

# Image Processing untuk Mengubah Warna Objek Citra Hasil Segmentasi

Haikal Lazuardi Fadil / 13519027  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
13519027@std.stei.itb.ac.id

**Abstract**—Teknik pemrosesan citra dapat diterapkan untuk berbagai hal, salah satunya mengubah warna objek pada citra. Untuk melakukannya, diperlukan informasi-informasi yang terdapat pada citra. Informasi pada citra dapat diekstraksi dengan pemrosesan citra. Salah satu metode untuk mendapatkan informasi dari suatu citra adalah dengan melakukan pendeteksian tepi. Setelah tepi-tepi yang ada di dalam citra diperoleh, citra dapat disegmentasi dan diubah warnanya.

**Kata kunci**—pemrosesan citra; warna; pendeteksian tepi; segmentasi;

## I. PENDAHULUAN

Di era teknologi seperti saat ini, pemrosesan citra menjadi salah satu bagian yang telah berkembang pesat. Teknologi pemrosesan citra menjadi dasar dari berbagai macam teknologi-teknologi lain yang jauh lebih maju seperti *license plate recognition* pada kamera lalu lintas, *vehicle detection* pada *self-driving vehicle*, dan lain-lain. Pemrosesan citra dapat dilakukan untuk perbaikan kualitas citra, analisis citra, restorasi citra, pemampatan citra, restorasi citra, rekonstruksi citra, dan segmentasi citra.

Segmentasi citra adalah teknik yang digunakan untuk membagi suatu citra menjadi sekumpulan *pixel* yang saling terhubung. Dalam prosesnya, terdapat berbagai macam langkah dan pendekatan berbeda untuk mengolah citra agar dapat tersegmentasi dengan baik. Kunci dari segmentasi citra yang baik adalah pendeteksian tepi yang baik. Segmentasi citra merupakan tahap awal sebelum melakukan *image understanding*, *object recognition*, perubahan warna objek, dan lain-lain. Segmentasi citra memiliki beberapa aplikasi yang cukup penting seperti untuk mendiagnosis kanker, *self-driving vehicle*, *object recognition*, *scene understanding*, dan lain-lain.

Dalam fotografi, terkadang warna objek yang tidak begitu sesuai dengan latarnya. Hal tersebut dapat membuat gambar yang diambil menjadi kurang hidup. Selain itu, dengan mengubah warna objek gambar, kita dapat memiliki berbagai macam gambar yang dapat digunakan dalam kondisi berbeda.

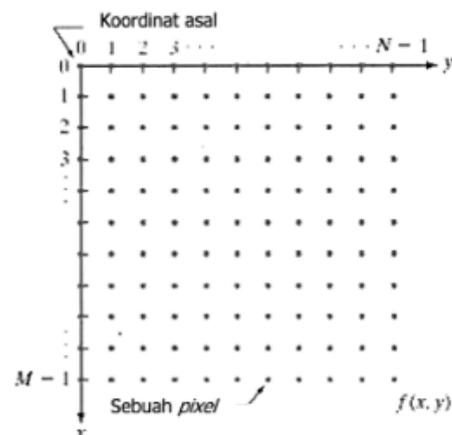
Makalah ini akan membahas segmentasi citra kemudian mengganti warna objek hasil segmentasi. Program memberikan citra keluaran berupa citra yang objeknya telah berubah warna

## II. LANDASAN TEORI

### A. Citra Digital

Secara umum, pemrosesan citra digital mengacu pada pemrosesan citra dua dimensi pada komputer. Dalam arti yang lebih luas, pemrosesan citra digital mengacu pada pemrosesan data dua dimensi apa pun. Citra digital adalah larik yang berisi nilai yang diwakili oleh kumpulan bit tertentu.

Sebuah citra dapat didefinisikan sebagai fungsi  $f(x,y)$  berukuran M baris dan N kolom, dimana  $x$  dan  $y$  adalah koordinat spasial yang disebut intensitas atau skala keabuan pada citra tersebut. Jika nilai  $f$ ,  $x$ , dan  $y$  diskrit serta berhingga, maka dapat dianggap bahwa citra tersebut adalah citra digital.



Gambar 1. Ilustrasi Koordinat Citra Digital

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \dots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Gambar 2. Koordinat Citra Digital dalam Matriks

Kedua citra tersebut menunjukkan ilustrasi digitalisasi citra yang memiliki 16 baris dan 16 kolom. Masing-masing nilai  $f$

pada matriks tersebut merepresentasikan derajat warna dari setiap *pixel* yang ada <sup>[1]</sup>.

### B. Pemrosesan Citra Digital

Pemrosesan citra digital adalah proses mengubah suatu citra menjadi citra lainnya untuk memenuhi kebutuhan tertentu. Biasanya, pemrosesan citra dilakukan untuk memenuhi kebutuhan tertentu seperti perbaikan untuk meningkatkan kualitas atau menitikberatkan pada bagian tertentu untuk menggali informasi pada citra, pengukuran elemen-elemen yang ada di dalam citra, dan penggabungan bagian citra yang satu dan yang lain.

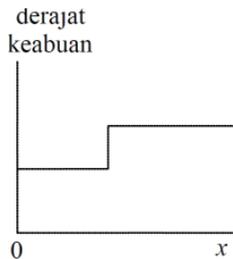
Dalam prosesnya, pemrosesan citra dapat dilakukan dengan berbagai metode seperti perbaikan kualitas citra, analisis citra, restorasi citra, pemampatan citra, restorasi citra, dan rekonstruksi citra.

### C. Pendeteksian Tepi

Pendeteksian tepi adalah metode yang paling umum digunakan untuk menemukan diskontinuitas skala abu-abu <sup>[2]</sup>. Diskontinuitas ini dapat ditemukan ketika terdapat perubahan derajat keabuan yang cukup besar dalam jarak *pixel* yang pendek. Dalam citra digital, terdapat tiga macam tepi yaitu:

#### 1. Tepi Curam

Tepi curam adalah tepi dengan selisih derajat keabuan yang sangat jauh. Tepi jenis ini biasanya memiliki nilai-nilai derajat keabuan yang cukup konsisten.



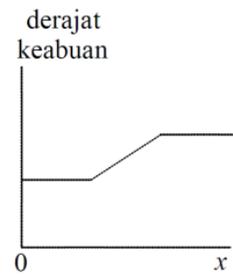
Gambar 3. Grafik Tepi Curam

$$\begin{bmatrix} 4 & 4 & 4 & 8 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 & 8 \end{bmatrix}$$

Gambar 4. Derajat Keabuan Tepi Curam

#### 2. Tepi Landai

Tepi landai adalah tepu dengan selisih derajat keabuan yang bertahap. Biasanya tepi ini terdiri dari sejumlah tepi-tepi kecil yang jaraknya berdekatan.



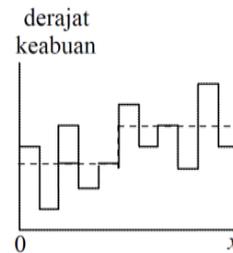
Gambar 5. Grafik Tepi Landai

$$\begin{bmatrix} 4 & 4 & 5 & 6 & 7 & 8 & 8 & 8 \\ 4 & 4 & 5 & 6 & 7 & 8 & 8 & 8 \\ 4 & 4 & 5 & 6 & 7 & 8 & 8 & 8 \\ 4 & 4 & 5 & 6 & 7 & 8 & 8 & 8 \\ 4 & 4 & 5 & 6 & 7 & 8 & 8 & 8 \end{bmatrix}$$

Gambar 6. Derajat Keabuan Tepi Landai

#### 3. Tepi Berderau

Tepi berderau biasanya di dapat pada citra hasil aplikasi *computer vision*. Sebelum melakukan pendeteksian tepi pada citra berderau, sebaiknya dilakukan peningkatan kualitas citra terlebih dahulu agar derau tidak terdeteksi sebagai tepi.



Gambar 7. Grafik Tepi Berderau

Pendeteksian tepi merupakan langkah yang esensial untuk mendapatkan informasi yang ada di dalam citra. Tepi - tepi tersebut. Operasi pendeteksian tepi bertujuan untuk menonjolkan penampakan garis batas suatu daerah atau objek dalam citra. Tepi pada citra termasuk ke dalam bagian-bagian citra yang memiliki frekuensi tinggi. Oleh karena itu, pendeteksian tepi lebih baik dilakukan menggunakan penapis lolos tinggi. Operator pendeteksian tepi memiliki matriks kernel yang biasanya ditandai dengan variabel H untuk horizontal dan V untuk vertikal. Berikut adalah beberapa operator yang dapat digunakan untuk operasi pendeteksian tepi:

#### 1. Operator Sobel

Operator Sobel merupakan operator yang paling populer diantara operator lainnya. Hal ini disebabkan kesederhanaan dan keefektifitasannya. Operator ini

merupakan operator tingkat lanjut dari operator Robert yang menggunakan penapis lolos tinggi. Operator ini memiliki kelebihan yaitu mengurangi noise sebelum pendeteksian tepi dilakukan.

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Gambar 8. Matriks Kernel Operator Sobel

## 2. Operator Prewitt

Operator Prewitt juga merupakan operator tingkat lanjut dari operator Roberts. Operator ini menggunakan penapis lolos tinggi yang diberi angka nol sebagai penyangga. Mengambil prinsip laplacian, operator ini menggunakan kekuatan gradient yang ditinjau dari sudut pandang horizontal dan vertikal.

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Gambar 9. Matriks Kernel Operator Prewitt

## 3. Operator Roberts

Operator Roberts merupakan variasi dari Operator Gradien yang memiliki arah orientasi sebesar 45 dan 135 derajat relatif terhadap bidang citra. Berupa penjabaran dari teknik diferensial, operator ini menambah proses konversi biner pada arah horizontal dan vertikal dengan meratakan distribusi warna hitam dan putih.

$$H = \begin{bmatrix} -1 & 1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Gambar 10. Matriks Kernel Operator Roberts

## 4. Operator Canny

Operator Canny adalah operator modern yang memiliki beberapa kriteria pendeteksi tepian beberapa kriteria pendeteksi tepian yaitu mendeteksi dengan baik, respon yang jelas, dan melokalisasi dengan baik. Hasil pendeteksian dengan operator Canny sangat dipengaruhi oleh parameter-parameternya. Parameter-parameter tersebut adalah nilai ambang dan standar deviasi gaussian<sup>[3]</sup>.

## D. Segmentasi Citra

Segmentasi citra adalah teknik yang digunakan untuk membagi suatu citra menjadi sekumpulan *pixel* yang saling

terhubung. Berbagai jenis potongan citra biasanya digunakan. Contohnya termasuk partisi menjadi beberapa wilayah (biasanya menutupi seluruh citra), struktur linier seperti garis atau kurva, dan bentuk geometris dua dimensi seperti lingkaran, elips, kotak, dan lain-lain.

Segmentasi citra menjadi beberapa wilayah atau wilayah dapat digunakan untuk berbagai hal, seperti membagi citra menjadi segmen dan kelompok objek yang berbeda, atau memisahkan objek utama dari latar belakang citra. Hasil dari segmentasi citra kemudian dapat digunakan untuk mengolah informasi yang paling penting pada citra, terhadap pemrosesan seluruh bagian citra yang tidak semuanya mengandung informasi penting. Tujuan akhir segmentasi citra adalah menemukan objek atau bagian yang seragam atau spesifik untuk diproses sesuai dengan tugas yang akan dilakukan, seperti deteksi objek, analisis, atau ekstraksi informasi dari citra.

Teknik segmentasi citra dapat dilakukan berdasarkan berbagai fitur yang dipilih pada citra, seperti warna, tekstur atau kecerahan. Setiap bagian digabungkan dan dihomogenkan sesuai dengan karakteristik penentu yang dipilih. Hasil segmentasi ini dapat digunakan di berbagai bidang seperti kedokteran, deteksi kendaraan otonom, pengenalan objek, dan pemahaman pemandangan. Metode segmentasi citra biasanya dikelompokkan menjadi dua pendekatan yaitu:

### 1. Kemiripan

Citra dipartisi berdasarkan fitur yang dijadikan sebagai penentu dalam proses segmentasi citra. Metode-metode yang termasuk dalam pendekatan kemiripan adalah *clustering*, *region growing*, *thresholding*, dan *split and merge*. Pendekatan ini melihat kemiripan dari fitur yang dijadikan penentu untuk menentukan *pixel* mana saja yang merupakan daerah dari suatu citra.

### 2. Diskontinuitas

Dalam pendekatan ini, segmentasi citra didasarkan pada nilai intensitas *pixel* yang berubah dengan cepat. Salah satu contoh perubahan nilai adalah rand. Tepi citra diperoleh dengan menggunakan teknik deteksi tepi<sup>[4]</sup>.

## E. Warna

Warna adalah suatu spektrum yang terdapat di dalam suatu cahaya sempurna (berwarna putih). Spektrum tersebut terdiri dari spektrum-spektrum lainnya. Gelombang warna yang dapat dilihat oleh mata manusia disebut juga dengan sinar tampak yaitu gelombang yang memiliki panjang 400nm hingga 700nm

Citra mengandung berbagai informasi yang dapat digunakan untuk berbagai keperluan. Salah satu informasi tersebut adalah warna. Warna adalah informasi yang dapat digunakan untuk berbagai hal seperti deteksi objek, fitur dan ekstraksi data, pengenalan wajah berdasarkan warna, segmentasi citra menjadi bagian yang berbeda berdasarkan warna, dan lain-lain.

Dalam citra digital, komponen warna direpresentasikan dalam bentuk ruang. Terdapat berbagai jenis warna dalam citra yang memiliki tujuan tersendiri. Jenis-jenis warna tersebut diantaranya lain:

1. RGB
2. HSI dan HSV
3. CMY dan CMYK
4. XYZ
5. YCbCr

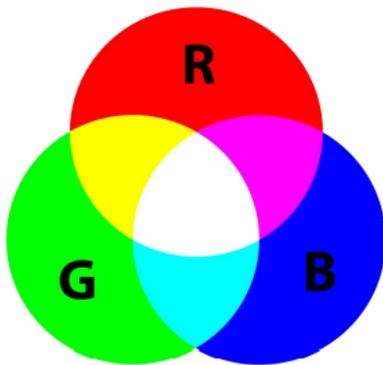
Jenis warna yang digunakan pada tugas makalah ini adalah jenis RGB dan HSV. Berikut adalah penjelasan mengenai RGB dan HSV.

1. RGB

RGB adalah jenis warna yang terdiri dari tiga komponen yaitu *Red*, *Green*, dan *Blue*. Ketiga warna tersebut merupakan warna primer. Secara umum, mayoritas citra digital menggunakan jenis warna ini. Untuk mendefinisikan jenis warna ini, terdapat rumus yang memanfaatkan tiga komponen warna tersebut yaitu

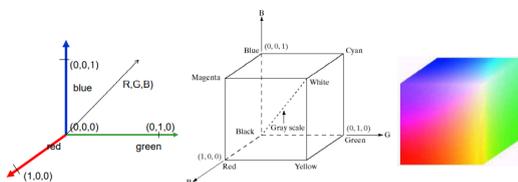
$$C = aR + bG + cB$$

Berikut adalah ilustrasi jenis warna RGB yang direpresentasikan dengan spektrum.



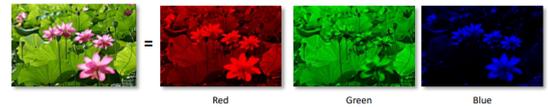
Gambar 11. Spektrum RGB

Warna yang dihasilkan berdasarkan rumus tersebut merupakan warna akhir dari jenis warna RGB sesuai dengan ruangnya. Berikut adalah ilustrasi ruang RGB yang telah dinormalisasi dari 0 hingga 1.



Gambar 12. Ruang RGB

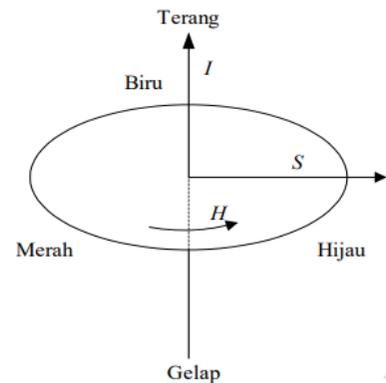
RGB adalah jenis warna yang bersifat aditif karena menjumlahkan setiap komponen-komponennya untuk menghasilkan warna. Berikut adalah contoh dekomposisi citra yang memiliki jenis warna RGB.



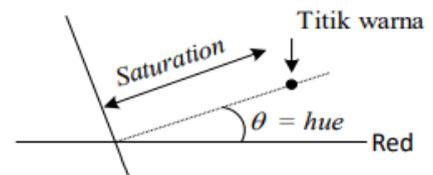
Gambar 13. Citra RGB

2. HSI dan HSV

Jenis warna HSI merupakan jenis warna yang deskripsinya mendekati warna manusia. Hal ini disebabkan oleh manusia yang tidak umum mendeskripsikan warna dalam bentuk nilai-nilai RGB. Manusia menjelaskan warna dalam komponen *Hue*, *Saturation*, dan *Intensity*. Oleh karena itu, jenis ini dinamakan dengan HSI. Berikut adalah ilustrasi sumbu dari HSI.



Gambar 14. Sumbu HSI



Gambar 15. Sumbu HSI Close-Up

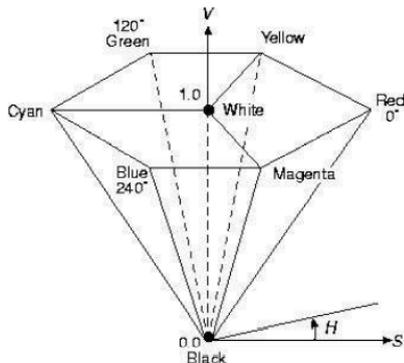
Berdasarkan gambar di atas, *hue* adalah sudut yang bernilai 0 hingga  $2\pi$ , *saturation* adalah jarak menuju sumbu vertikal yang bernilai 0 hingga 1, dan *intensity* adalah tinggi sepanjang garis vertikal yang bernilai 0 hingga 1.

*Hue* merepresentasikan perbedaan warna seperti kemerahan, kehijauan, dan lain-lain. Komponen ini juga menyatakan warna yang sebenarnya. *Hue* sendiri berhubungan dengan panjang gelombang cahaya.

*Saturation* merepresentasikan tingkat kemurnian cahaya. Dengan kata lain, *saturation* adalah indikator jumlah dari warna putih yang ada pada suatu warna. Bisa dibilang, *saturation* menyatakan seberapa dalam suatu warna.

*Intensity* merepresentasikan seberapa banyak cahaya yang diterima oleh mata tanpa menghiraukan warna. Nilai dari *intensity* sendiri menyatakan gelap jika bernilai 0 dan terang jika 1.

HSV merupakan jenis warna yang mirip dengan HSI. Perbedaannya ada pada nilai V yang menyatakan *Value* atau kecerahan dari suatu warna di mana memiliki rentang 0 hingga 1. Berikut adalah ilustrasi dari sumbu HSV.



Gambar 16. Sumbu HSV

### III. IMPLEMENTASI

Solusi dari masalah ini diimplementasikan dengan membuat sebuah program menggunakan bahasa pemrograman MATLAB. Program akan menerima sebuah citra masukan yang kemudian akan dilakukan segmentasi pada citra tersebut. Hasil segmentasi kemudian akan diubah warnanya kemudian akan digabung dengan gambar asal. Citra keluaran berupa sebuah citra di mana objek hasil segmentasi telah berubah warnanya<sup>[5]</sup>.

#### A. Pengkonversian Citra Menjadi Grayscale

Berikut adalah citra masukan yang digunakan dalam tahap implementasi solusi.



Gambar 17. Citra Masukan

Agar pendeteksian tepi dapat dilakukan, citra dikonversi menjadi *grayscale* dengan bantuan fungsi milik MATLAB. Berikut kode yang ditulis untuk membaca citra masukan dan mengkonversinya menjadi citra *grayscale*.

```
img = imread("img/flower.png");
I = rgb2gray(img);
```

Berikut adalah hasil pengkonversian citra masukan menjadi citra *grayscale*.



Gambar 18. Citra Grayscale

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

#### B. Pendeteksian Tepi dan Segmentasi Citra

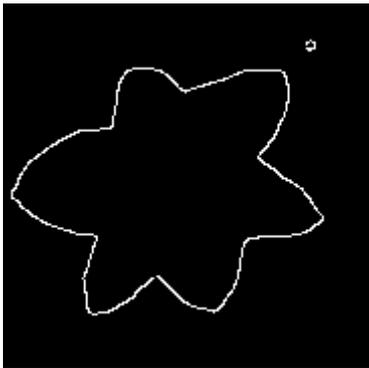
Untuk mendapatkan informasi mengenai bagian mana saja yang merupakan objek dan mana yang bukan, dilakukan pendeteksian tepi pada citra *grayscale*. Setelah melakukan berbagai percobaan, operator pendeteksian tepi yang digunakan adalah operator Canny dengan menggunakan fungsi milik MATLAB. Untuk meningkatkan keefektifitasan program, penulis menggunakan operator Sobel untuk mendapatkan *threshold* dari citra kemudian memodifikasi *threshold* tersebut dan menggunakannya pada operator Canny.

```
[N_, threshold] = edge(I, 'Sobel');
step = 0.4;
sensitivity = threshold + step;
N_edge = edge(I, 'Canny', sensitivity);
```

Dengan mengubah-ubah *threshold*, penulis dapat mendapatkan hasil pendeteksian tepi yang cukup akurat pada citra. Penulis menyimpan penambahan *threshold* pada variabel *step*. Untuk citra masukan yang penulis gunakan, *step* yang sesuai adalah 0.4. Berikut adalah perbandingan sebelum dan setelah memodifikasi *threshold*.



Gambar 19. Step Bernilai Nol



Gambar 20. Step Bernilai 0.4

Berdasarkan gambar diatas, bisa dilihat bahwa pendeteksian tepi sudah bekerja dengan baik dengan menyeleksi hanya pada bagian kelopak bunga saja. Namun, terdapat tepi yang terputus-putus. Oleh karena itu, penulis menggunakan fungsi milik MATLAB untuk menyambungkan garis tepi yang terputus.

```
N_close =
imclose(N_edge, strel('line', 20, 0));
```



Gambar 21. Hasil Penyambungan Garis Tepi

Setelah tepi yang dibutuhkan tersambung, bagian-bagian yang berada di dalam tepi diisi untuk membedakan objek citra yang

ingin diseleksi dan yang tidak menggunakan fungsi milik MATLAB.

```
object = imfill(N_close, 'holes');
```



Gambar 22. Hasil Pengisian Tepi

Setelah tepi diisi, kita mulai dapat melihat gambaran objek yang akan disegmentasi. Namun, masih terdapat sedikit derau yang dianggap sebagai tepi. Untuk mengatasi hal ini, digunakan fungsi milik MATLAB yang disebut `bwareaopen` yang akan menghapus pendeteksian tepi di bawah jumlah *pixel* tertentu.

```
N = bwareaopen(object, 2000);
```



Gambar 23. Hasil Penghapusan Derau

Setelah semua berjalan dengan lancar, dilakukan segmentasi citra dengan cara melakukan *element-wise multiplication* terhadap citra masukan untuk mengekstraksi objek.

```
red_processed = img(:,:,1).*uint8(N);
green_processed = img(:,:,2).*uint8(N);
blue_processed = img(:,:,3).*uint8(N);
segmented_image = cat(3, red_processed,
```

```
green_processed, blue_processed);
```



Gambar 24. Hasil Segmentasi

Setelah citra berhasil tersegmentasi dengan baik, selanjutnya dilakukan perubahan warna dari objek yang tersegmentasi. Untuk melakukan hal tersebut, citra perlu dikonversi menjadi HSV. Setelah itu, penulis menggunakan variabel  $k$  sebagai nilai pergeseran warna dari warna asal. Setelah warna dari objek selesai diubah, citra kembali dikonversi menjadi RGB kemudian dilakukan denormalisasi agar *pixel* kembali bernilai 0 hingga 255.

```
k = 0.1;
hsvImage = rgb2hsv(segmented_image);
hsvImage(:, :, 1) =
mod(hsvImage(:, :, 1)+k, 1);
rgbImage = hsv2rgb(hsvImage);
rgbImage_ = uint8(rgbImage .* 255);
```



Gambar 25. Hasil Perubahan Warna pada Objek

Langkah terakhir yang perlu dilakukan adalah mengganti nilai-nilai *pixel* pada citra masukan dengan objek yang telah diubah warnanya. Dengan begitu, citra yang dihasilkan adalah citra masukan dengan warna objek yang berubah.

```
[x, y, z] = size(rgbImage_);
```

```
temp = zeros(x, y, z);
for i = 1 : x
    for j = 1 : y
        for k = 1 : z
            if (rgbImage_(i, j, k) == 0)
                temp(i, j, k) = img(i, j,
k);
            else
                temp(i, j, k) =
rgbImage_(i, j, k);
            end
        end
    end
end
```



Gambar 26. Citra Keluaran

#### IV. KESIMPULAN

Pemrosesan citra dapat dimanfaatkan untuk berbagai hal. Salah satu aplikasinya untuk mengubah warna suatu objek gambar. Berdasarkan hasil keluaran citra, program berhasil mengubah warna objek pada citra.

Berdasarkan hasil percobaan, terdapat berbagai kekurangan pada program seperti sebagian kecil latar belakang yang terdeteksi sebagai objek. Selain itu, peningkatan atau penurunan *threshold* untuk citra berbeda satu dengan yang lain. Sehingga perlu dilakukan *trial and error* untuk masing-masing citra.

#### LINK VIDEO

<https://youtu.be/u1hIU6DMhdc>

#### REFERENCES

- [1] Putra, D. (2010). Pengolahan citra digital. Penerbit Andi.
- [2] Febriani, F., & ETP, L. (2008). Analisis Penelusuran Tepi Citra menggunakan Detektor Tepi Sobel dan Canny. In Proceeding, Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2008) Auditorium Universitas Gunadarma, Depok, 20-21 Agustus 2008.
- [3] Pratama, B. Y. (2007). Pendeteksian Tepi Pengolahan Citra Digital.

- [4] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/22-Segmentasi-Citra-Bagian1-2022.pdf>. Diakses 16 Desember 2022
- [5] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/16-Warna-bagian1-2022.pdf>. Diakses 16 Desember 2022

#### 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, 17 Desember 2022



Haikal Lazuardi Fadil  
13519027

