileNetV2 dan DenseNet121 dalam Klasifikasi Citra Terumbu Karang. Evaluasi meliputi akurasi, *confusion matrix*, *kurva training* dan *validasi*, *precision*, *recall*, dan *F1 Score*. Evaluasi ini diharapkan dapat memberikan penjelasan mengenai keandalan masing-masing model dalam mendeteksi kondisi karang secara akurat.
2. Membandingkan Efisiensi Komputasi MobileNetV2 dan DenseNet121 pada Tugas Klasifikasi Citra Terumbu Karang. Terutama dalam hal waktu pelatihan dan ukuran model/penggunaan memori. Dengan demikian, penelitian dapat menilai arsitektur mana yang lebih efisien.
3. Menentukan Arsitektur yang Lebih Unggul dalam keseimbangan terbaik (*Trade-off*) antara Akurasi dan Efisiensi Komputasi.

1.5 Manfaat

Diharapkan hasil dari penelitian ini dapat memberikan manfaat bagi:

1. Konservasi Lingkungan : Memberikan kontribusi dalam upaya konservasi terumbu karang dengan menyediakan sistem klasifikasi otomatis yang akurat dan efisien untuk mendeteksi pemutihan pada terumbu karang, sehingga tindakan pemulihan yang tepat dapat dilakukan secara tepat waktu.
2. Peneliti dan Akademisi : Menjadi referensi bagi peneliti dan akademisi di bidang ekologi, biologi laut, dan teknologi informasi, serta sebagai bahan kajian untuk penelitian lanjutan terkait aplikasi *deep learning* dalam klasifikasi citra lingkungan.
3. Praktisi dan Lembaga Pengelola Sumber Daya Laut : Membantu praktisi dan lembaga yang bergerak di bidang pengelolaan sumber daya laut dalam menerapkan teknologi pengawasan yang lebih baik, sehingga mereka dapat memantau kesehatan ekosistem terumbu karang secara lebih efektif.

4. Masyarakat Umum : Meningkatkan pemahaman masyarakat mengenai pentingnya perlindungan terumbu karang dan pengaruh perubahan iklim terhadap ekosistem laut, serta mendorong keterlibatan dalam program pelestarian lingkungan.

1.6 Sistematika Penulisan

Sistematika penulisan yang akan digunakan oleh penulis dalam pembuatan laporan tugas akhir adalah sebagai berikut:

BAB I : PENDAHULUAN

Pada BAB I menjelaskan tentang latar belakang, pemilihan judul, rumusan masalah, batasan masalah, tujuan penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Pada BAB II memuat tentang penelitian terdahulu dan landasan teori yang berkaitan untuk membantu memahami objek terumbu karang serta klasifikasinya, algoritma *Convolutional Neural Network* (CNN), arsitektur MobileNetV2, dan arsitektur DenseNet121 dalam melengkapi penelitian ini.

BAB III : METODE PENELITIAN

Pada BAB III menjelaskan proses penelitian yang dimulai dari pengumpulan data hingga evaluasi performa dan efisiensi dalam klasifikasi.

BAB IV : HASIL DAN ANALISIS PENELITIAN

Pada BAB IV berisi tentang pemaparan hasil penelitian yang dimulai dari hasil akhir sistem, klasifikasi data uji, performa dan efisiensi, serta mana yang lebih unggul dari kedua arsitektur dalam klasifikasi.

BAB V : KESIMPULAN DAN SARAN

Pada BAB V berisi rangkuman keseluruhan proses penelitian dari awal hingga akhir.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Berbagai riset yang telah ada sebelumnya dengan memanfaatkan algoritma *Convolutional Neural Network* (CNN) dalam klasifikasi biota laut, termasuk terumbu karang. Klasifikasi terumbu karang pada riset tersebut dilakukan dengan metode *transfer learning* memanfaatkan arsitektur MobileNet. Dataset yang digunakan mencakup total 1.582 citra terumbu karang, yang terbagi menjadi tiga kelas utama, yaitu 720 citra *bleached*, 150 citra *dead*, dan 712 citra *healthy*. Setelah melalui proses pengujian, penelitian tersebut berhasil mencapai akurasi sebesar 88% (Hadi, Rachmawanto, & Sari, 2024).

Studi yang dilakukan sebelumnya menunjukkan bahwa metode pengolahan citra dapat diterapkan untuk mengenali ciri-ciri atau fitur pada daun kentang yang terinfeksi penyakit. Tujuan dari studi ini yaitu untuk mengembangkan model CNN dan menganalisis perbedaan dari dua arsitektur, yakni DenseNet 121 dan MobileNet. Berdasarkan hasil pengujian, MobileNet terbukti lebih unggul dibandingkan DenseNet121 dalam mengklasifikasikan citra penyakit pada daun kentang, dengan akurasi sebesar 98.00%. Ini mengindikasikan jika MobileNet mempunyai efektivitas yang lebih unggul dalam mengklasifikasikan gambar pada penyakit daun kentang (Khultsum & Taufik, 2023).

Penelitian sebelumnya yang membandingkan arsitektur MobileNetV2 dan DenseNet121 digunakan untuk klasifikasi jenis tanaman rempah. Selama proses pelatihan, MobileNetV2 menunjukkan peningkatan akurasi yang cepat pada *epoch* awal, namun mengalami *overfitting* pada *epoch* berikutnya. Sebaliknya, DenseNet121 tidak mengalami masalah ini, dengan akurasi yang terus meningkat secara konsisten hingga mencapai *epoch* ke-30. Berdasarkan evaluasi akhir, DenseNet121 menunjukkan kinerja yang lebih baik daripada

MobileNetV2 dalam mengidentifikasi lima jenis rempah pada dataset empon-empon. (Hatur & Sabri, 2024).

Penelitian sebelumnya telah mengidentifikasi kondisi terumbu karang menggunakan teknik *deep learning* berbasis CNN, di mana citra diklasifikasikan menjadi dua kategori utama: terumbu karang Sehat dan Tertekan. Terumbu karang Tertekan adalah kondisi menengah antara terumbu karang yang sehat dan yang mengalami pemutihan. Dalam penelitian ini, model pra-pelatihan ResNet50 dan Inception V3 digunakan untuk mengklasifikasikan citra. Hasil klasifikasi dari Inception V3 dan ResNet50 meningkat menjadi 70% dan 55% setelah dilakukan penyesuaian terhadap *hyperparameter*, seperti *dropout* dan normalisasi *batch*. Dengan penyetelan yang sesuai, model CNN yang baru dikembangkan ini berhasil mencapai akurasi maksimal hingga 90% (Thamarai & Aruna, 2023).

Penelitian sebelumnya menggunakan berbagai arsitektur CNN dalam menganalisis kelayakan ban. Proses klasifikasi pada dataset uji ditunjukkan dengan menggunakan nilai fungsi rugi *cross-entropy*, serta dihitung akurasi, yang menggambarkan persentase citra ban yang berhasil diklasifikasikan dengan benar dibandingkan dengan total citra dalam data uji. Berdasarkan lima percobaan yang dilakukan, model DenseNet-121 menunjukkan kinerja yang paling unggul (Listyalina dkk., 2022).

2.2 Dasar Teori

2.2.1 Pemutihan Terumbu Karang dan Klasifikasinya

Pemutihan terumbu karang, atau *coral bleaching* merujuk pada fenomena di mana terumbu karang kehilangan warna alami dikarenakan stress dari lingkungan. Salah satu contoh pemutihan terumbu karang merujuk pada peristiwa di Taman Laut Bunaken, Taman Laut Bunaken termasyhur sebagai destinasi wisata bawah laut dengan berbagai jenis terumbu karang. Fenomena ini menimbulkan dampak buruk terhadap sektor pariwisata, mengurangi kemampuan terumbu karang dalam melindungi Pantai secara alami, serta merugikan para nelayan yang mengandalkan ekosistem terumbu karang (Sulfa dkk., 2024).

Penyebab kerusakan pada terumbu karang, meliputi:

Tabel 2. 1 Penyebab Kerusakan Terumbu Karang (Sulfa dkk., 2024)

No.	Penyebab	Persentase
1	Pemutihan terumbu karang disebabkan oleh kenaikan suhu air laut yang dipicu oleh fenomena <i>El Nino</i> .	60%
2	Situasi ini semakin memburuk yang disebabkan oleh aktivitas manusia dan mengakibatkan pemanasan global, di mana laut menyerap sekitar 93% dari kenaikan suhu bumi.	10%
3	Ketika terumbu karang terinfeksi penyakit, pemutihan terjadi saat karang menghadapi peristiwa gangguan serta kehilangan alga yang disebut <i>Zooxanthellae</i> , yang bertanggung jawab menyediakan pigmen warna pada terumbu karang.	5%
4	Penyebab lainnya yaitu polusi. Terumbu karang akan kembali pulih apabila kondisi lingkungan kembali normal.	10%
5	Pemakaian bahan peledak dan penangkapan ikan secara ilegal juga dapat merusak terumbu karang.	5%
6	Permukaan air laut yang meningkat.	10%

Beberapa faktor utama yang menyebabkan pemutihan karang di Taman Laut Bunaken adalah adanya Fenomena cuaca *El Nino* yang mengakibatkan suhu tinggi di permukaan laut. Selain itu, pemanasan global yang diakibatkan oleh aktivitas manusia, seperti pencemaran yang berasal dari limbah pertanian, *domestic*, dan sektor pariwisata, juga berkontribusi pada peningkatan suhu laut. Perubahan suhu laut sering kali membuat karang melepaskan *Zooxanthellae*, yang mengakibatkan penurunan proses pengapuran. Akibatnya, karang menjadi stres dan kehilangan warnanya.

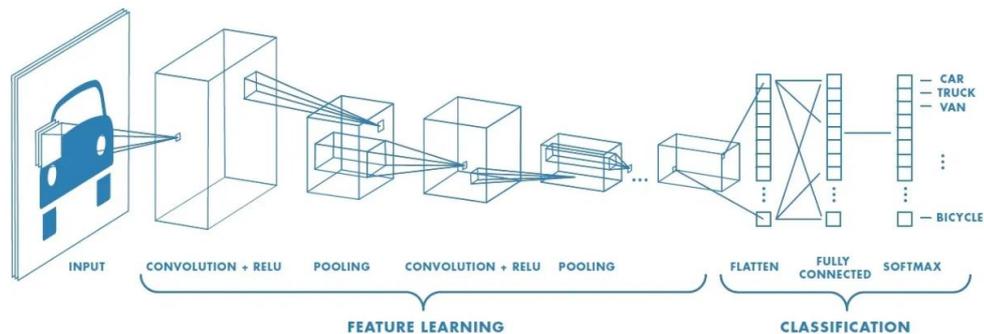
Pemutihan terumbu karang dapat mengakibatkan hilangnya habitat bagi ikan, yang berdampak pada nelayan dan komunitas setempat yang mengandalkan pada sumber daya tersebut. Hilangnya ekosistem terumbu karang juga menurunkan

jumlah ikan dan organisme laut lainnya, yang dapat berdampak negatif pada keberlanjutan perikanan dan mengancam ketahanan pangan bagi komunitas setempat (Sulfa dkk., 2024).

Klasifikasi terumbu karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*) ini adalah untuk mendeteksi secara akurat perubahan kondisi kesehatan terumbu karang yang disebabkan oleh pemutihan. Dengan klasifikasi ini, pemantauan ekosistem terumbu karang dapat dilakukan lebih cepat dan efektif, terutama dalam mengidentifikasi area yang terdampak oleh pemutihan akibat perubahan suhu laut. Hal ini penting untuk melindungi habitat ikan dan ekosistem laut lainnya yang bergantung pada terumbu karang. Data hasil klasifikasi juga dapat membantu dalam pengambilan keputusan untuk tindakan konservasi yang tepat, mencegah kerusakan lebih lanjut, serta memitigasi dampak negatif pada sektor perikanan dan keamanan pangan masyarakat lokal yang menggantungkan hidup pada keberadaan ekosistem karang. Selain itu, klasifikasi ini berperan dalam memprediksi potensi kerusakan di masa depan, sehingga langkah mitigasi dapat dilakukan lebih dini untuk menjaga keberlanjutan ekosistem laut dan mata pencaharian masyarakat pesisir.

2.2.2 Convolutional Neural Network (CNN)

CNN merupakan salah satu bentuk jaringan saraf tiruan yang diperuntukkan untuk menyelesaikan kendala pemrosesan gambar. CNN adalah pendekatan yang sangat efisien dalam melakukan klasifikasi, identifikasi, serta mengenali pola pada gambar. CNN mampu menginterpretasikan detail gambar dengan lebih akurat karena strukturnya mirip dengan cara otak manusia memproses data visual. Data yang diproses oleh CNN biasanya berbentuk dua dimensi, seperti citra atau audio, mengandalkan operasi konvolusi pada matriks dan bobot berstruktur empat dimensi yang terdiri dari kumpulan kernel konvolusi. Berdasarkan sifat proses konvolusi ini, CNN hanya dapat diterapkan pada data yang memiliki struktur dua dimensi (I Wayan Suartika E. P, 2016).



Gambar 2. 1 Arsitektur CNN (ANHAR & PUTRA, 2023)

Pada Gambar 2.1, terdapat dua komponen utama dalam proses kerja *Convolutional Neural Network* (CNN), yakni ekstraksi fitur dan klasifikasi. Dapat ditemukan tiga langkah utama pada tahap pembelajaran fitur, antarlain:

1. Operasi konvolusi adalah langkah inti dalam CNN yang bertujuan mengekstraksi fitur dari gambar masukan.
2. ReLU (*Rectified Linear Unit*) ialah fungsi aktivasi yang memberikan *output* bernilai 0 apabila *input* bernilai kurang dari 0. Sebaliknya, jika *input* bernilai positif, *output* akan sama dengan nilai *input* tersebut.
3. *Pooling* bertujuan guna menurunkan nilai parameter pada peta fitur dan mengekstrak informasi utama melalui proses *down-sampling*. Jenis *pooling* yang umum diterapkan meliputi *max pooling* dan *average pooling*.

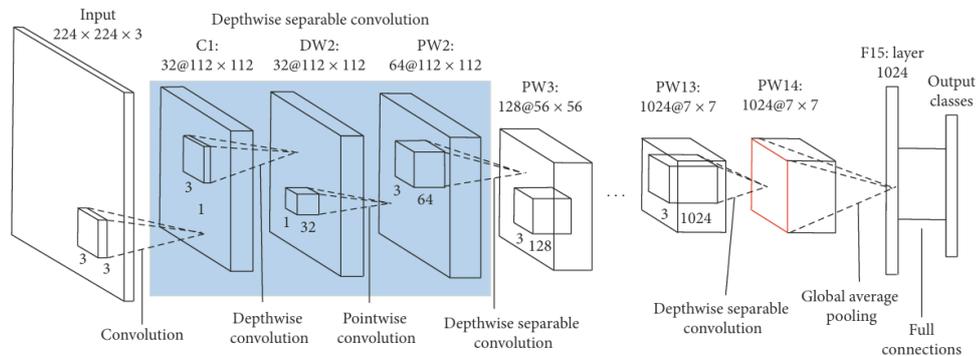
Pada tahap klasifikasi, prosesnya meliputi tiga langkah:

1. *Flatten*, adalah langkah yang mengubah *output* dari layer konvolusi menjadi sebuah kolom vektor fitur yang siap diproses oleh layer *fully connected*.
2. *Fully connected layer*, berfungsi mirip *Artificial Neural Network* (ANN) dalam menjalankan tugas klasifikasi.
3. *Softmax*, diterapkan untuk menghitung probabilitas masing-masing kelas, di mana kelas dengan probabilitas tertinggi akan menjadi hasil prediksi.

2.2.3 MobileNetV2

MobileNetV2 ialah sebuah rangka jaringan saraf tiruan yang dirancang oleh Google guna diimplementasikan untuk *mobile* atau *smartphone*. Sebagai pembaruan terkini dari arsitektur MobileNet yang ada sebelumnya, MobileNetV2 dirancang dengan dimensi yang lebih kecil, memungkinkan kinerja yang lebih efisien pada perangkat mobile yang mempunyai kapasitas terbatas. Arsitektur ini

memanfaatkan sistem *depthwise separable convolution*, di mana memungkinkan ukuran model menjadi lebih kecil dibandingkan model *Convolutional Neural Network* (CNN) lainnya. Selain itu, MobileNetV2 memiliki fitur-fitur lain yang meningkatkan efisiensinya, sebagai contoh penerapan *bottlenecks* serta lapisan residual, sehingga lebih hemat dalam konsumsi daya.

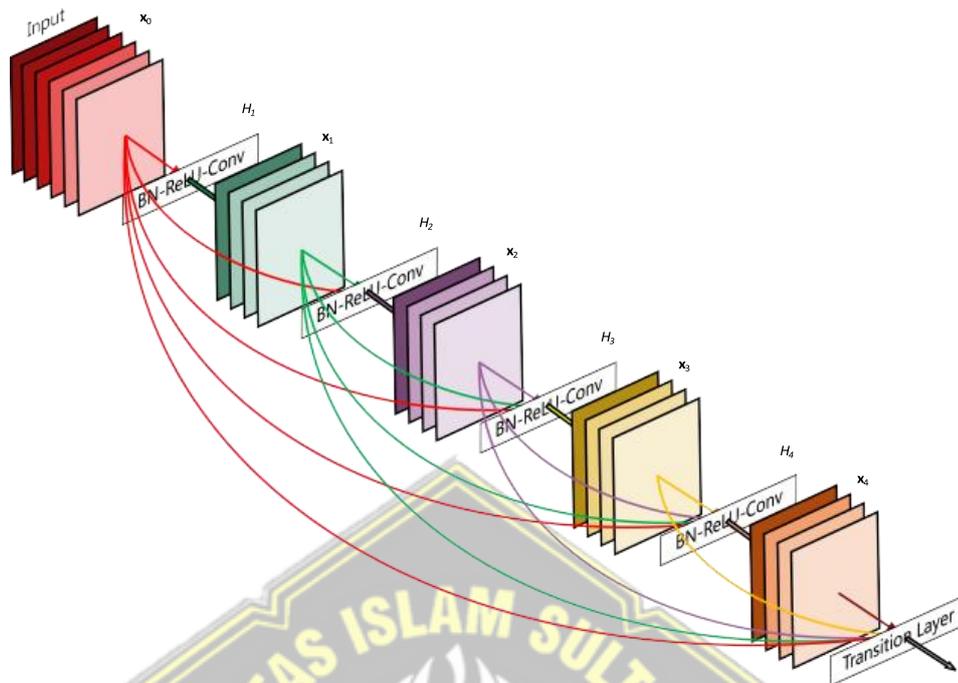


Gambar 2. 2 Arsitektur MobileNetV2 (W. Wang dkk., 2020)

Gambar 2.2 menunjukkan rangka dasar dari MobileNetV2, yang mencakup komponen konvolusi yang dikenal sebagai *depthwise separable convolution* serta komponen *fully-connected*. Di samping itu, MobileNetV2 dilengkapi dengan berbagai fitur lainnya yang meningkatkan efisiensinya dalam penggunaan energi, misalnya penerapan *bottlenecks* dan lapisan residual (Zhou dkk., 2020).

2.2.4 DenseNet121

DenseNet 121 adalah arsitektur jaringan saraf konvolusional, dalam hal ini masing-masing komponen terkoneksi secara langsung dengan komponen-komponen setelahnya yang lebih dalam pada susunan strukturnya. Dengan kata lain, komponen pertama terkoneksi dengan komponen kedua, ketiga, keempat, dan seterusnya, sementara lapisan berikutnya saling terkoneksi dengan lapisan ketiga, keempat, kelima, dan seterusnya. Pendekatan ini bertujuan agar memperkuat arus data di antara lapisan-lapisan dalam jaringan (Vellaichamy dkk., 2021).



Gambar 2. 3 Arsitektur DenseNet121 (Saputra dkk., 2023)

DenseNet121 dapat dijelaskan sebagai $5 + (6 + 12 + 24 + 16) * 2 = 121$, di mana angka 5 merujuk pada lapisan konvolusi dan *pooling*, angka 3 menunjukkan lapisan transisi (6, 12, 24), angka 1 merujuk pada lapisan klasifikasi (16), dan angka 2 menggambarkan *DenseBlock* yang terdiri dari konvolusi 1x1 dan 3x3 (Saputra dkk., 2023).

2.2.5 Perbandingan Arsitektur MobileNetV2 dan DenseNet121

Untuk memahami perbedaan performa dan karakteristik masing-masing model dalam proses klasifikasi terumbu karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*), berikut disajikan tabel perbandingan arsitektur MobileNetV2 dan DenseNet121 berdasarkan beberapa aspek penting, seperti tipe arsitektur, jumlah parameter, ukuran model, kecepatan inferensi, akurasi, kebutuhan memori, dan kemudahan dalam penerapan *transfer learning*.

Tabel 2. 2 Perbandingan Karakteristik Arsitektur MobileNetV2 dan DenseNet121

Aspek	MobileNetV2	DenseNet121
Tipe Arsitektur	<i>Depthwise Separable Convolution + Inverted Residual</i>	<i>Dense Convolutional Network (Dense Blocks)</i>

Jumlah Parameter	~3,4 juta (lebih sedikit)	~8 juta (lebih banyak)
Ukuran Model	Kecil (ringan dan efisien)	Lebih besar dibanding MobileNetV2
Kecepatan Inferensi	Cepat (cocok untuk real-time)	Lebih lambat karena arsitektur yang kompleks
Akurasi	Cukup tinggi untuk tugas klasifikasi ringan	Lebih tinggi pada tugas klasifikasi kompleks
Kebutuhan Memori	Rendah	Tinggi
<i>Transfer Learning</i>	Mudah diintegrasikan pada model ringan	Cocok untuk dataset besar dan kompleks

Tabel 2.2 menunjukkan perbandingan karakteristik arsitektur MobileNetV2 dan DenseNet121. MobileNetV2 menggunakan arsitektur *Depthwise Separable Convolution* dengan *Inverted Residual* yang menghasilkan model berukuran kecil dan ringan, dengan jumlah parameter sekitar 3,4 juta. Keunggulan ini membuat MobileNetV2 memiliki kecepatan inferensi yang cepat dan kebutuhan memori yang rendah, sehingga sangat cocok digunakan untuk aplikasi *real-time* di perangkat dengan komputasi ringan seperti *website* dan *mobile*. DenseNet121 menawarkan akurasi yang lebih tinggi untuk tugas klasifikasi yang lebih kompleks. DenseNet121 menggunakan *Dense Convolutional Network* dengan *Dense Blocks*, memiliki jumlah parameter sekitar 8 juta, dan ukuran model yang lebih besar, yang memerlukan memori dan daya komputasi lebih tinggi.

Perbandingan ini menunjukkan bahwa pemilihan arsitektur model bergantung pada kebutuhan antara efisiensi dan akurasi, di mana MobileNetV2 ideal untuk aplikasi ringan dan cepat, sedangkan DenseNet121 unggul dalam akurasi pada data yang lebih kompleks.

BAB III

METODE PENELITIAN

Dalam penelitian ini, dataset yang digunakan adalah citra terumbu karang (*coral*) yang diperoleh dari Kaggle. Dataset tersebut kemudian diproses melalui beberapa tahap, yaitu perubahan ukuran citra, pembagian data, pelatihan model, dan pengujian model. Setelah itu, dilakukan evaluasi terhadap model, implementasi model menggunakan MobileNetV2 dan DenseNet121, dan bagian terakhir adalah analisis hasil. Adapun tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut :

1. Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari citra karang yang terbagi dalam dua kelas, yaitu terumbu karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*). Sumber data diperoleh dari Kaggle.

a. Kelas *Bleached*

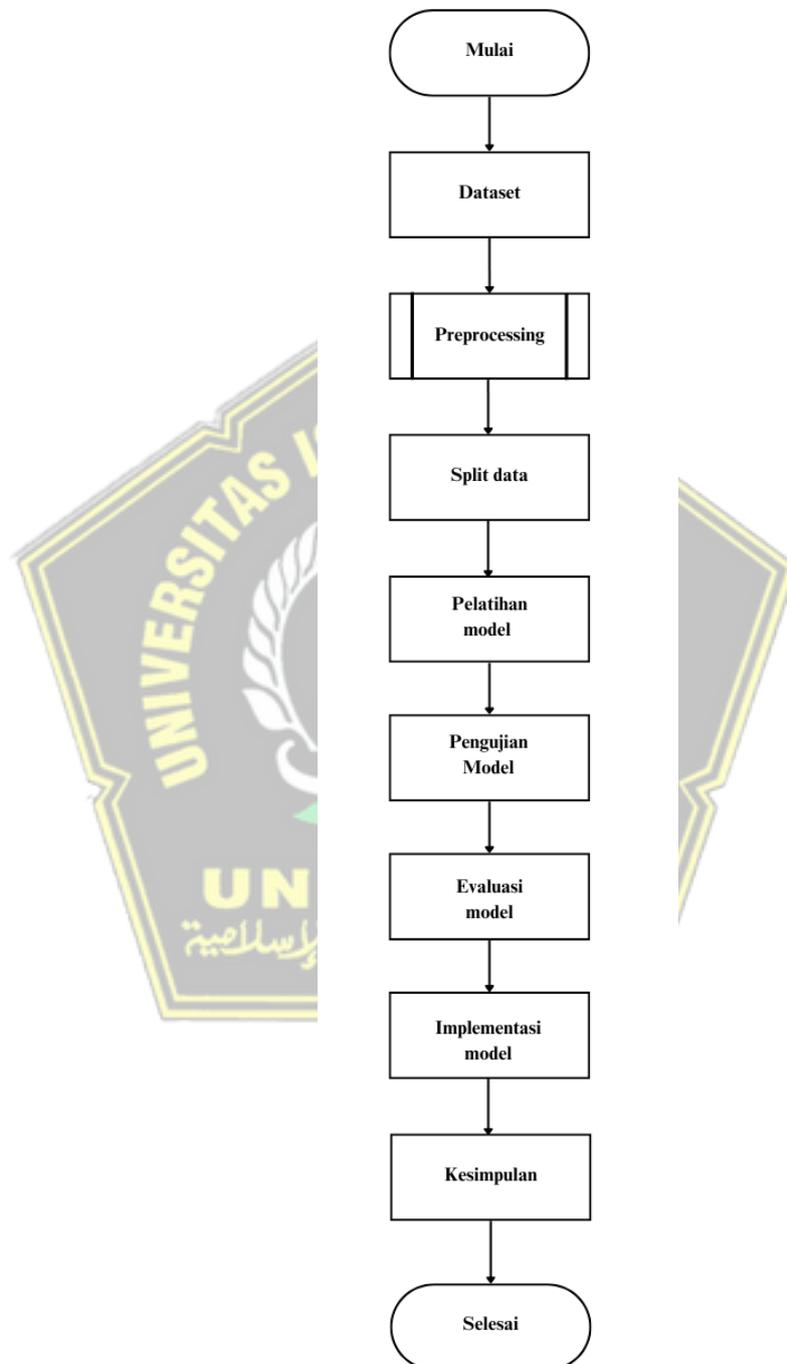
Terdapat 184 citra dan memiliki ukuran 227 x 227 piksel. Kondisi terumbu karang yang kehilangan warna dan terlihat putih. Akibat dari peningkatan suhu air, pemanasan global, polusi, atau perubahan salinitas.

b. Kelas *Unbleached*

Terdapat 158 citra dan memiliki ukuran 227 x 227 piksel. Kondisi terumbu karang yang sehat dan tidak mengalami pemutihan (berwarna-warni).

2. Perancangan Model

Tahapan perancangan model pada penelitian ini ditunjukkan pada *flowchart* gambar 3.1 berikut :

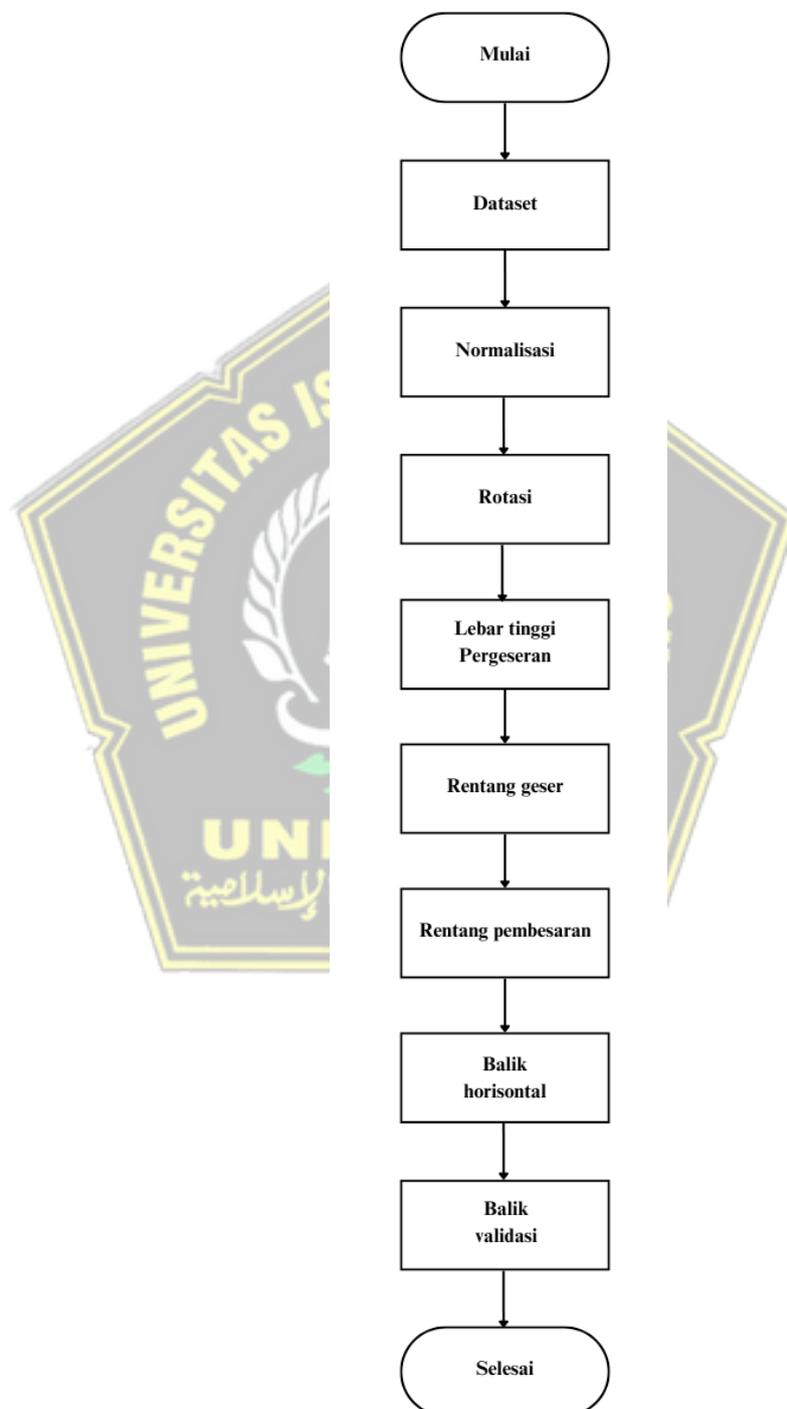


Gambar 3. 1 *Flowchart* Alur Perancangan Sistem

Proses dalam perancangan model ini, secara umum memiliki 6 tahap yang akan diuraikan sebagai berikut :

1) *Preprocessing*

Tahap *preprocessing* dilakukan untuk mempersiapkan data agar dapat digunakan oleh model. Berikut merupakan *flowchart* pada data *preprocessing*.



Gambar 3. 2 *Flowchart* Data *Preprocessing*

Tahap *preprocessing* dilakukan untuk mempersiapkan data agar dapat digunakan oleh model. Berikut merupakan penjelasan setiap tahap *preprocessing* yang dilakukan :

- a) Langkah pertama dalam *preprocessing* adalah Menormalisasi nilai piksel agar berada pada rentang (0, 1). Nilai piksel yang lebih kecil membantu stabilitas dan konvergensi model selama pelatihan.
- b) Kemudian dilakukan rotasi citra secara acak dalam satuan derajat. Membantu model belajar dari citra yang tampak dari berbagai sudut.
- c) Lalu, menggeser citra secara acak lebar atau tinggi citra. Berguna untuk membuat model lebih tangguh terhadap variasi posisi objek dalam citra.
- d) Selanjutnya, menerapkan *shear* transformasi (distorsi miring). Membuat model lebih adaptif terhadap perubahan sudut kemiringan objek.
- e) Kemudian, memperbesar atau memperkecil citra secara acak. Membantu model belajar dari citra dengan skala objek yang berbeda-beda.
- f) Setelah itu, membalik citra secara horizontal secara acak. Augmentasi ini sangat berguna terutama jika objek dalam citra memiliki simetri horizontal.
- g) Terakhir, Memisahkan beberapa persen dari data sebagai data validasi. Memungkinkan model untuk dievaluasi pada subset data yang berbeda dari data pelatihan tanpa harus membuat generator baru.

2) *Split Data*

Pada tahap ini, untuk memuat dan membagi dataset menjadi data pelatihan dan validasi secara langsung. Mengubah ukuran citra menjadi 224x224 piksel sesuai dengan input yang dibutuhkan oleh MobileNetV2 dan DenseNet121, dan mengatur jumlah sampel dalam satu batch yang akan diproses sekaligus.

3) Pelatihan Model

Pada tahap ini, dua arsitektur *Convolutional Neural Network* (CNN), yaitu MobileNetV2 dan DenseNet121 dilatih menggunakan data *training*.

Sehingga model dapat mempelajari fitur-fitur penting yang membedakan antara kondisi terumbu karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*).

4) Pengujian Model

Untuk memastikan kinerja dari model yang sudah dilatih, perlu dilakukan pengujian model dengan menggunakan data validasi. Hasil dari pengujian ini dapat dijadikan bahan untuk analisa dan evaluasi model, yang dapat dilihat melalui, kurva pelatihan dan validasi pada akurasi dan *loss*, nilai *accuracy*, *precision*, *recall*, *f1-score*, *confusion matrix*, serta waktu pelatihan dan ukuran modelnya.

5) Evaluasi Model

Evaluasi Perbandingan kedua model dilakukan dengan 2 paramater yaitu performa dan efisiensi. Evaluasi Performa yang pertama adalah kurva pelatihan dan validasi pada akurasi membantu memvisualisasikan perubahan akurasi model selama pelatihan pada data latih dan validasi, sehingga kita bisa melihat kemampuan model dalam belajar dan menangani data baru. Kurva pelatihan dan validasi pada *loss* digunakan untuk melacak penurunan kesalahan (*loss*) model selama pelatihan, yang penting untuk mendeteksi apakah model sudah optimal atau justru mengalami *overfitting* atau *underfitting*. Selanjutnya akurasi digunakan untuk mengukur persentase prediksi yang benar. Lalu, *precision*, *recall*, dan *F1-score* digunakan untuk mengevaluasi keseimbangan antara prediksi positif dan negatif. Kemudian, *confusion matrix* digunakan untuk memahami bagaimana model mendistribusikan prediksinya pada setiap kelas. Setelah melakukan evaluasi performa dari kedua model, dilakukan juga evaluasi efisiensi dalam hal waktu pelatihan dan ukuran model/penggunaan memori.

6) Implementasi Model

Pada tahap ini, kedua model diterapkan dalam bentuk *web* bertujuan untuk memudahkan dalam melakukan klasifikasi terumbu karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*). Pada *web*

tersebut, terdapat kolom unggah *file* untuk mengunggah citra terumbu karang yang ingin diklasifikasikan.

Setelah citra diunggah, sistem akan memproses citra dan memberikan hasil klasifikasi. Hasil klasifikasi akan ditampilkan berupa informasi apakah citra tersebut menunjukkan terumbu karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*). Selain itu, citra yang diunggah juga akan ditampilkan kembali di halaman hasil untuk memberikan konteks visual dan memastikan *input* yang dimasukkan sudah sesuai.



BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 HASIL

4.1.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini terdiri dari citra karang yang terbagi dalam dua kelas, yaitu terumbu karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*). Sumber data diperoleh dari Kaggle dengan nama “*Bleached and Unbleached Corals Classification*”. Dataset yang digunakan dalam penelitian ini terdiri dari citra terumbu karang yang berasal dari Great Barrier Reef, Australia. Sebagian besar citra tersebut diperoleh melalui pemotretan menggunakan drone bawah air (Jamil dkk., 2021).

Sumber dataset: <https://www.kaggle.com/datasets/sonainjamil/bhd-corals>.

a. Kelas *Bleached*

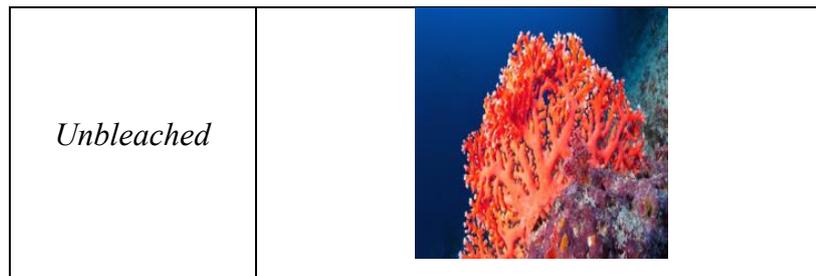
Terdapat 184 citra dan memiliki ukuran 227 x 227 piksel. Kondisi terumbu karang yang kehilangan warna dan terlihat putih. Akibat dari peningkatan suhu air, pemanasan global, polusi, atau perubahan salinitas.

b. Kelas *Unbleached*

Terdapat 158 citra dan memiliki ukuran 227 x 227 piksel. Kondisi terumbu karang yang sehat dan tidak mengalami pemutihan (berwarna-warni). Berikut sampel citra Terumbu karang pada setiap kelas, dapat dilihat pada tabel 4.1.

Tabel 4. 1 Sampel Dataset Citra Terumbu Karang

Kelas	Sampel Citra
<i>Bleached</i>	



4.1.2 Preprocessing

Tahap *preprocessing* dilakukan untuk mempersiapkan data agar dapat digunakan oleh model. Berikut merupakan penjelasan setiap tahap *preprocessing* yang dilakukan :

1. Langkah pertama adalah menormalisasi nilai piksel agar berada pada rentang (0, 1). Nilai piksel yang lebih kecil membantu stabilitas dan konvergensi model selama pelatihan.
2. Kemudian dilakukan rotasi citra secara acak dalam satuan derajat. Membantu model belajar dari citra yang tampak dari berbagai sudut.
3. Lalu, menggeser citra secara acak lebar atau tinggi citra. Berguna untuk membuat model lebih tangguh terhadap variasi posisi objek dalam citra.
4. Selanjutnya, menerapkan *shear* transformasi (distorsi miring). Membuat model lebih adaptif terhadap perubahan sudut kemiringan objek.
5. Kemudian, memperbesar atau memperkecil citra secara acak. Membantu model belajar dari citra dengan skala objek yang berbeda-beda.
6. Setelah itu, membalik citra secara horizontal secara acak. Augmentasi ini sangat berguna terutama jika objek dalam citra memiliki simetri horizontal.
7. Terakhir, memisahkan beberapa persen dari data sebagai data validasi. Memungkinkan model untuk dievaluasi pada subset data yang berbeda dari data pelatihan tanpa harus membuat generator baru.

4.1.3 Split Data

Pada tahap ini, dataset yang terdiri dari dua kelas, yaitu *Bleached* dan *Unbleached*, dilakukan proses pemuatan dan pembagian data menjadi data pelatihan dan validasi. Proses ini dilakukan dengan mempertimbangkan proporsi

data yang ideal agar model dapat dilatih secara optimal. Langkah-langkah yang dilakukan adalah sebagai berikut:

1. Penyesuaian Resolusi Citra

Semua citra dalam dataset, baik dari kelas *Bleached* maupun *Unbleached*, diubah ukurannya menjadi 224 x 224 piksel. Ukuran ini disesuaikan dengan persyaratan *input* arsitektur MobileNetV2 dan DenseNet121 yang digunakan dalam penelitian ini.

2. Pembagian Dataset

Dataset dibagi menjadi dua subset utama:

- a. Sebanyak 80% dari total data, yaitu 275 citra, akan digunakan sebagai data pelatihan untuk melatih model. Data ini berfungsi untuk membantu model mempelajari pola dari kedua kelas.
- b. Data validasi sebanyak 20% (67 Citra) dari total data digunakan untuk mengevaluasi performa model selama proses pelatihan.

Pembagian dataset dilakukan secara acak (*random split*) untuk memastikan distribusi data di setiap kelas tetap proporsional dan representatif dalam subset data *training* dan validasi.

3. Pengaturan *Batch Size*

Jumlah sampel yang diproses sekaligus dalam satu iterasi (*batch size*) diatur agar sesuai dengan kapasitas perangkat keras (GPU/CPU) yang digunakan. Dalam penelitian ini, *batch size* diatur dengan nilai 32, yang mempermudah proses pelatihan model secara efisien.

4.1.4 Pelatihan Model

Pada tahap ini, model dilatih menggunakan dataset yang telah dibagi menjadi data pelatihan dan data validasi. Proses pelatihan dilakukan untuk mengoptimalkan parameter pada arsitektur MobileNetV2 dan DenseNet121 sehingga model dapat mengenali pola dari kedua kelas, yaitu *Coral Bleached* dan *Unbleached*. Berikut adalah penjelasan rinci mengenai proses pelatihan yang dilakukan:

1. Proses Pelatihan

Pelatihan dilakukan dengan menggunakan fungsi *model.fit()*, yang bekerja dengan langkah-langkah berikut:

- a. Data pelatihan: Model menerima data pelatihan secara bertahap dalam bentuk batch melalui generator. Data ini telah melalui *preprocessing* *resizing* ke ukuran 224x224 piksel.
 - b. Langkah per *epoch*: Setiap *epoch* terdiri dari sejumlah langkah yang dihitung berdasarkan total data pelatihan dibagi ukuran *batch* 32. Dalam setiap langkah, model melakukan pembaruan bobot melalui mekanisme *backpropagation* dan optimasi untuk meminimalkan *loss*.
 - c. Evaluasi dengan data validasi: Setelah menyelesaikan seluruh langkah dalam satu *epoch*, model dievaluasi menggunakan data validasi dari generator. Evaluasi ini bertujuan untuk memantau kinerja model pada data yang tidak dilatih. Evaluasi ini dilakukan secara bertahap sesuai dengan jumlah langkah validasi, yang dihitung berdasarkan total data validasi dibagi ukuran *batch* 32.
 - d. Iterasi *epoch*: Proses pelatihan dilakukan selama maksimal 100 *epoch*, di mana setiap *epoch* mencakup seluruh data pelatihan dan validasi.
2. Mekanisme *Early Stopping*

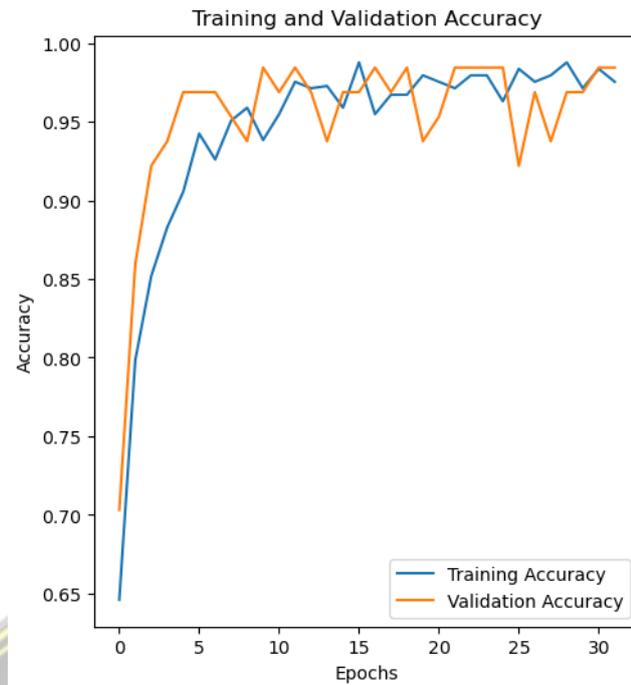
Untuk menghindari *overfitting*, diterapkan mekanisme *Early Stopping*. Mekanisme ini memonitor performa model, seperti validasi *loss*, selama pelatihan. Jika performa tidak menunjukkan peningkatan dalam sejumlah *epoch* tertentu (contoh: 10 *epoch* berturut-turut), pelatihan akan dihentikan lebih awal secara otomatis. Langkah ini membantu mencegah pelatihan yang tidak diperlukan dan menghemat waktu komputasi.

4.1.5 Pengujian Model

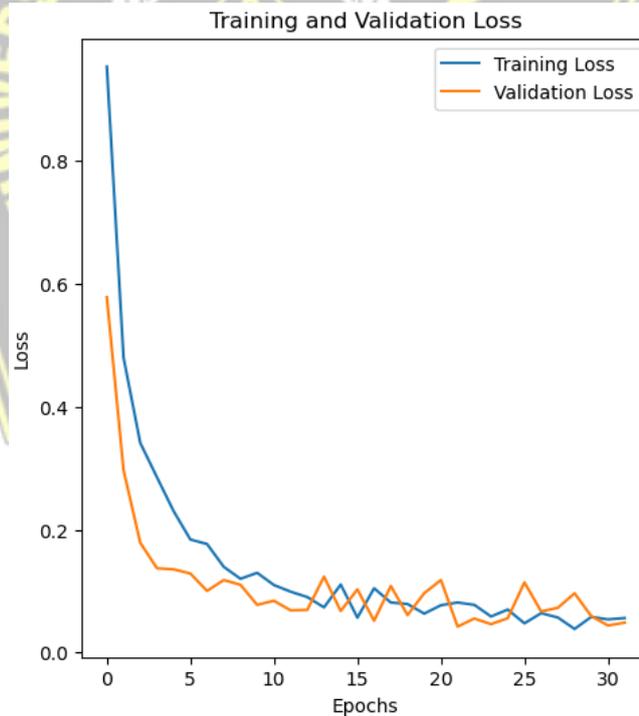
Pengujian kedua model dilakukan menggunakan data validasi. Pengujian meliputi performa dan efisiensi. Berikut pengujian yang dilakukan pada model MobileNetV2 dan DenseNet121:

1. MobileNetV2
 - a. Kurva *Training* dan *Validation*

Kurva ini membantu memahami performa model selama proses pelatihan dan validasi.



Gambar 4. 1 Kurva *Training* dan *Validation* Akurasi MobileNetV2



Gambar 4. 2 Kurva *Training* dan *Validation Loss* MobileNetV2

Kurva pada gambar 4.1 dan 4.2 menunjukkan hasil pengujian model MobileNetV2 selama proses pelatihan dan validasi. Berikut penjelasannya:

- 1) Akurasi pelatihan meningkat secara stabil hingga mencapai tingkat mendekati 1.0 (100%).
 - 2) Akurasi validasi mengalami peningkatan signifikan di awal proses pelatihan dan cenderung mengikuti pola dari akurasi pelatihan, meskipun terdapat sedikit fluktuasi menjelang akhir pelatihan.
 - 3) Akurasi validasi dan pelatihan cukup sejajar, menunjukkan bahwa model tidak mengalami *overfitting* yang signifikan.
 - 4) *Loss* pelatihan menurun secara signifikan hingga mendekati nol seiring dengan bertambahnya *epoch*.
 - 5) *Loss* validasi juga turun secara cepat pada awal pelatihan dan tetap stabil mendekati *training loss*, meskipun ada sedikit fluktuasi di beberapa titik.
- b. Akurasi dan *Loss Validation*

Validation loss dan *validation accuracy* digunakan untuk menguji performa model pada data validasi.

Tabel 4. 2 Akurasi dan *Loss Validation* MobileNetV2

<i>Validation Loss</i>	0.1280210316181183
<i>Validation Accuracy</i>	0.9253731369972229

Berikut Penjelasan mengenai hasil *Validation loss* dan *validation accuracy* pada gambar di atas:

- 1) Nilai *loss* **0.1280** menunjukkan bahwa kesalahan yang dibuat model pada data validasi sangat kecil. Semakin kecil nilai ini, semakin baik performa model.
- 2) akurasi model adalah **92.54%**, yang menunjukkan performa sangat baik karena model mampu memprediksi dengan benar hampir semua data validasi.

c. *Classification Report*

Classification Report digunakan untuk mengukur performa model klasifikasi, seperti *precision*, *recall*, *f1-score*, dan *support*, untuk setiap kelas, serta akurasinya.

Tabel 4. 3 *Classification Report* MobileNetV2

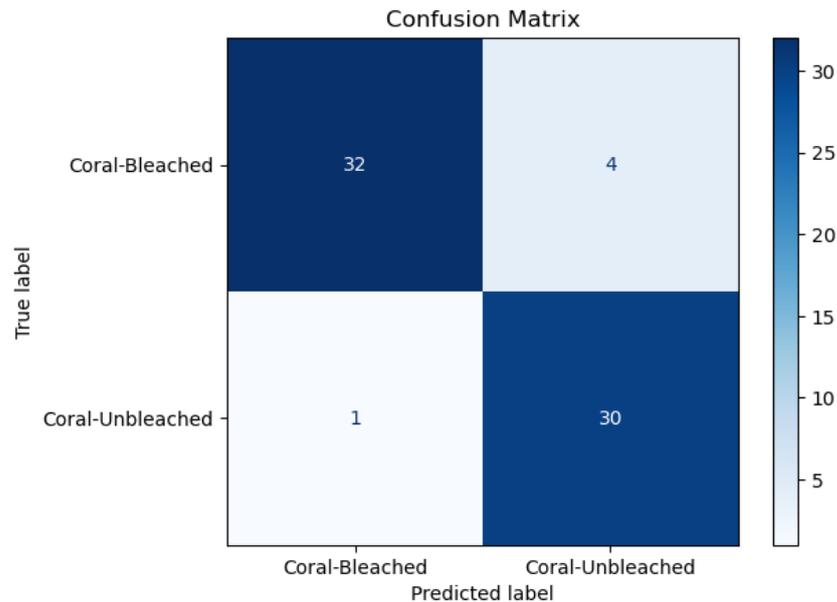
	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<i>Coral-Bleached</i>	0.97	0.89	0.93	36
<i>Coral-Unbleached</i>	0.88	0.97	0.92	31
<i>Accuracy</i>			0.93	67
<i>Macro avg</i>	0.93	0.93	0.93	67
<i>Weighted avg</i>	0.93	0.93	0.93	67

Tabel 4.3 menunjukkan hasil *Clasification Report* pada model MobileNetV2, berikut penjelasannya:

- 1) *Precision* dan *recall* untuk kedua kelas (*Coral-Bleached* dan *Coral-Unbleached*) cukup tinggi, menunjukkan model mampu mengenali kedua kelas dengan baik.
- 2) *F1-score* menggambarkan keseimbangan yang baik antara *precision* dan *recall*, dengan skor masing-masing 93 untuk kelas *Coral-Bleached* dan 92 untuk kelas *Coral-Unbleached*.
- 3) Akurasi model adalah 93%, yang berarti model benar-benar dapat mengklasifikasikan sebagian besar data validasi dengan baik.

d. *Confusion Matrix*

Confusion Matrix berfungsi untuk mengevaluasi performa model klasifikasi dengan menyajikan jumlah prediksi yang benar (*True Positive* dan *True Negative*) serta jumlah prediksi yang salah (*False Positive* dan *False Negative*) dalam format matriks.



Gambar 4. 3 *Confusion Matrix* MobileNetV2

Matriks pada gambar 4.3 menunjukkan hasil dari pengujian, berikut penjelasannya:

- 1) 32 Data dengan label sebenarnya *Coral-Bleached* yang diprediksi dengan benar sebagai *Coral-Bleached* (*True Positive*).
- 2) 30 Data dengan label sebenarnya *Coral-Unbleached* yang diprediksi dengan benar sebagai *Coral-Unbleached* (*True Negative*).
- 3) 4 Data dengan label sebenarnya *Coral-Bleached* yang diprediksi salah sebagai *Coral-Unbleached* (*False Positive*).
- 4) 1 Data dengan label sebenarnya *Coral-Unbleached* yang diprediksi salah sebagai *Coral-Bleached* (*False Negative*).

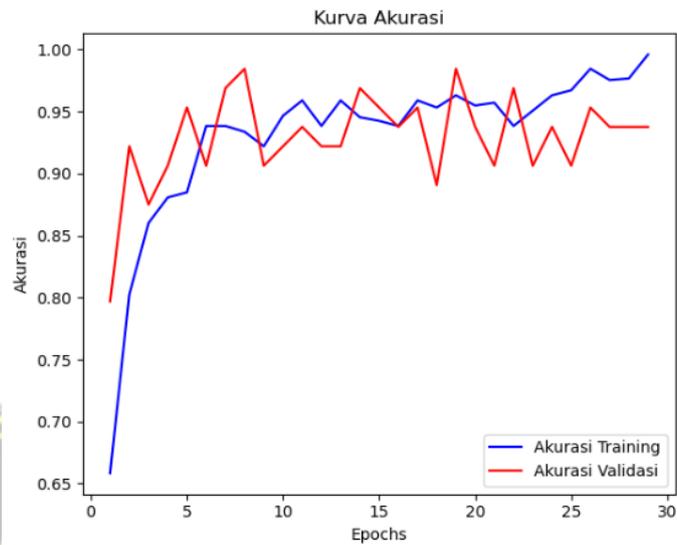
e. Waktu Pelatihan dan Ukuran Model

Rata-rata waktu per *epoch* sekitar 10-12 detik (beberapa *epoch* awal sedikit lebih lambat, sekitar 18 detik). Total waktu pelatihan yang dibutuhkan adalah 6 menit untuk 32 *epoch*, Penggunaan memori adalah 10,7 MB. Hal ini tergolong cepat dan ringan untuk sebuah model *deep learning*, yang menunjukkan bahwa MobileNetV2 sangat efisien untuk pelatihan.

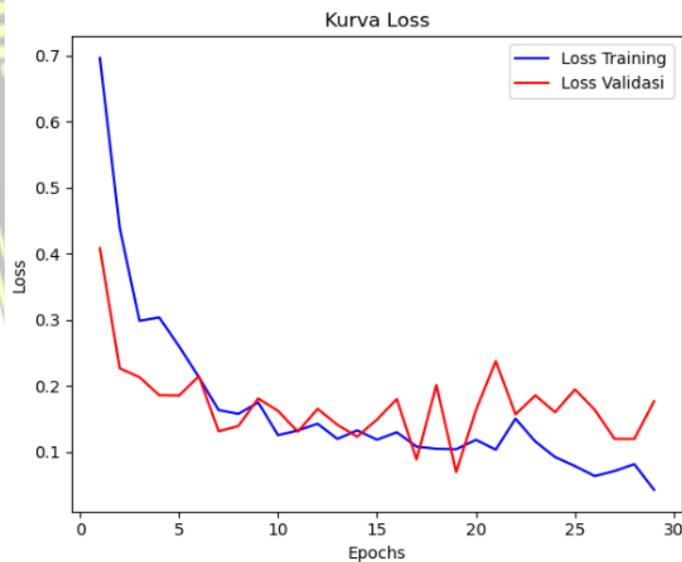
2. DenseNet121

a. Kurva *Training* dan *Validation*

Kurva ini membantu memahami performa model selama proses pelatihan dan validasi.



Gambar 4. 4 Kurva *Training* dan *Validation* Akurasi DenseNet121



Gambar 4. 5 Kurva *Training* dan *Validation* Loss DenseNet121

Kurva pada gambar 4.4 dan 4.5 menunjukkan hasil pengujian model DenseNet121 selama proses pelatihan dan validasi. Berikut penjelasannya:

- 1) Pada data *training*, akurasi model mengalami peningkatan signifikan selama beberapa *epoch* awal dan hampir mencapai nilai maksimum (~1.0) setelah sekitar 10-15 *epoch*.
- 2) Akurasi validasi juga menunjukkan peningkatan signifikan pada awal pelatihan, meskipun terlihat lebih fluktuatif dibandingkan akurasi *training*.
- 3) Nilai akurasi validasi stabil di akhir pelatihan tetapi sedikit lebih rendah dibandingkan akurasi *training*, menandakan kemungkinan adanya sedikit *overfitting*.
- 4) *Loss* pada data *training* menurun secara konsisten selama pelatihan, dengan penurunan yang tajam pada beberapa *epoch* pertama.
- 5) Fluktuasi *loss* validasi menandakan bahwa model mungkin mengalami kesulitan kecil dalam generalisasi atau dipengaruhi oleh variasi dalam data validasi.

b. Akurasi dan *Loss*

Validation loss dan *validation accuracy* digunakan untuk menguji performa model pada data validasi.

Tabel 4. 4 Akurasi dan *Loss Validation* DenseNet121

<i>Validation Loss</i>	0.3691
<i>Validation Accuracy</i>	0.8806

Berikut penjelasan Tabel 4.4 mengenai hasil *Validation Loss* dan *Validation Accuracy* pada gambar di atas:

- 1) Nilai 0.3691 menunjukkan bahwa model memiliki *error* yang relatif rendah pada data validasi.
- 2) Nilai 88.06%, hal ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang memadai, ditandai dengan tingkat akurasi yang tinggi pada data validasi.

c. *Classification Report*

Classification Report digunakan untuk mengukur performa model klasifikasi, seperti *precision*, *recall*, *f1-score*, dan *support*, untuk setiap kelas, serta akurasinya.

Tabel 4. 5 *Classification Report* DenseNet121

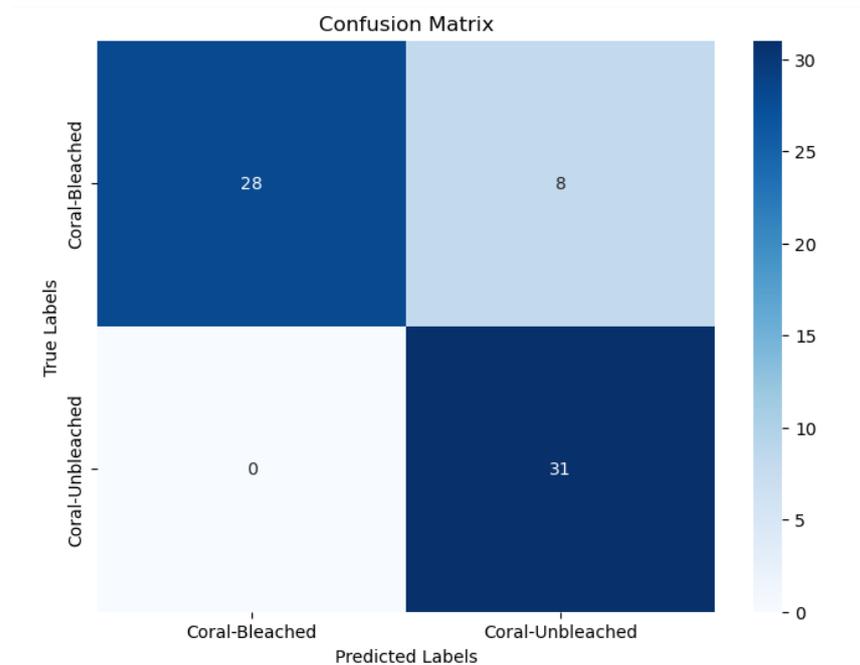
	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<i>Coral-Bleached</i>	1.00	0.78	0.88	36
<i>Coral-Unbleached</i>	0.79	1.00	0.89	31
<i>Accuracy</i>			0.88	67
<i>Macro avg</i>	0.90	0.89	0.88	67
<i>Weighted avg</i>	0.91	0.88	0.88	67

Tabel 4.5 menunjukkan hasil *Clasification Report* pada model DenseNet121, berikut penjelasannya:

- 1) Model memiliki *precision* tinggi pada kelas "*Coral-Bleached*" (1.00) dan *recall* tinggi pada kelas "*Coral-Unbleached*" (1.00), menunjukkan kemampuan mendeteksi kelas-kelas ini dengan sangat baik.
- 2) *F1-score* untuk kedua kelas hampir seimbang (~0.88-0.89), yang menunjukkan performa yang stabil.
- 3) Akurasi model adalah 88%, yang berarti model sudah sangat bagus mengklasifikasikan sebagian besar data validasi dengan baik.

d. *Confusion Matrix*

Confusion Matrix digunakan untuk mengevaluasi kinerja model klasifikasi dengan menyajikan jumlah prediksi yang tepat (*True Positive* dan *True Negative*) serta prediksi yang keliru (*False Positive* dan *False Negative*) dalam bentuk tabel matriks.



Gambar 4. 6 *Confusion Matrix* DenseNet121

Matriks pada gambar 4.6 menunjukkan hasil dari pengujian kinerja model DenseNet121 dalam mengklasifikasi data ke dalam dua kelas, berikut penjelasannya:

- 1) 28 Data dengan label sebenarnya *Coral-Bleached* yang diprediksi dengan benar sebagai *Coral-Bleached* (*True Positive*).
- 2) 31 Data dengan label sebenarnya *Coral-Unbleached* yang diprediksi dengan benar sebagai *Coral-Unbleached* (*True Negative*)
- 3) 8 Data dengan label sebenarnya *Coral-Bleached* yang diprediksi salah sebagai *Coral-Unbleached* (*False Positive*).
- 4) 0 Data dengan label sebenarnya *Coral-Unbleached* yang diprediksi salah sebagai *Coral-Bleached* (*False Negative*).

e. Waktu Pelatihan dan Ukuran Model

Rata-rata waktu per *epoch* sekitar 36-54 detik. Beberapa *epoch* memerlukan waktu lebih lama (misalnya *Epoch* 1 membutuhkan 54 detik per langkah, sementara *Epoch* 2 dan setelahnya umumnya sekitar 36-39 detik per langkah). Total waktu pelatihan yang dibutuhkan adalah 18 menit untuk 29 *epoch*, Penggunaan memori adalah 29 MB.

4.2 Analisa

4.2.1 Evaluasi Model

Pada bagian ini, dilakukan analisis hasil perbandingan dari pengujian model yang sudah dilakukan. Evaluasi ini bertujuan untuk menilai kinerja kedua model berdasarkan beberapa aspek, yaitu kurva pelatihan dan validasi, akurasi, *loss*, *classification report*, *confusion matrix* dalam pelatihan. Setiap aspek dievaluasi untuk menentukan model mana yang memiliki performa terbaik dan mana yang lebih efisien dalam hal penggunaan sumber daya.

Berikut hasil perbandingan evaluasi pada kedua model yaitu, MobileNetV2 dan DenseNet121:

1. Kurva Pelatihan dan Validasi MobileNetV2 dengan DenseNet121

MobileNetV2 lebih konsisten, stabil, dan efisien dalam pelatihan dibandingkan dengan DenseNet121. Model ini tidak hanya memberikan akurasi tinggi, tetapi juga menunjukkan generalisasi yang lebih baik tanpa indikasi *overfitting*. Meskipun DenseNet121 juga sangat baik, sedikit fluktuasi dan *overfitting* ringan pada kurva validasinya.

2. Akurasi dan *Loss*

Tabel 4. 6 Perbandingan Akurasi dan *Loss* MobileNetV2 dan DenseNet121

Arsitektur	Akurasi	<i>Loss</i>
MobileNetV2	92,54%	0,1280
DenseNet121	88,06%	0,3691

Pada tabel 4.2 MobileNetV2 unggul dalam akurasi (92.54%) dan memiliki *loss* yang jauh lebih kecil (0.1280) menjadikannya model yang lebih ringan dan cepat dalam pelatihan, dibandingkan DenseNet121 (88.06% akurasi dan 0.3691 *loss*) hal tersebut tetap memberikan hasil yang baik, namun kompleksitas model ini menghasilkan kinerja yang sedikit lebih rendah dalam hal akurasi dan lebih tinggi pada *loss*.

3. *Classification Report*

MobileNetV2 menunjukkan kinerja yang unggul secara keseluruhan dengan mencapai akurasi sebesar 93%, *precision* dan *recall* yang tinggi serta seimbang untuk kedua kelas, menghasilkan *F1-score* yang konsisten (93 untuk *Coral-Bleached* dan 92 untuk *Coral-Unbleached*). Model ini menunjukkan keseimbangan dan kemampuan generalisasi yang sangat baik. Sedangkan DenseNet121 memiliki akurasi 88%, dengan keunggulan *precision* tinggi pada kelas *Coral-Bleached* (1.00) dan *recall* tinggi pada kelas *Coral-Unbleached* (1.00). Namun, model ini kurang seimbang dibandingkan MobileNetV2, dengan *F1-score* yang sedikit lebih rendah (~0.88-0.89).

4. *Confusion Matrix*

MobileNetV2 memiliki keunggulan dalam memprediksi kedua kelas secara konsisten dan seimbang, dengan hanya 4 kesalahan pada kelas *Coral-Bleached* dan 1 kesalahan pada kelas *Coral-Unbleached*, sehingga total kesalahan model ini sangat rendah. Sementara Densenet121 memiliki performa sangat baik dengan tidak membuat kesalahan pada kelas *Coral-Unbleached*, tetapi memiliki 8 kesalahan pada kelas *Coral-Bleached*, menunjukkan adanya bias model terhadap salah satu kelas.

4.2.2 Perbandingan MobileNetV2 dan DenseNet121

Pada perbandingan model MobileNetV2 dan DenseNet121 yang pertama menggunakan beberapa parameter pengujian, antarlain *Validation Accuracy*, *Validation Loss*, *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Perbandingan dari pengujian tersebut dapat dilihat pada Tabel 4.7.

Tabel 4. 7 Perbandingan Performa MobileNetV2 dan DenseNet121

Perbandingan Performa						
Metode	<i>Validation Accuracy</i>	<i>Validation Loss</i>	<i>Accuracy</i>	<i>Presicion</i>	<i>Recall</i>	<i>F1-Score</i>
MobileNetV2	0.925	0.128	0.93	0.93	0.93	0.93
DenseNet121	0.880	0.369	0.88	0.90	0.89	0.88

Berdasarkan perbandingan dari model MobileNetV2 dan DenseNet121 menunjukkan bahwa MobileNetV2 lebih unggul secara keseluruhan pada parameter *Validation Accuracy* sebesar 0.925 yang menunjukkan Tingkat akurasi tinggi pada data validasi, *Validation Loss* 0.128 yang menunjukkan kesalahan prediksi yang rendah, *Accuracy* sebesar 0.93, *Precision* sebesar 0.93, *Recall* sebesar 0.93, dan *F1-Score* sebesar 0.93 yang menunjukkan model MobileNetV2 lebih akurat dalam memprediksi citra terumbu karang. Sedangkan pada model DenseNet121 menghasilkan *Validation Accuracy* sebesar 0.880 yang lebih rendah dibandingkan MobileNetV2, *Validation Loss* sebesar 0.369 menunjukkan kesalahan prediksi yang lebih tinggi, *Accuracy* sebesar 0.88, *Precision* sebesar 0.90, *Recall* sebesar 0.89, dan *F1-Score* sebesar 0.88. Hal tersebut menunjukkan bahwa MobileNetV2 lebih akurat, efisien, dan stabil dalam mengklasifikasikan citra terumbu Karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*).

Pada perbandingan MobileNetV2 dan DenseNet121 yang kedua menggunakan parameter rata-rata waktu pelatihan dalam satuan menit, jumlah *epoch*, dan penggunaan memori untuk mengetahui perbandingan efisiensi dari model tersebut.

Tabel 4. 8 Perbandingan Efisiensi MobileNetV2 dan DenseNet121

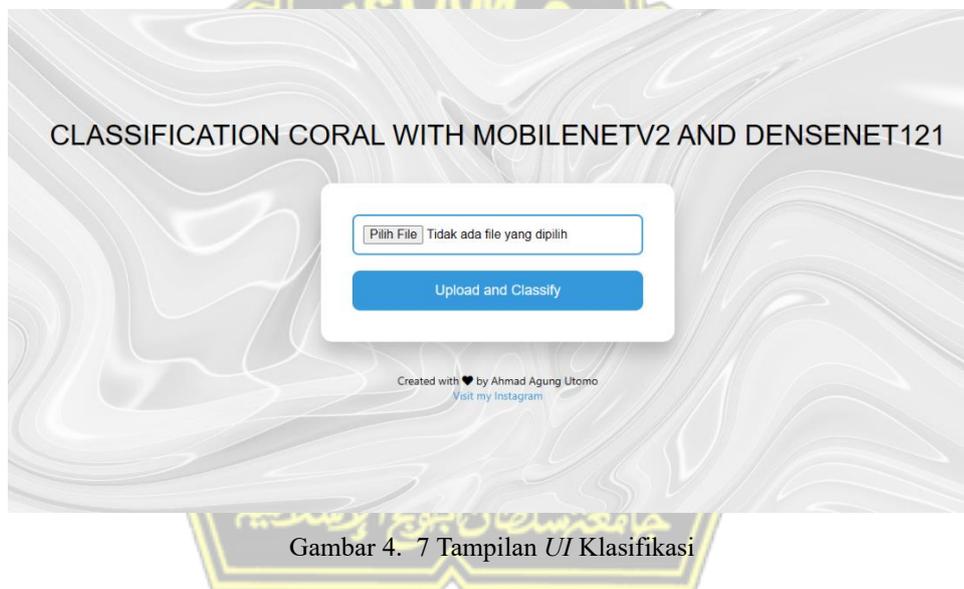
Perbandingan Efisiensi			
Metode	Waktu Pelatihan	Epoch	Memori
MobileNetV2	6 Menit	32	10,7 MB
DenseNet121	18 Menit	29	29 MB

Berdasarkan Tabel 4.8 menunjukkan perbedaan kinerja antara arsitektur MobileNetV2 dan DenseNet121. Waktu pelatihan pada model MobileNetV2 memerlukan waktu 6 menit untuk menyelesaikan pelatihan, sedangkan DenseNet121 membutuhkan waktu 18 menit, yang berarti 3 kali lebih lama dibandingkan MobileNetV2 yang lebih efisien dari segi waktu. Pada MobileNetV2, pelatihan berlangsung selama 32 *epoch* dan DenseNet121 selesai dalam 29 *epoch*. Meskipun DenseNet121 memerlukan lebih sedikit epoch, waktu per *epoch*-nya lebih lama, sehingga total waktu pelatihannya tetap lebih tinggi. Memori yang digunakan pada MobileNetV2 sebesar 10,7 MB sedangkan DenseNet121 sebesar

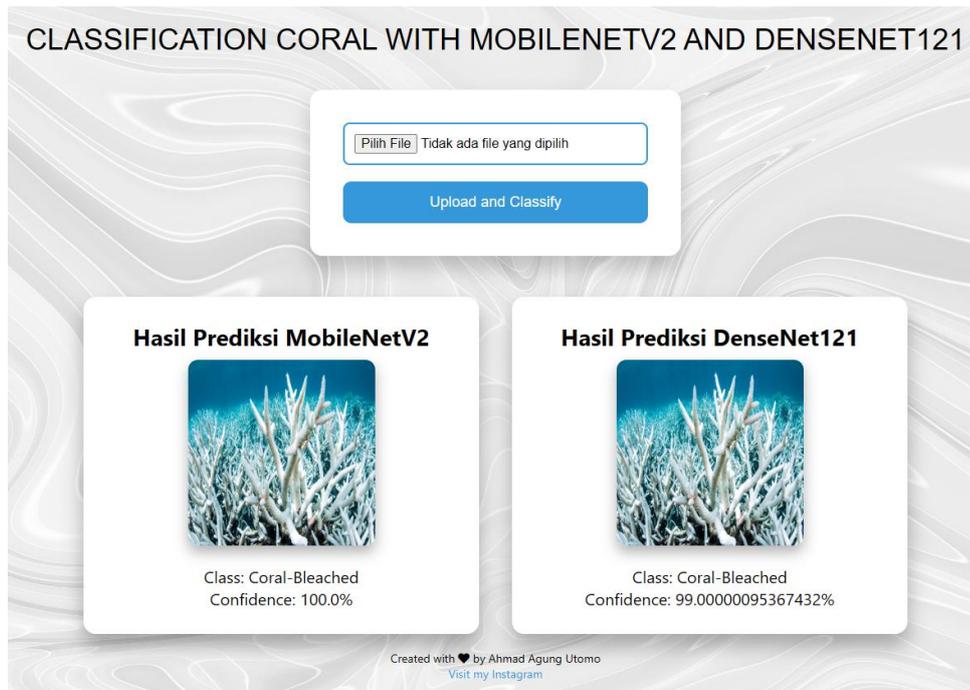
29 MB yang menunjukkan MobileNetV2 lebih ringan untuk perangkat dengan keterbatasan sumber daya.

4.2.3 Hasil Implementasi Model

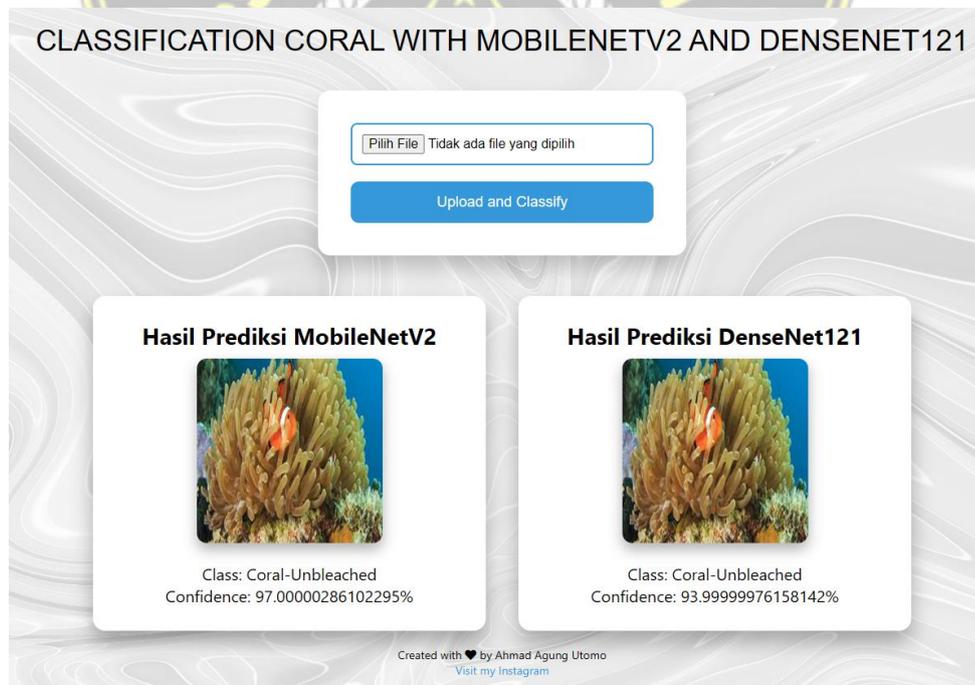
Setelah mendapatkan hasil pengujian dan perbandingan evaluasi model MobileNetV2 dan DenseNet121, kedua model diimplementasikan dalam bentuk *website* agar mudah digunakan dalam proses klasifikasi terumbu karang yang mengalami pemutihan (*bleached*) dan yang tidak (*unbleached*). Implementasi kedua model ini menjadi langkah final untuk memastikan bahwa teori dan eksperimen yang dilakukan selama proses pelatihan dan evaluasi dapat diterapkan dalam konteks praktis.



Gambar 4. 7 Tampilan UI Klasifikasi



Gambar 4. 8 Tampilan Hasil Klasifikasi



Gambar 4. 9 Tampilan Hasil Klasifikasi

Pada gambar 4.7, 4.8, dan 4.9 merupakan tampilan *UI* utama, terdapat kolom unggah *file* atau pilih *file* untuk mengunggah citra terumbu karang yang mengalami pemutihan (*bleached*) atau tidak (*unbleached*) yang ingin diklasifikasikan. *File*

citra yang dipilih akan dikirimkan ke *server* untuk diproses lebih lanjut. Tujuannya untuk memberikan *input* data yang akan diklasifikasikan oleh model.

Setelah citra diterima oleh *backend*, citra tersebut akan diubah ukurannya agar sesuai dengan input model MobileNetV2 dan DenseNet121 yaitu 224x224 piksel. Kemudian nilai piksel citra dinormalisasi pada rentang 0, 1 untuk memastikan konsistensi *input* dengan data yang digunakan selama pelatihan model. Tujuannya untuk menyesuaikan format gambar dengan standar *input* model sehingga prediksi dapat dilakukan dengan akurat.

Citra yang telah diproses akan dimasukkan ke dalam model MobileNetV2 dan DenseNet121. Model akan melakukan ekstraksi fitur melalui lapisan *convolutional* dan mengidentifikasi pola visual yang relevan untuk menentukan apakah terumbu karang termasuk kategori mengalami pemutihan (*bleached*) atau tidak (*unbleached*). Tujuannya untuk Menghasilkan prediksi klasifikasi berdasarkan data gambar yang diberikan.

Hasil klasifikasi berupa label "*Coral-Bleached*" atau "*Coral-Unbleached*" dan tingkat kepercayaan model terhadap prediksi tersebut, dinyatakan dalam bentuk persentase (contoh: 99%).

Tabel 4. 9 Perbandingan Percobaan Klasifikasi MobileNetV2 dan DenseNet121

No. Percobaan	Hasil Prediksi Terumbu Karang		Kepercayaan(%)		Asli
	MobileNet V2	DenseNet 121	MobileNet V2	DenseNet 121	
Percobaan-1	<i>Unbleached</i>	<i>Unbleached</i>	95%	84%	<i>Unbleached</i>
Percobaan-2	<i>Bleached</i>	<i>Bleached</i>	100%	99%	<i>Bleached</i>
Percobaan-3	<i>Unbleached</i>	<i>Unbleached</i>	100%	99%	<i>Unbleached</i>
Percobaan-4	<i>Unbleached</i>	<i>Bleached</i>	93%	70%	<i>Unbleached</i>
Percobaan-5	<i>Unbleached</i>	<i>Bleached</i>	92%	54%	<i>Unbleached</i>
Percobaan-6	<i>Unbleached</i>	<i>Unbleached</i>	97%	93%	<i>Unbleached</i>
Percobaan-7	<i>Bleached</i>	<i>Bleached</i>	100%	100%	<i>Bleached</i>
Percobaan-8	<i>Bleached</i>	<i>Bleached</i>	88%	75%	<i>Bleached</i>
Percobaan-9	<i>Bleached</i>	<i>Bleached</i>	100%	82%	<i>Bleached</i>
Percobaan-10	<i>Bleached</i>	<i>Bleached</i>	99%	91%	<i>Bleached</i>

Dari tabel 4.9 diperoleh hasil dari 10 kali percobaan klasifikasi terumbu karang dengan menggunakan gambar yang sama, di mana setiap kelas (*bleached* dan *unbleached*) menggunakan data sebanyak 5 gambar. Berdasarkan hasil percobaan,

arsitektur MobileNetV2 berhasil mengklasifikasikan seluruh data dengan benar, yaitu 10 dari 10 percobaan, dengan tingkat kepercayaan berkisar antara 88% hingga 100%.

Sementara itu, arsitektur DenseNet121 berhasil mengklasifikasikan data dengan benar pada 8 dari 10 percobaan, dengan tingkat kepercayaan antara 54% hingga 100%. Kesalahan prediksi pada DenseNet121 terjadi pada Percobaan-4 dan Percobaan-5, di mana model memprediksi data *unbleached* sebagai *bleached*. Hasil ini menunjukkan bahwa MobileNetV2 memiliki performa yang lebih akurat dan stabil dalam mengklasifikasikan terumbu karang jika dibandingkan dengan DenseNet121 pada percobaan ini.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari penelitian yang berjudul “Perbandingan MobileNetV2 Dan DenseNet121 Pada Klasifikasi Citra Terumbu Karang Menggunakan *Convolutional Neural Network*”. Pada penelitian ini MobileNetV2 menunjukkan performa yang lebih baik dan stabil serta konsisten dibandingkan DenseNet121. MobileNetV2 mencapai akurasi 92,54% dengan *loss* 0,1280, sedangkan DenseNet121 mencapai akurasi 88,06% dengan *loss* 0,3691. MobileNetV2 lebih seimbang dengan nilai tinggi untuk kedua kelas, sementara DenseNet121 menunjukkan ketidakseimbangan performa. MobileNetV2 juga lebih efisien dengan waktu pelatihan yang lebih cepat dan penggunaan memori yang lebih rendah dibandingkan DenseNet121. Implementasi model MobileNetV2 pada data nyata membuktikan hasil yang konsisten dengan hasil evaluasi, menjadikannya solusi yang efektif dan dapat diandalkan untuk klasifikasi terumbu karang yang mengalami pemutihan (*bleached*) dan tidak (*unbleached*).

5.2 Saran

Penelitian mendatang dapat memanfaatkan *dataset* yang lebih luas dan bervariasi dapat digunakan untuk meningkatkan kemampuan generalisasi model pada kondisi lingkungan nyata yang kompleks. Selain itu, efisiensi tinggi MobileNetV2 membuka peluang penerapannya pada perangkat *edge*, seperti *drone* atau perangkat bawah laut. Evaluasi jangka panjang juga penting untuk menguji kinerja model terhadap perubahan kondisi lingkungan dan citra yang dinamis.

DAFTAR PUSTAKA

- ANHAR, A., & PUTRA, R. A. (2023). Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN). *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11(2), 466. <https://doi.org/10.26760/elkomika.v11i2.466>
- Hadi, H. P., Rachmawanto, E. H., & Ali, R. R. (2024). Comparison of DenseNet-121 and MobileNet for Coral Reef Classification. *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 23(2), 333–342. <https://doi.org/10.30812/matrik.v23i2.3683>
- Hadi, H. P., Rachmawanto, E. H., & Sari, C. A. (2024). Klasifikasi Terumbu Karang Menggunakan Cnn Mobilenet. *Semnas Ristek (Seminar Nasional Riset dan Inovasi Teknologi)*, 8(01), 326–332. <https://doi.org/10.30998/semnasristek.v8i01.7177>
- Hatur, Y., & Sabri, A. (2024). Perbandingan Arsitektur MobileNetV2 dan DenseNet121 untuk Klasifikasi. *Jurnal Ilmiah Komputasi*, 23(1), 67–74. <https://doi.org/10.32409/jikstik.23.1.3502>
- I Wayan Suartika E. P, A. Y. W. dan R. S. (2016). JURNAL TEKNIK ITS Vol. 5, No. 1, (2016) ISSN: 2337-3539 (2301-9271 Print). *Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101, Vol. 5, No(1), 1–2.*
- Jamil, S., Rahman, M., & Haider, A. (2021). Bag of features (Bof) based deep learning framework for bleached corals detection. *Big Data and Cognitive Computing*, 5(4). <https://doi.org/10.3390/bdcc5040053>
- Khultsum, U., & Taufik, G. (2023). Komparasi Kinerja DenseNet 121 dan MobileNet untuk Klasifikasi Citra Penyakit Daun Kentang. *JURIKOM (Jurnal Riset Komputer)*, 10(2), 558. <https://doi.org/10.30865/jurikom.v10i2.6047>
- Listyalina, L., Buyung, I., Munir, A. Q., Mustiadi, I., & Dharmawan, D. A. (2022). Conv-Tire: Tire Feasibility Assessment using Convolutional Neural Networks
Conv-Tire: Asesmen Kelayakan Ban berbasis Convolutional Neural Network.

- Jurnal Informatika dan Teknologi Informasi*, 19(3), 323–336.
<https://doi.org/10.31515/telematika.v19i3.7697>
- Permatasari, A., Yustisia, D., Alamsyah, R., & Fauzi, I. (2023). Kondisi Terumbu Karang Di Perairan Pulau Batanglampe Kabupaten Sinjai. *Sebatik*, 27(2), 651–656. <https://doi.org/10.46984/sebatik.v27i2.2400>
- Saputra, A. D., Hindarto, D., & Santoso, H. (2023). Disease Classification on Rice Leaves using DenseNet121, DenseNet169, DenseNet201. *Sinkron*, 8(1), 48–55. <https://doi.org/10.33395/sinkron.v8i1.11906>
- Sulfa, A. M. F., Zahirah, N. R. P., & Assidiq, F. M. (2024). Dampak Coral Bleaching Pada Masyarakat Lokal Di Kawasan Taman Laut Bunaken. *Riset Sains dan Teknologi Kelautan*, 7(1), 22–27. <https://doi.org/10.62012/sensistek.v7i1.31626>
- Sully, S., Burkepile, D. E., Donovan, M. K., Hodgson, G., & van Woesik, R. (2019). A global analysis of coral bleaching over the past two decades. *Nature Communications*, 10(1), 1–5. <https://doi.org/10.1038/s41467-019-09238-2>
- Thamarai, M., & Aruna, S. P. (2023). Stressed Coral Reef Identification Using Deep Learning CNN Techniques. *Journal of Electronic & Information Systems*, 5(2), 1–9. <https://doi.org/10.30564/jeis.v5i2.5808>
- Vellaichamy, A. S., Swaminathan, A., Varun, C., & S, K. (2021). Multiple Plant Leaf Disease Classification Using Densenet-121 Architecture. *International Journal of Electrical Engineering and Technology*, 12(5), 38–57. <https://doi.org/10.34218/ijeet.12.5.2021.005>
- Wang, S., Chen, N. L., Song, Y. D., Wang, T. T., Wen, J., Guo, T. Q., Zhang, H. J., Mo, L., Ma, H. R., & Xiang, L. (2024). ML-Net: A Multi-Local Perception Network for Healthy and Bleached Coral Image Classification. *Journal of Marine Science and Engineering*, 12(8). <https://doi.org/10.3390/jmse12081266>
- Wang, W., Li, Y., Zou, T., Wang, X., You, J., & Luo, Y. (2020). A novel image classification approach via dense-mobilenet models. *Mobile Information Systems*, 2020. <https://doi.org/10.1155/2020/7602384>
- Zhou, D., Hou, Q., Chen, Y., Feng, J., & Yan, S. (2020). Rethinking Bottleneck

Structure for Efficient Mobile Network Design. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12348 LNCS, 680–697.
https://doi.org/10.1007/978-3-030-58580-8_40

