

# Penerapan CNN untuk Identifikasi Pemutihan pada Terumbu Karang

Alex Sander (13521061)

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail (gmail): maximun154@gmail.com

**Abstrak**— Pemutihan terumbu karang merupakan masalah lingkungan yang signifikan yang memengaruhi ekosistem laut secara global. Identifikasi pemutihan terumbu karang secara cepat dan akurat sangat penting untuk upaya pelestarian dan mitigasi dampaknya. Makalah ini membahas penerapan Convolutional Neural Network (CNN) dalam proses identifikasi pemutihan pada terumbu karang. Metode ini menggunakan data citra terumbu karang sebagai input, memanfaatkan kemampuan CNN untuk mengekstraksi fitur penting melalui serangkaian lapisan konvolusi, dan menghasilkan model yang mampu mendeteksi kondisi pemutihan dengan tingkat akurasi yang tinggi. Eksperimen dilakukan menggunakan dataset citra terumbu karang yang beragam, serta dilakukan evaluasi terhadap berbagai arsitektur CNN untuk menilai performa dan kehandalan sistem dalam mendeteksi pemutihan.

**Keywords**—CNN, Pemutihan Terumbu Karang, Citra, Identifikasi, Ekosistem Laut

## I. PENDAHULUAN

Identifikasi dan pemantauan pemutihan terumbu karang merupakan aspek penting dalam upaya pelestarian ekosistem laut. Pemutihan terumbu karang, yang umumnya disebabkan oleh perubahan suhu laut dan tekanan lingkungan lainnya, dapat mengancam kelangsungan hidup terumbu karang sebagai habitat laut yang kaya keanekaragaman hayati. Pendekatan manual untuk mengidentifikasi pemutihan terumbu karang membutuhkan keahlian tinggi, waktu, dan sumber daya yang signifikan, sehingga sulit untuk diterapkan pada skala besar.

Dalam beberapa tahun terakhir, teknologi Deep Learning, terutama Convolutional Neural Network (CNN), telah menunjukkan potensi besar dalam analisis data citra. CNN dikenal mampu secara efektif mengekstraksi dan mengenali pola-pola visual yang kompleks, menjadikannya alat yang menjanjikan dalam mengotomatiskan proses identifikasi pemutihan terumbu karang. Kemajuan dalam pengolahan citra berbasis CNN telah membuka peluang baru untuk mengembangkan sistem pemantauan terumbu karang yang lebih cepat dan efisien.

Makalah ini bertujuan untuk menyelidiki penerapan CNN dalam mendeteksi pemutihan pada terumbu karang berdasarkan data citra. Penelitian ini akan mengeksplorasi kemampuan CNN dalam mengekstraksi fitur-fitur visual penting dari gambar terumbu karang dan mengklasifikasikan kondisi pemutihan dengan tingkat akurasi yang tinggi. Selain itu, analisis terhadap berbagai arsitektur CNN akan dilakukan untuk mengevaluasi keefektifan dan keandalan pendekatan ini.

Diharapkan, penelitian ini dapat memberikan kontribusi signifikan dalam memperluas penerapan teknologi Deep Learning untuk konservasi lingkungan, khususnya dalam pemantauan ekosistem laut. Selain itu, hasil penelitian ini diharapkan dapat memberikan wawasan yang berguna bagi pengembangan sistem berbasis teknologi untuk mendukung pelestarian terumbu karang secara berkelanjutan.

## II. LANDASAN TEORI

### A. Terumbu Karang

Terumbu karang adalah ekosistem laut yang terbentuk dari kerangka kalsium karbonat yang dihasilkan oleh karang, organisme laut yang hidup berkoloni. Ekosistem ini mendukung keanekaragaman hayati yang tinggi, menjadikannya salah satu ekosistem paling produktif di dunia.



Gambar 2.1. Terumbu Karang

Sumber: [https://id.wikipedia.org/wiki/Terumbu\\_karang](https://id.wikipedia.org/wiki/Terumbu_karang)

Terumbu karang memiliki peran penting dalam ekosistem laut, antara lain:

1. Habitat: Menjadi tempat tinggal dan perlindungan bagi berbagai spesies laut.
2. Sumber Pangan: Menyediakan makanan bagi ikan dan organisme laut lainnya.
3. Pelindung Pantai: Mengurangi dampak gelombang dan mencegah erosi pantai.

Terumbu karang sangat rentan terhadap perubahan lingkungan. Faktor-faktor yang mengancam kelestariannya meliputi:

1. Kenaikan Suhu Laut: Menyebabkan stres pada karang dan memicu pemutihan (coral bleaching).
2. Polusi: Masuknya bahan kimia dan limbah ke ekosistem laut.
3. Kenaikan Asidifikasi Laut: Perubahan pH air laut akibat peningkatan karbon dioksida di atmosfer.



Gambar 2.2. Perbandingan Terumbu Karang yang sehat (kiri) dengan Terumbu Karang yang mengalami pemutihan (kanan)

Sumber: <https://www.tempo.co/sains/terumbu-karang-dunia-memutih-912602>

Pemutihan terumbu karang terjadi ketika karang kehilangan alga simbiotik (*zooxanthellae*) akibat stres lingkungan, seperti suhu yang terlalu tinggi. *Zooxanthellae* memberikan warna dan nutrisi bagi karang, sehingga kehilangan alga ini menyebabkan karang berubah menjadi putih dan rentan terhadap kematian. Pemutihan terumbu karang menjadi indikator utama kerusakan ekosistem laut.

### B. Citra

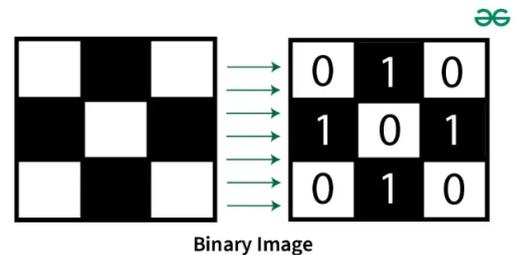
Citra digital, atau yang dikenal sebagai representasi gambar dalam format digital, adalah array dua dimensi (2D) yang terdiri atas angka-angka (piksel) yang disusun dalam baris dan kolom. Secara matematis, citra digital dapat dinyatakan melalui fungsi  $f(x,y)$  di mana  $x$  dan  $y$  merepresentasikan koordinat horizontal dan vertikal, sementara  $f$  menunjukkan intensitas bayangan pada titik tersebut.

Setiap citra terdiri atas elemen-elemen terbatas yang dikenal sebagai elemen gambar atau piksel. Piksel ini merupakan unit terkecil dalam citra, yang masing-masing

memiliki nilai tertentu, seperti intensitas warna, bayangan, atau opasitas. Piksel berfungsi sebagai penyusun citra digital, memungkinkan representasi visual yang kompleks diubah menjadi data numerik yang dapat diproses oleh komputer.

Berikut merupakan jenis citra berdasarkan warna yang tersedia pada citra tersebut.

#### 1. Citra Biner



Gambar 2.3. Konsep Citra Biner

Citra biner terdiri atas piksel-piksel yang hanya dapat berupa dua nilai, yaitu 0 untuk merepresentasikan warna hitam, dan 1 untuk merepresentasikan warna putih. Citra biner merupakan bentuk paling sederhana dari suatu citra yang dapat mengandung sebuah informasi.

#### 2. Citra Grayscale



Gambar 2.4. Contoh Citra Grayscale

Citra *grayscale* atau citra abu terdiri atas piksel-piksel yang memiliki nilai dalam rentang 0 hingga  $(L-1)$ , di mana  $L$  adalah tingkat intensitas maksimum yang dapat direpresentasikan.

Citra *grayscale* dapat menyimpan lebih banyak informasi dibandingkan citra biner karena rentang penyimpanan informasi dalam satu piksel lebih besar. Ukuran suatu citra abu juga lebih besar dibandingkan citra biner karena masing-masing piksel harus menyimpan  $\log_2(L)$  bit untuk menyimpan nilai tingkat keabuan piksel tersebut.

#### 3. Citra Berwarna



Gambar 2.5. Contoh Citra Berwarna

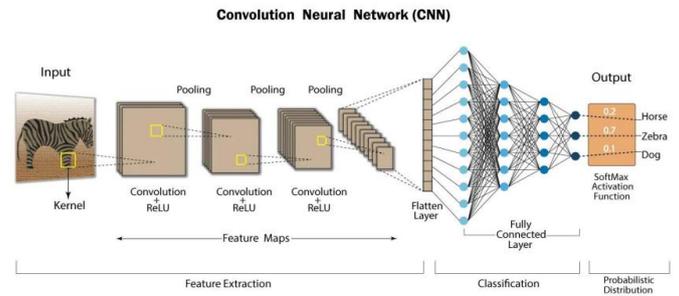
Citra berwarna adalah jenis citra digital yang terdiri atas piksel-piksel yang merepresentasikan informasi warna melalui kombinasi beberapa kanal warna, seperti merah (Red), hijau (Green), dan biru (Blue), yang sering dikenal sebagai model warna RGB. Setiap piksel dalam citra berwarna memiliki nilai intensitas untuk masing-masing kanal warna, dengan rentang nilai biasanya dari 0 hingga  $(L-1)$ , di mana  $L$  adalah tingkat intensitas maksimum yang dapat direpresentasikan.

Citra berwarna dapat menyimpan informasi yang jauh lebih kaya dibandingkan citra grayscale karena setiap piksel mencakup tiga komponen warna, yang secara bersamaan memberikan spektrum warna yang lebih luas. Namun, ukuran citra berwarna juga lebih besar dibandingkan citra grayscale karena masing-masing piksel membutuhkan  $3 \times \log_2(L)$  bit untuk menyimpan nilai intensitas pada ketiga kanal warna tersebut. Hal ini menjadikan citra berwarna lebih kompleks dalam hal penyimpanan dan pengolahan, tetapi memungkinkan representasi visual yang lebih mendetail dan realistis.

### C. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN atau ConvNet) adalah salah satu algoritma deep learning yang sangat populer, khususnya dalam pengolahan data dengan struktur topologi berbentuk grid, seperti citra digital. CNN dirancang sebagai arsitektur jaringan yang mampu mempelajari pola langsung dari data mentah, sehingga tidak memerlukan proses ekstraksi fitur secara manual.

CNN dapat dipahami sebagai bentuk jaringan saraf tiruan (Artificial Neural Network/ANN) yang dilengkapi dengan operasi konvolusi, menjadikannya kombinasi unik antara ANN dan teknik konvolusi untuk mengolah dan mengenali pola dalam data visual secara efisien.



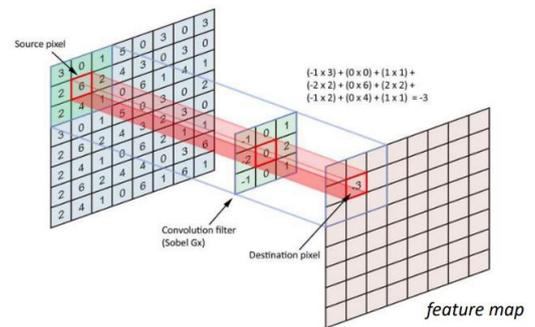
Gambar 2.6. Arsitektur Convolutional Neural Network

Sumber: : <https://developersbreach.com/convolutional-neural-network-deep-learning/>

Pada umumnya, CNN terdiri atas tiga jenis lapisan utama, yaitu:

#### 1. Convolutional Layer

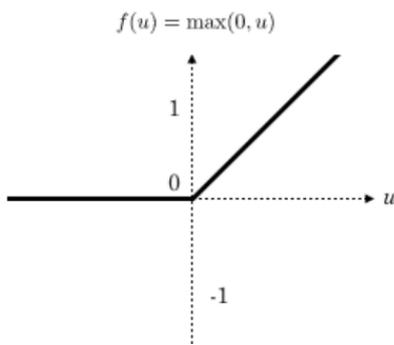
Convolutional Layer atau Lapisan Konvolusi melakukan operasi konvolusi pada citra masukan dengan sejumlah penapis (*filter*). Tiap penapis menghasilkan luaran yang disebut feature map.



Gambar 2.7. Convolutional Layer

Sumber: Silde Materi Kuliah, Rinaldi (2024)

Salah satu lapisan tambahan yang umum digunakan pada setelah lapisan konvolusi untuk CNN adalah Rectified Linear Unit atau biasa disebut sebagai ReLU. ReLU adalah layer tambahan yang memungkinkan pelatihan yang lebih cepat dan efektif dengan memetakan nilai negatif ke nol dan mempertahankan nilai positif. Pada dasarnya ReLU adalah operasi perpixel dengan cara mengganti nilai negatif pixel di dalam feature map menjadi nol.



Gambar 2.8. Grafik ReLU

Sumber: Silde Materi Kuliah, Rinaldi (2024)

## 2. Pooling Layer

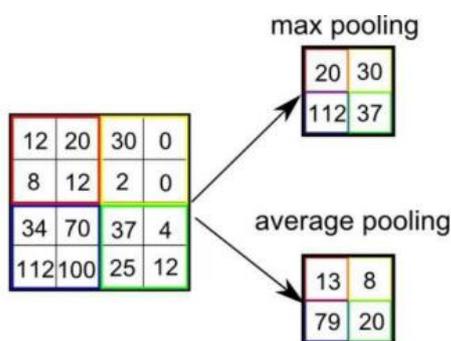
Mirip dengan Convolutional Layer, Pooling Layer bertanggung jawab untuk mengurangi ukuran spasial dari matriks fitur hasil konvolusi. Hal ini bertujuan untuk mengurangi daya komputasi yang diperlukan untuk memproses data melalui pengurangan dimensi. Berikut merupakan dua jenis pooling dasar yang umum digunakan, yaitu:

### a. Max Pooling

Max Pooling akan mengembalikan nilai maksimum dari bagian gambar yang dicakup oleh kernel.

### b. Average Pooling

Average Pooling akan mengembalikan rata-rata semua nilai dari bagian gambar yang dicakup oleh Kernel.



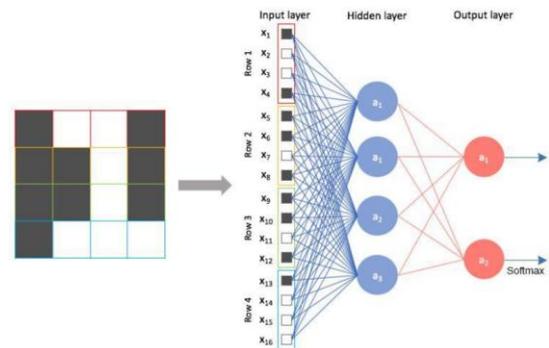
Gambar 2.9. Pooling Layer

Sumber: Silde Materi Kuliah, Rinaldi (2024)

## 3. Fully Connected Layer

Setelah deteksi fitur, arsitektur CNN beralih ke klasifikasi. Lapisan terakhir di dalam CNN adalah *Fully Connected Layer* (FC) yang menghasilkan vektor dimensi K, dalam hal ini

K adalah jumlah kelas yang dapat diprediksi oleh jaringan. Vektor ini berisi probabilitas untuk setiap kelas dari setiap gambar yang diklasifikasikan. Lapisan terakhir dari arsitektur CNN menggunakan fungsi softmax untuk menyediakan luaran klasifikasi.

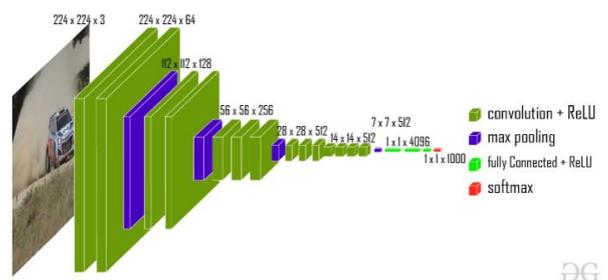


Gambar 2.10. Fully Connected Layer

Sumber: Silde Materi Kuliah, Rinaldi (2024)

## D. VGG-16

VGG-16 adalah arsitektur Convolutional Neural Network (CNN) yang dikembangkan oleh tim Visual Geometry Group di University of Oxford. Arsitektur ini terdiri dari 16 lapisan dengan bobot yang dapat dilatih, termasuk 13 lapisan konvolusi dan 3 lapisan fully connected. Ciri khas utama dari VGG-16 adalah penggunaan kernel konvolusi kecil berukuran 3x3 dengan stride dan padding yang menjaga dimensi spasial, serta penggunaan lapisan pooling maksimal (max pooling) berukuran 2x2 untuk mengurangi dimensi data secara bertahap.



Gambar 2.10. Arsitektur VGG-16

Sumber: <https://www.geeksforgeeks.org/vgg-16-cnn-model/>

Pendekatan desainnya yang sederhana namun efektif membuat VGG-16 menjadi model yang sangat populer untuk berbagai tugas pengolahan citra, seperti klasifikasi gambar, deteksi objek, dan segmentasi citra. Model ini dirancang dengan fokus pada kedalaman jaringan, di mana setiap penambahan lapisan bertujuan untuk meningkatkan kemampuan jaringan dalam menangkap fitur yang lebih kompleks.

VGG-16 dikenal memiliki performa tinggi pada dataset seperti ImageNet, yang mencakup jutaan gambar dalam ribuan kategori. Meski begitu, model ini memiliki kelemahan berupa ukuran file model yang besar dan kebutuhan komputasi yang tinggi, sehingga memerlukan perangkat keras dengan performa yang kuat untuk pelatihan dan inferensi.

Dalam aplikasi praktis, VGG-16 sering digunakan sebagai model dasar untuk transfer learning, di mana bobot yang telah dilatih pada dataset besar digunakan kembali untuk tugas-tugas spesifik dengan dataset yang lebih kecil. Arsitektur ini juga menjadi inspirasi bagi banyak model CNN yang lebih modern.

### III. PEMBAHASAN

Pada Bab ini akan dijelaskan mengenai proses pembentukan solusi, mulai dari pengumpulan data hingga pengujian data.

#### A. Pengumpulan Data

Data yang digunakan dalam penelitian ini didapat secara publik yang berjudul Healthy and Bleached Corals Image Classification. Dataset ini berisi 923 gambar. Gambar-gambar tersebut dibagi menjadi dua, yaitu healthy dan bleached. Untuk setiap kelas ada sekitar 400 gambar. Gambar tidak beresolusi tinggi, yaitu maksimal 300 piksel untuk panjang dan lebar gambar.

#### B. Preprocessing Data

Pada bagian ini, pre-processing data dilakukan untuk mengoptimalkan proses maupun hasil prediksi. Sebelum adanya perubahan pada gambar, dataset dibagi menjadi dua, yaitu dataset training dan dataset testing, serta isi dataset dilakukan shuffling. Kemudian, dilakukan resizing gambar menjadi 224x224. Proses resizing ini diimplementasikan dengan bantuan fungsi `preprocess_input` dari library `vgg16` yang disediakan oleh `tensorflow`. Berikut ini adalah potongan kode yang dimaksud.

```
#converting to gray and resizing image
picture=image.load_img(training_Dirs[i],target_size=(pic_size,pic_size))
#Converting the grayscale image to numpy vector, i believe
picture_array=image.img_to_array(picture)
#preprocessing image for model
picture_array_Train[i,:,:]=preprocess_input(picture_array)
```

#### C. Pengimplementasian CNN

Model CNN yang digunakan dalam penelitian ini dibuat dengan menggunakan kelas `VGG16` yang disediakan library `tensorflow`. Berikut ini adalah rincian layer-layer dari model CNN yang dibuat.

```
#VGG16 pre-trained
base_model= VGG16(weights='imagenet', include_top=False,input_shape=(224,224,3))
base_model.trainable=False
#New top layers
inputs=tf.keras.layers.Input((224,224,3))
x=base_model(inputs,training=False)
x= tf.keras.Layers.Flatten()(x)

x=tf.keras.layers.Dense(5,activation='relu')(x)
x=tf.keras.layers.Dense(2,activation='relu')(x)
outputs=tf.keras.layers.Dense(1,activation='sigmoid')(x)
extra_layer=tf.keras.Model(inputs,outputs)

extra_layer.compile(optimizer='RMSprop',loss='mean_absolute_error', metrics=['accuracy'])
```

Berikut merupakan summary dari model CNN yang diimplementasikan pada kode di atas.

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
vgg16 (Functional)	(None, 7, 7, 512)	14,714,688
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 5)	125,445
dense_1 (Dense)	(None, 2)	12
dense_2 (Dense)	(None, 1)	1

Total params: 14,840,148 (56.61 MB)  
 Trainable params: 125,468 (490.88 KB)  
 Non-trainable params: 14,714,680 (56.13 MB)

Kemudian dilakukan training pada model yang telah dibuat dengan dataset training. Proses training menggunakan fungsi `fit` yang disediakan `tensorflow` dengan nilai `validation_fit` 0.1 dan dilakukan sebanyak 50 epoch untuk menghindari overfitting.

#### D. Evaluasi Model

Berdasarkan model yang telah dibuat, sistem memiliki akurasi sebesar 94.52%. Setelah itu, evaluasi model dilakukan dengan mengukur akurasi data tes. Berdasarkan hasil pengujian, didapatkan akurasi sebesar 75%.

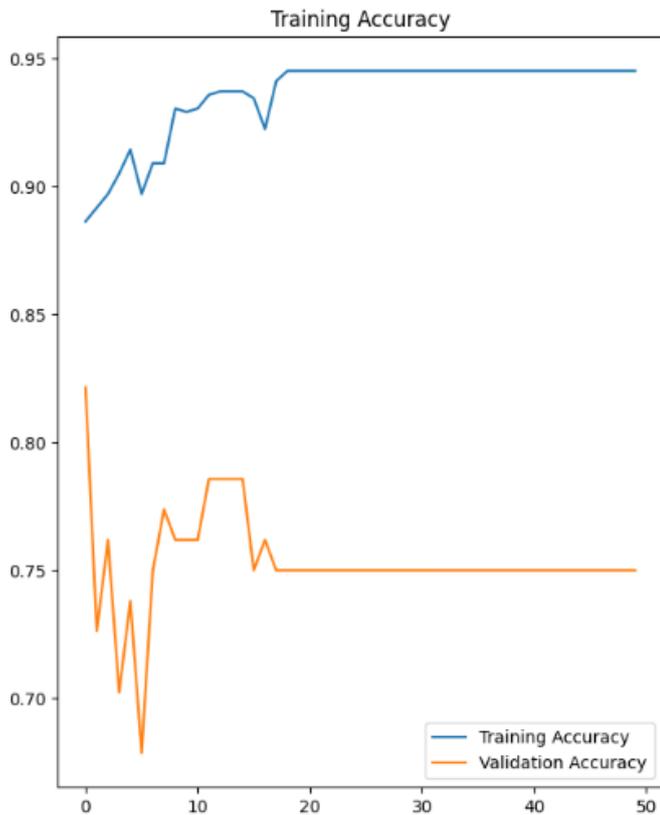
```
#Plotting accuracy and loss of the model using mean absolute error
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

epochs_range = range(50)

plt.figure(figsize=(14, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training Accuracy')

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training Loss')
plt.show()
```



#### IV. KESIMPULAN

Penggunaan CNN, khususnya VGG-16, telah menunjukkan efektivitas yang tinggi dalam mendeteksi dan mengklasifikasikan pemutihan pada terumbu karang. Dengan arsitektur yang dirancang secara optimal dan pengolahan data yang terstruktur, model ini mampu mencapai tingkat akurasi yang memuaskan, bahkan ketika dihadapkan pada dataset yang kompleks dan bervariasi. Meskipun akurasi yang diperoleh belum mencapai hasil maksimal akibat keterbatasan performa perangkat yang digunakan penulis, hal ini tidak memberikan dampak signifikan terhadap kualitas hasil. Hal tersebut terbukti dari prediksi yang dihasilkan, yang secara konsisten sesuai dengan label pada gambar input.

#### LINK VIDEO YOUTUBE

<https://youtu.be/y0j9ooKYAXQ>

#### UCAPAN TERIMA KASIH

Dengan penuh rasa syukur, penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan tugas makalah ini dengan baik. Penulis juga menyampaikan apresiasi yang sebesar-besarnya kepada Bapak Rinaldi Munir, sebagai dosen mata kuliah Interpretasi dan Pengolahan Citra, atas bimbingan dan arahan yang telah diberikan selama perkuliahan. Selain itu, penulis merasa sangat terbantu dengan materi perkuliahan, latihan soal untuk kuis dan ujian, serta berbagai sumber pembelajaran lainnya yang disediakan oleh beliau melalui website khusus. Semua upaya tersebut sangat mendukung proses pembelajaran mata kuliah ini.

#### REFERENCES

- [1] <https://www.kaggle.com/datasets/vencerlanz09/healthy-and-bleached-corals-image-classification/data> . Diakses pada 25 Desember 2024
- [2] <https://www.geeksforgeeks.org/types-of-image/> . Diakses pada 25 Desember 2024.
- [3] <https://www.geeksforgeeks.org/vgg-16-cnn-model/> . Diakses pada 25 Desember 2024.
- [4] <https://reefresilience.org/id/stressors/bleaching/> . Diakses pada 25 Desember 2024.
- [5] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2024-2025/21-CNN-2024.pdf> . Diakses pada 25 Desember 2024.
- [6] <https://www.trivusi.web.id/2022/09/image-processing.html> . Diakses pada 25 Desember 2024.

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 15 Januari 2024  
Ttd

Alex Sander 13521061