

# **LAPORAN PENGOLAHAN CITRA DIGITAL**

## **ANALISIS DOMINANSI WARNA DAN BENTUK OBJEK DARI 3 KATEGORI GAMBAR PEMANDANGAN ALAM DENGAN *IMAGE PROCESSING***



*Mencerdaskan dan  
Memartabatkan Bangsa*

### **Disusun oleh:**

Hernanda Khoiriyah Putri (1306620025)

### **Dosen Pengampu:**

Dr. Hadi Nasbey, S.Pd, M.Si

Haris Suhendar, M.Si

**PROGRAM STUDI FISIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI JAKARTA**

**2022**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>i</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan .....	2
<b>BAB II TEORI DASAR.....</b>	<b>3</b>
2.1 <i>Library OpenCV</i> .....	3
2.2 <i>Edge Detection</i> .....	3
2.2.1 <i>Sobel Edge Detection</i> .....	3
2.2.2 <i>Canny Edge Detection</i> .....	5
2.3 <i>Thresholding</i> .....	5
2.4 RGB Histogram.....	6
2.5 Konvolusi Kernel .....	6
<b>BAB III HASIL DAN PEMBAHASAN .....</b>	<b>8</b>
<b>BAB IV KESIMPULAN.....</b>	<b>15</b>
<b>DAFTAR PUSTAKA.....</b>	<b>16</b>

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Keindahan alam Indonesia dengan beraneka ragam etnik dan keunikan budaya yang dimiliki oleh bangsa Indonesia sejak dulu menjadi perhatian dan daya tarik wisatawan mancanegara maupun wisatawan nusantara. Berdasarkan pada potensi itu, Indonesia menempatkan sektor Pariwisata menjadi sektor andalan untuk pemasukan devisa negara sekaligus peningkatan perekonomian masyarakat Indonesia. Indonesia adalah negara yang sangat kaya akan keindahan alam dan beraneka ragam budaya.

Pada segi keindahan alamnya dan letaknya yang sangat strategis serta banyak perbukitan, laut dan pegunungan yang indah, hal itu menjadi daya tarik bagi wisatawan untuk datang ke negara ini. Keindahan alam ini tidak lepas dari peran pemerintah dan masyarakat sebagai agen untuk meningkatkan pengembangan dan kemajuan pada sektor pengembangan khususnya di bidang pariwisata.

Pengolahan citra (*image processing*) merupakan suatu sistem di mana proses dilakukan dengan masukan berupa citra (*image*) dan hasilnya juga berupa citra (*image*). Pada awalnya pengolahan citra ini, dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer, serta munculnya ilmu-ilmu komputasi yang memungkinkan manusia dapat mengambil informasi dari suatu citra, maka *image processing* tidak dapat dilepaskan dengan bidang *computer vision*.

Perkembangan teknologi *image processing* sekarang ini menyediakan kemungkinan manusia untuk membuat suatu sistem yang dapat mengenali suatu citra digital. Setiap pemandangan alam memiliki ciri-ciri khusus yang membedakan antara pemandangan alam satu dan yang lainnya yang disebut dengan objek alam. Ciri-ciri tersebut berupa pepohonan, lautan, dan perbukitan. Diantara ciri-ciri tersebut, lautan merupakan bentuk yang paling mudah untuk dikenali dan diamati karena bagian tersebut tidak tersembunyi

dan ukuran objek yang luas.

Untuk membangun program analisis pemandangan alam ini penulis menggunakan metode-metode *image processing* berdasarkan warna dan objek yang diantaranya yaitu; *edge detection*, *thresholding variation*, *RGB histogram*, *image color analysis*, dan *convolution kernel*.

## 1.2 Tujuan

Tujuan dari pembuatan laporan ini yaitu:

1. Menganalisis dominansi warna dari 3 kategori gambar pemandangan alam
2. Menganalisis bentuk objek dari 3 kategori gambar pemandangan alam
3. Menerapkan metode-metode *image processing*
4. Mengevaluasi hasil analisis warna dan bentuk objek dari 3 kategori gambar pemandangan alam
5. Mengukur intensitas warna dari 3 kategori gambar pemandangan alam
6. Membandingkan 3 kategori gambar pemandangan alam dengan metode-metode *image processing*

## BAB II

### TEORI DASAR

#### 2.1 *Library OpenCV*

OpenCV (*Open Source Computer Vision Library*) adalah perpustakaan komputer *open source* visi dan perangkat lunak pembelajaran. OpenCV dibangun untuk menyediakan infrastruktur umum untuk aplikasi visi komputer dan untuk mempercepat penggunaan persepsi mesin dalam produk komersial. Menjadi produk berlisensi BSD, OpenCV memudahkan bisnis untuk memanfaatkan dan memodifikasi. Kode *library* ini memiliki lebih dari 2500 algoritma yang dapat dioptimalkan, yang mencakup satu set lengkap visi komputer klasik dan *state-of-the-art* komputer visi dan mesin pembelajaran algoritma. OpenCV adalah metode yang paling cepat dan memiliki library paling lengkap untuk komputer visi.

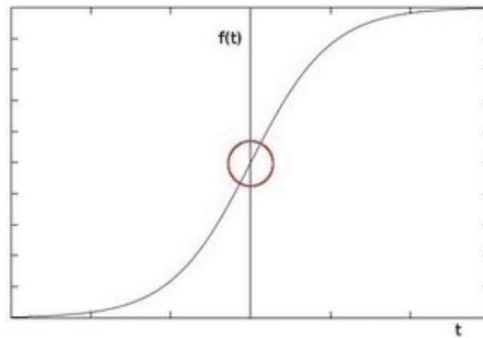
#### 2.2 *Edge Detection*

*Edge detection* (deteksi tepi) adalah teknik pemrosesan gambar yang digunakan untuk mengidentifikasi batas (tepi) objek atau wilayah di dalam gambar. Tepi adalah salah satu fitur paling penting yang terkait dengan gambar. Metode ini mengetahui struktur yang mendasari sebuah gambar melalui tepinya. Oleh karena itu, pemrosesan *computer vision* secara ekstensif menggunakan deteksi tepi dalam aplikasinya.

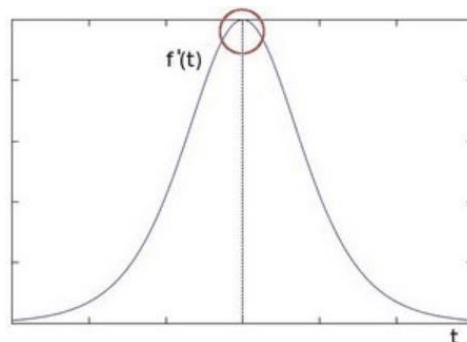
Tepi ditandai dengan perubahan intensitas piksel secara tiba-tiba. Untuk mendeteksi tepi, kita perlu mencari perubahan seperti itu di piksel lainnya. Terdapat algoritma metode deteksi tepi yang tersedia di OpenCV, yaitu *Sobel Edge Detection* dan *Canny Edge Detection*.

##### 2.2.1 *Sobel Edge Detection*

*Sobel Edge Detection* adalah salah satu algoritma yang paling banyak digunakan untuk deteksi tepi. Operator sobel mendeteksi tepi yang ditandai dengan perubahan intensitas piksel secara tiba-tiba, seperti yang ditunjukkan pada gambar di bawah ini.



Gambar 1. Intensitas piksel sebagai fungsi dari  $t$   
Kenaikan intensitas lebih nyata ketika memplot turunan pertama dari fungsi intensitas.



Gambar 2. Turunan pertama dari intensitas piksel sebagai fungsi dari  $t$

Plot di atas menunjukkan bahwa tepi dapat dideteksi di area di mana gradien lebih tinggi dari nilai ambang tertentu. Selain itu, perubahan turunan yang tiba-tiba akan mengungkapkan perubahan intensitas piksel juga. Dengan menerapkan ini, kita dapat memperkirakan turunannya menggunakan kernel  $3 \times 3$ . Dapat menggunakan satu kernel untuk mendeteksi perubahan mendadak dalam intensitas piksel ke arah X dan satu lagi ke arah Y.

Ini adalah kernel yang digunakan untuk *Sobel Edge Detection*:

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \text{X-Direction Kernel} \quad \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \text{Y-Direction Kernel}$$

Saat kernel ini digabungkan dengan gambar asli, selanjutnya akan mendapatkan *sobel edge image*.

$G_x$  dan  $G_y$  masing-masing mewakili gradien intensitas dalam arah  $x$  dan  $y$ . Jika A dan B menunjukkan kernel X dan Y yang didefinisikan di atas, maka:

$$G_x = A * I \quad \text{dan} \quad G_y = B * I$$

Di mana  $*$  menunjukkan operator konvolusi, dan  $I$  mewakili gambar masukan.

Perkiraan akhir besarnya gradien  $G$  dapat dihitung sebagai:

$$G = \sqrt{G_x^2 + G_y^2}$$

Dan orientasi gradien kemudian dapat didekati sebagai:

$$\theta = \arctan(G_y/G_x)$$

### 2.2.2 Canny Edge Detection

*Canny Edge Detection* adalah salah satu metode deteksi tepi paling populer yang digunakan saat ini karena sangat kuat dan fleksibel. Algoritmanya mengikuti proses tiga tahap untuk mengekstraksi tepi dari sebuah gambar. Terdapat 4 langkah pemrosesan yaitu:

1. *Noise reduction*
2. *Calculating intensity gradient of the image*
3. *Suppression of false edges*
4. *Hysteresis thresholding*

## 2.3 Thresholding

*Thresholding* digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Dengan menggunakan *thresholding* maka derajat keabuan dapat diubah sesuai keinginan, misalkan diinginkan menggunakan derajat keabuan 16, maka tinggal membagi nilai derajat keabuan dengan 16. Proses *thresholding* ini pada dasarnya adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan *thresholding* dengan derajat keabuan dapat digunakan rumus:

$$x = b \cdot \text{int}\left(\frac{w}{b}\right)$$

Di mana  $w$  adalah nilai derajat keabuan sebelum *thresholding* dan  $x$  adalah nilai derajat keabuan setelah *thresholding*.

*Thresholding* adalah Pelabelan pada *image gray level*. *Thresholding* memisahkan antara piksel yang memiliki nilai *gray* tinggi dengan piksel yang

memiliki nilai *gray* rendah. Piksel yang memiliki nilai *gray* tinggi diberi nilai 1, sedangkan piksel yang memiliki nilai *gray* rendah diberi nilai 0. Nilai *threshold* digunakan sebagai batasan untuk memisahkan *range* nilai yang termasuk nilai *gray* tinggi dengan *range* nilai yang termasuk nilai *gray* rendah.

Nilai *threshold* digunakan untuk memisahkan antara *background* yang gelap dengan objek yang terang, ataupun sebaliknya. Untuk bisa memisahkan antara *background* dengan objek, maka harus diketahui distribusi piksel-piksel gelap dan piksel-piksel terang. *Thresholding* adalah salah satu bentuk segmentasi citra yang paling sederhana.

## 2.4 RGB Histogram

RGB adalah singkatan dari *Red-Green-Blue*, merupakan tiga warna dasar (*primary colors*) yang secara umum dijadikan acuan warna lainnya. Dari basis RGB, kita dapat mengkonversi warna menjadi kode-kode angka yang membuat warna tersebut akan tampil universal. Komputer sudah mengemas informasi warna menjadi model warna yang sama sehingga membuat pengolahan warna RGB dapat dilakukan dengan mudah.

Histogram warna atau *color* histogram adalah representasi distribusi warna dalam sebuah gambar yang didapatkan dengan menghitung jumlah piksel dari setiap bagian *range* warna, secara tipikal dalam dua dimensi atau tiga dimensi. Dalam pembuatan histogram, nilai RGB mempunyai *range* dari 0 sampai 255 akan punya kombinasi warna sebesar 16777216 (didapatkan dari:  $255 \times 255 \times 255$ ).

## 2.5 Konvolusi Kernel

Metode konvolusi kernel merupakan metode yang digunakan untuk memfilter gambar atau citra yang mengalami derau (*noise*). *Filtering* dengan teknik konvolusi terhadap citra yang mengalami *noise* dapat mengurangi atau bahkan menghilangkan *noise* yang melekat pada citra tersebut. Konvolusi merupakan penjumlahan perkalian setiap kernel dengan setiap titik pada fungsi masukan. Kernel adalah matriks yang pada umumnya berukuran kecil dengan elemen-elemennya adalah berupa angka. Kernel dioperasikan secara



bergeser pada fungsi masukan  $f(x)$ . Jumlah hasil kali setiap titik pada fungsi tersebut merupakan hasil konvolusi yang dinyatakan dengan  $h(x)$ .

### BAB III

#### HASIL DAN PEMBAHASAN

Pada pemrosesan gambar kali ini peneliti menggunakan data gambar yang terdiri dari 3 kategori gambar pemandangan alam, yaitu pemandangan laut, pemandangan hutan, dan pemandangan taman bunga.



Gambar 3. Pemandangan laut, hutan, dan taman bunga

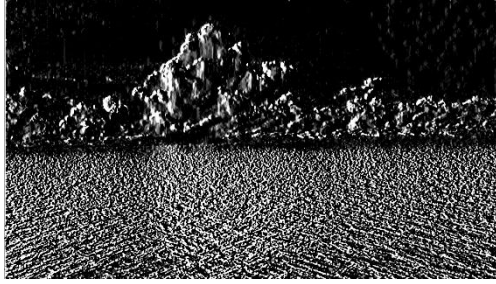
Pada ketiga gambar tersebut dapat disimpulkan bahwa pada gambar pemandangan laut objeknya didominasi oleh air, lalu pada gambar pemandangan hutan objeknya didominasi oleh pohon, dan yang terakhir pada gambar pemandangan taman bunga objeknya didominasi oleh berbagai jenis bunga dengan warna yang berbeda-beda.

*Library* yang digunakan pada pemrosesan gambar ini adalah sebagai berikut:

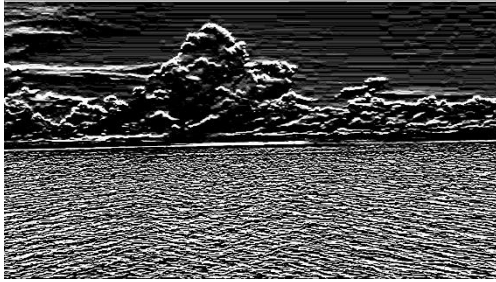
- `import cv2`
- `import matplotlib.pyplot as plt`
- `import numpy as np`
- `from matplotlib import pyplot as plt`
- `from matplotlib.pyplot import *`
- `from matplotlib import colors`
- `from skimage.io import imread`
- `from PIL import Image`
- `from pylab import *`
- `from collections import Counter`
- `from sklearn.cluster import Kmeans`

Pada *edge detection* dengan metode *sobel edge detection* dihasilkan *output* sebagai berikut:

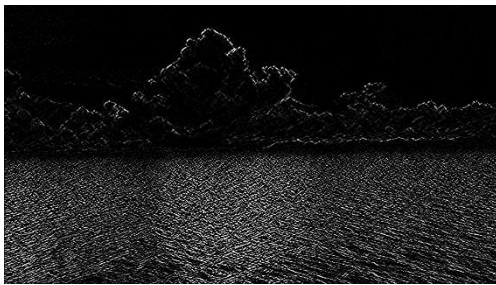
- a) Pemandangan Laut



Gambar 4. *Output* Sobel x pada gambar pemandangan laut

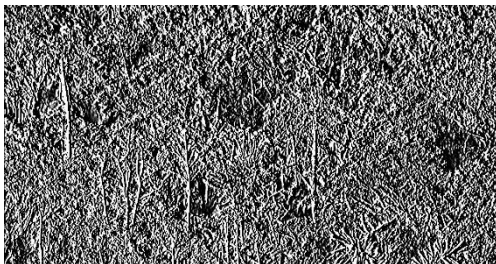


Gambar 5. *Output* Sobel y pada gambar pemandangan laut

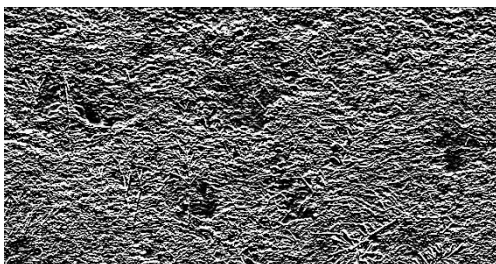


Gambar 6. *Output* Sobel xy pada gambar pemandangan laut

b) Pemandangan Hutan



Gambar 7. *Output* Sobel x pada gambar pemandangan hutan



Gambar 8. *Output* Sobel y pada gambar pemandangan hutan

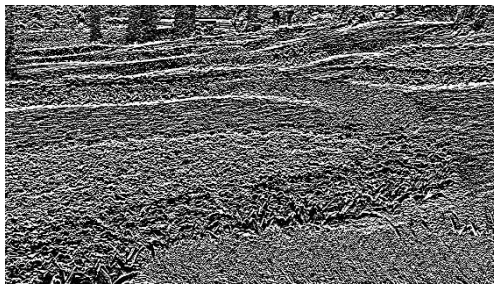


Gambar 9. *Output* Sobel xy pada gambar pemandangan hutan

c) Pemandangan Taman Bunga



Gambar 10. *Output* Sobel x pada gambar pemandangan taman bunga



Gambar 11. *Output* Sobel y pada gambar pemandangan taman bunga



Gambar 12. *Output* Sobel xy pada gambar pemandangan taman bunga

Pada *edge detection* dengan metode *canny edge detection* dihasilkan *output* gaussian blur sebagai berikut:



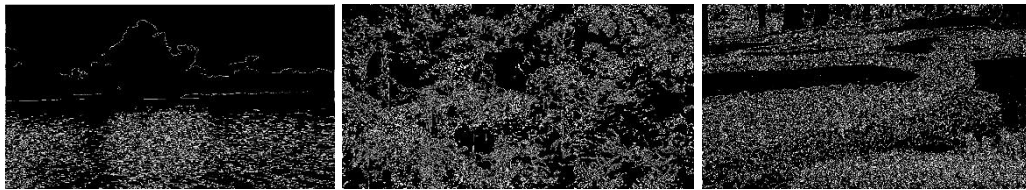


Gambar 13. *Output Gaussian Blur*

Sementara untuk *output median blur* dihasilkan gambar sebagai berikut:

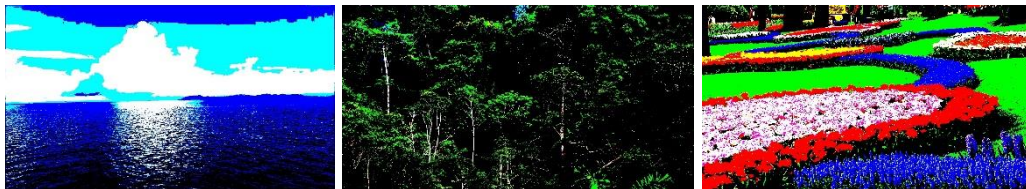
Gambar 14. *Output Median Blur*

Sementara untuk *output hysteresis thresholding* dihasilkan gambar sebagai berikut:

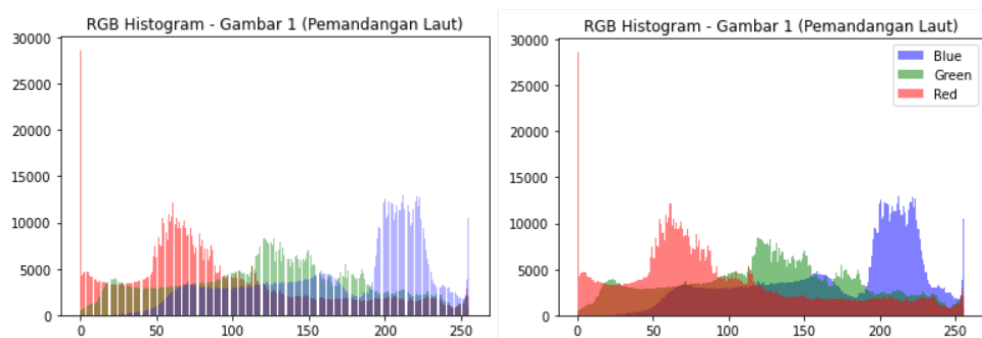
Gambar 15. *Output hysteresis thresholding*

Hasil dari *edge detection* ini yaitu bahwa objek masih terlihat jelas bentuknya walaupun berwarna hitam putih, karena gambar tercetak jelas oleh struktur tepinya. Metode ini mengetahui struktur yang mendasari sebuah gambar melalui tepinya.

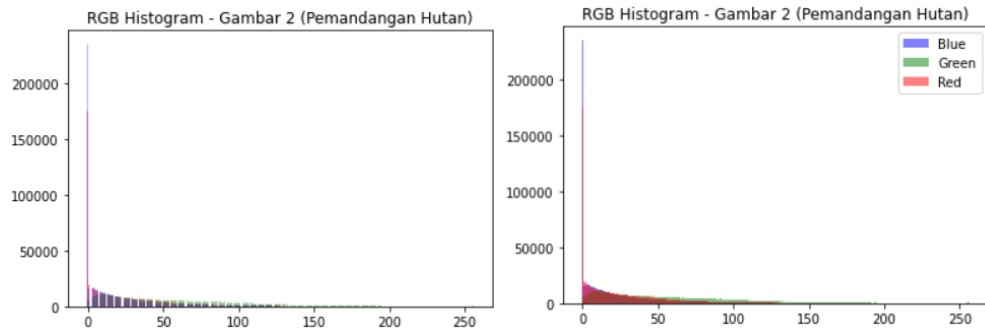
Pada variasi *thresholding* menggunakan  $\text{thresh} = 128$  dan menghasilkan *output* sebagai berikut:

Gambar 16. *Output variasi thresholding*

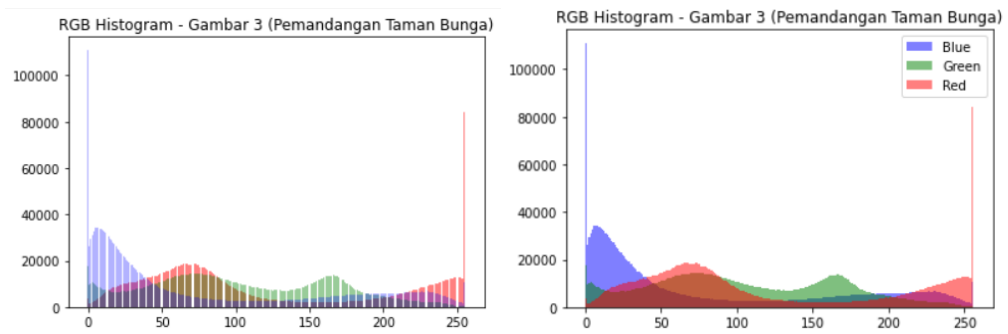
Pada analisis histogram RGB menghasilkan *output* histogram sebagai berikut:



Gambar 17. Histogram RGB pada gambar pemandangan laut

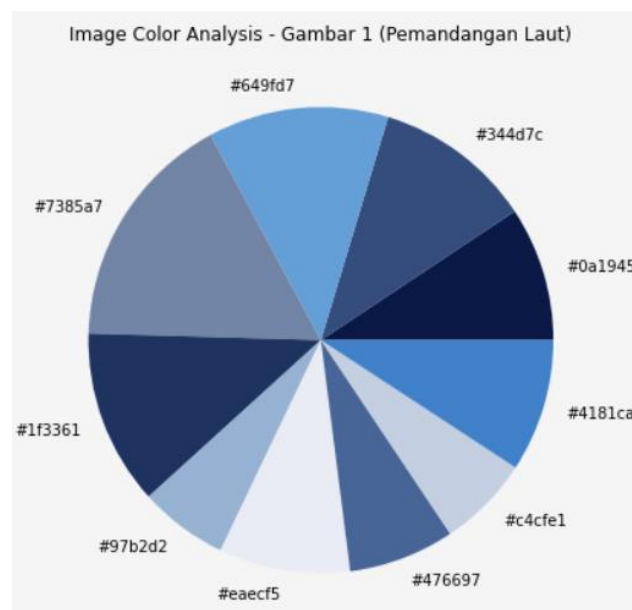


Gambar 18. Histogram RGB pada gambar pemandangan hutan

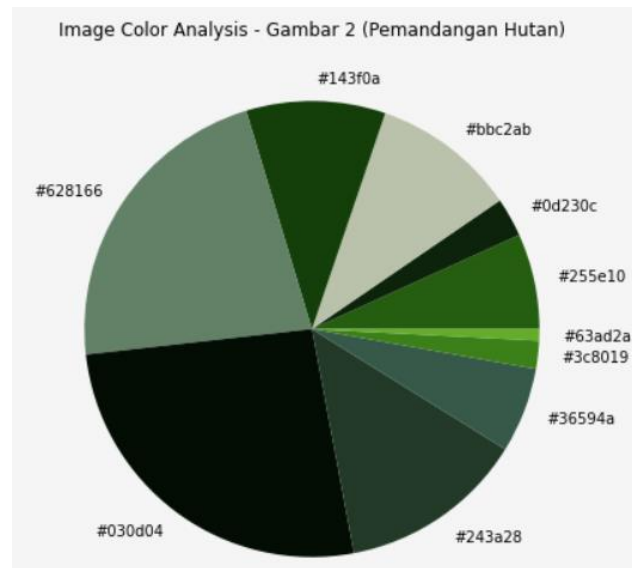


Gambar 19. Histogram RGB pada gambar pemandangan taman bunga

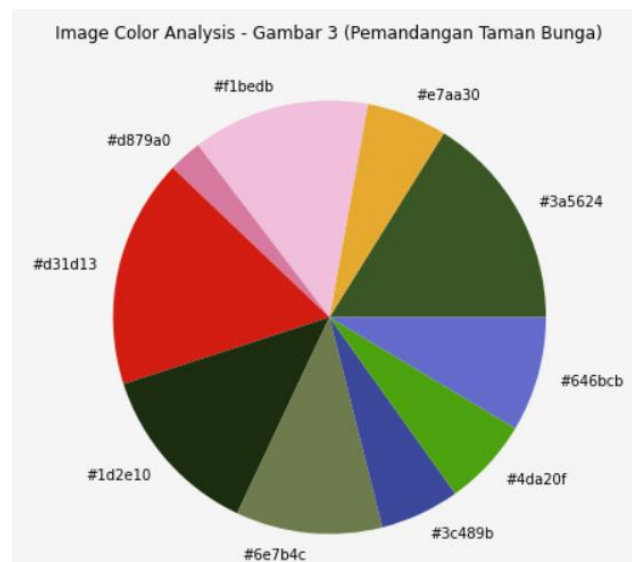
Pada analisis warna gambar menghasilkan *output* sebagai berikut:



Gambar 20. Diagram pie warna pada gambar pemandangan laut



Gambar 21. Diagram pie warna pada gambar pemandangan hutan



Gambar 22. Diagram pie warna pada gambar pemandangan taman bunga

Analisis warna gambar menggunakan metode diagram pie. Diagram ini terdiri dari 10 klaster warna, jumlah klaster ini bisa diubah sesuai keinginan. Hasil diagram pie warna adalah pada gambar pemandangan laut menghasilkan dominansi berbagai macam warna biru dari biru muda hingga biru tua, hal ini dikarenakan oleh objek gambar laut yang luas, awan, dan langit.

Sedangkan pada gambar pemandangan hutan menghasilkan dominansi berbagai macam warna hijau dari hijau muda hingga hijau tua, hal ini dikarenakan oleh objek gambar pepohonan berwarna hijau.

Sementara itu pada gambar pemandangan taman bunga menghasilkan

dominansi warna yang seimbang, ada warna merah, hijau, biru, oren, dan sebagainya, hal ini dikarenakan oleh objek gambar bunga yang mengandung berbagai macam warna.

Hasil konvolusi kernel adalah sebagai berikut:



Gambar 23. Hasil konvolusi kernel



## BAB IV

### KESIMPULAN

Pada *canny edge detection* menggunakan ambang bawah 100, dan ambang atas 200. Algoritma telah mengidentifikasi tepi dominan pada gambar, menghilangkan objek yang kurang penting bagi keseluruhan struktur. Namun, hasilnya dapat dengan mudah disesuaikan.

*Canny edge detection* memberikan hasil terbaik karena tidak hanya menggunakan *sobel edge detection*, tetapi juga *non-maximum suppression* dan *hysteresis thresholding*. Ini memberikan lebih banyak fleksibilitas dalam bagaimana tepi diidentifikasi dan dihubungkan pada tahap akhir algoritma.

Hasil dari *edge detection* ini yaitu bahwa objek masih terlihat jelas bentuknya walaupun berwarna hitam putih, karena gambar tercetak jelas oleh struktur tepinya. Metode ini mengetahui struktur yang mendasari sebuah gambar melalui tepinya.

Fitur yang menarik pada *image color analysis* adalah kita dapat menentukan jumlah kluster warna. Pada laporan ini menggunakan 10 kluster. Sisanya dilakukan dengan prediksi model Scikit-learn K-means dan OpenCV.

Hasil diagram pie warna adalah pada gambar pemandangan laut menghasilkan dominansi berbagai macam warna biru dari biru muda hingga biru tua, hal ini dikarenakan oleh objek gambar laut yang luas, awan, dan langit.

Sedangkan pada gambar pemandangan hutan menghasilkan dominansi berbagai macam warna hijau dari hijau muda hingga hijau tua, hal ini dikarenakan oleh objek gambar pepohonan berwarna hijau.

Sementara itu pada gambar pemandangan taman bunga menghasilkan dominansi warna yang seimbang, ada warna merah, hijau, biru, oren, dan sebagainya, hal ini dikarenakan oleh objek gambar bunga yang mengandung berbagai macam warna.

## DAFTAR PUSTAKA

<https://learnopencv.com/edge-detection-using-opencv/#sobel-edge>

<https://towardsdatascience.com/building-an-image-color-analyzer-using-python-12de6b0acf74>

Kusumaningtyas, S., & Asmara, R. A. (2016). Identifikasi Kematangan Buah Tomat Berdasarkan Warna Menggunakan Metode Jaringan Syaraf Tiruan (JST). *Jurnal Informatika Polinema*, 2(2), 72-72.

Mistry, K dan Saluja, A. (2016). An introduction to OpenCV using Python with Ubuntu. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, 1(2), hal. 65–68.

Nafi'iyah, N. (2017). Perbandingan Modus, Median, K\_Standar Deviasi, Iterative, Mean dan Otsu dalam Thresholding. *Spirit*, 8(2).

Ratna, S. (2020). Pengolahan Citra Digital dan Histogram dengan Phyton dan Text Editor Phycharm. *Technologia: Jurnal Ilmiah*, 11(3), 181-186.