

**UTS**

**PENGOLAHAN CITRA**



NAMA : Pramesti Anggreini Romadona

NIM : 202331021

KELAS : A

DOSEN : Dr.Dra.Dwina Kuswardani, M.kom

NO.PC : 20

ASISTEN : 1. Clarenca Sweetdiva Pereira

2. Viana Salsabila Fairuz Syahla

3. Kashrina Masyid Azka

4. Sasikirana Ramadhanty Setiawan Putri

**INSTITUT TEKNOLOGI PLN**

**TEKNIK INFORMATIKA**

**2024/2025**

## DAFTAR ISI

DAFTAR ISI .....	2
BAB I .....	3
PENDAHULUAN.....	3
<b>1.1 Rumusan Masalah</b> .....	3
<b>1.2 Tujuan Masalah</b> .....	3
<b>1.3 Manfaat Masalah</b> .....	4
BAB II .....	5
LANDASAN TEORI .....	5
<b>2.1. Pengolahan Citra Digital</b> .....	5
<b>2.2. Segmentasi warna dan Ruang Warna HSV</b> .....	5
<b>2.3. Teknik Thresholding dalam segmentasi</b> .....	6
<b>2.4. Perbaikan Citra Backlight</b> .....	7
<b>2.5. Peran Python dan OpenCV</b> .....	7
BAB III .....	9
HASIL.....	9
<b>3.1 Deteksi Warna Pada Citra</b> .....	9
<b>3.2 Mencari Ambang Batas Terkecil Samapai Dengan Terbesar</b> .....	11
<b>3.3 Memperbaiki Gambar Backlight</b> .....	12
BAB IV .....	14
PENUTUP .....	14
DAFTAR PUSTAKA .....	15

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Rumusan Masalah**

Pengolahan citra digital merupakan salah satu bidang penting dalam ilmu komputer dan teknik informatika yang memiliki peran strategis dalam membantu komputer untuk mengenali, memproses, dan mengambil informasi dari gambar atau video. Dalam dunia nyata, penerapan pengolahan citra sangat luas, mulai dari sistem pengawasan keamanan, pengenalan wajah, klasifikasi objek, hingga aplikasi medis dan kendaraan otonom. Oleh karena itu, pemahaman yang mendalam mengenai teknik dasar pengolahan citra menjadi hal yang sangat penting bagi mahasiswa dalam mengembangkan sistem berbasis visual secara komputasional.

Salah satu tantangan mendasar yang diangkat dalam praktikum ini adalah bagaimana mendeteksi warna-warna primer tertentu seperti merah, hijau, dan biru dari sebuah citra digital. Mendeteksi warna bukanlah tugas yang sepele, karena sebuah citra bisa memiliki beragam variasi warna dengan intensitas dan pencahayaan yang berbeda. Selain itu, distribusi warna bisa sangat kompleks karena adanya noise, bayangan, atau pencahayaan yang tidak merata. Oleh karena itu, dibutuhkan teknik segmentasi warna yang mampu bekerja secara efektif, salah satunya dengan menggunakan transformasi ke ruang warna HSV, yang memungkinkan pemisahan komponen warna (hue), saturasi, dan kecerahan secara lebih fleksibel dibandingkan dengan RGB.

Permasalahan lainnya adalah bagaimana menentukan ambang batas (threshold) warna yang paling optimal untuk mengkategorikan objek pada gambar. Threshold sangat berpengaruh dalam proses segmentasi, karena nilai ambang yang terlalu rendah atau terlalu tinggi dapat menyebabkan objek warna tidak terdeteksi dengan baik atau bercampur dengan warna lain. Penggunaan ambang batas juga berkaitan erat dengan analisis histogram yang menjadi acuan untuk mengidentifikasi distribusi warna dalam gambar dan menentukan nilai-nilai intensitas yang representatif.

Selain segmentasi warna, tantangan berikutnya adalah memperbaiki kualitas citra yang mengalami efek backlight. Backlight terjadi ketika sumber cahaya terlalu terang berada di belakang objek, sehingga bagian depan objek menjadi gelap dan tidak terlihat jelas. Kondisi ini mengharuskan dilakukan perbaikan citra dengan teknik peningkatan kontras dan konversi grayscale agar informasi visual pada citra tetap terbaca. Dengan demikian, praktikum ini tidak hanya menekankan pada proses segmentasi warna, tetapi juga bagaimana meningkatkan kualitas citra agar dapat digunakan secara optimal dalam proses analisis selanjutnya.

#### **1.2 Tujuan Masalah**

1. Menerapkan metode segmentasi warna berbasis warna HSV untuk mengekstraksi objek berwarna tertentu dari citra digital.

2. Menentukan dan mengurutkan nilai ambang batas (threshold) dari yang terkecil hingga terbesar yang digunakan untuk mengkategorikan warna pada citra.
3. Mengimplementasikan teknik perbaikan kualitas citra seperti grayscale dan peningkatan kontras untuk mengatasi permasalahan backlight.
4. Mengembangkan keterampilan praktis dalam menerapkan teori-teori pengolahan citra digital pada kasus nyata menggunakan bahasa pemrograman Python.

### **1.3 Manfaat Masalah**

1. Memberikan pemahaman mendalam tentang prinsip kerja ruang warna HSV dan penerapannya dalam proses segmentasi warna.
2. Melatih kemampuan analisis untuk menentukan ambang batas optimal dalam pemrosesan citra berdasarkan karakteristik warna yang ditargetkan
3. meningkatkan kemampuan dalam memperbaiki kualitas citra, khususnya citra dengan kondisi pencahayaan yang menantang seperti backlight.
4. Menyiapkan peserta praktikum untuk menghadapi permasalahan nyata dalam pengolahan citra digital yang relevan dengan dunia industri dan riset teknologi.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Pengolahan Citra Digital**

Pengolahan citra digital adalah proses memanipulasi gambar digital menggunakan komputer untuk mendapatkan informasi yang berguna. Citra digital terdiri dari matriks piksel yang mewakili intensitas cahaya dalam warna tertentu. Proses ini melibatkan tahapan-tahapan seperti akuisisi citra, pra-pemrosesan, segmentasi, ekstraksi fitur, dan klasifikasi. pengolahan citra telah menjadi pilar utama dalam berbagai aplikasi teknologi karena kemampuannya dalam memahami struktur visual objek, bahkan dalam lingkungan yang kompleks.

Dalam praktikum ini, pengolahan citra digital digunakan untuk melakukan deteksi warna serta perbaikan kualitas visual pada citra. Dengan bantuan bahasa pemrograman Python dan pustaka seperti OpenCV, proses seperti pembacaan gambar, transformasi ruang warna, thresholding, hingga peningkatan kontras dapat dilakukan secara efisien. Pentingnya penguasaan konsep dasar pengolahan citra adalah untuk memastikan bahwa hasil yang diperoleh dari proses analisis bersifat akurat dan dapat diandalkan. Tahapan umum pengolahan Citra :

- a) Akuisisi Citra: Proses mendapatkan gambar dari perangkat input seperti kamera atau scanner.
- b) Pra-pemrosesan: Menghilangkan noise, mengatur ukuran gambar, konversi ke grayscale atau HSV, dll.
- c) Segmentasi: Pemisahan objek dari latar belakang berdasarkan fitur tertentu (warna, intensitas, tekstur).
- d) Ekstraksi Fitur: Mengambil informasi penting dari objek, seperti bentuk, tepi, warna dominan.
- e) Klasifikasi dan Interpretasi: Menentukan kategori atau identitas objek berdasarkan fitur yang diekstrak[1].

#### **2.2. Segmentasi warna dan Ruang Warna HSV**

Segmentasi warna adalah proses penting dalam pengolahan citra yang bertujuan untuk memisahkan area tertentu berdasarkan karakteristik warnanya. Salah satu pendekatan yang banyak digunakan adalah konversi dari ruang warna RGB ke HSV. Ruang warna HSV (Hue, Saturation, Value) memisahkan informasi warna (hue) dari intensitasnya (value), membuatnya lebih cocok untuk pengenalan warna dalam kondisi pencahayaan yang bervariasi.

HSV memiliki keunggulan dalam mendeteksi warna yang konsisten dalam kondisi pencahayaan yang berubah-ubah, berbeda dengan RGB yang sangat bergantung pada pencahayaan. Hue mewakili jenis warna (misalnya merah atau hijau), saturasi menunjukkan

kejenuhan warna, dan value menggambarkan kecerahan. Segmentasi dilakukan dengan menentukan nilai minimum dan maksimum dari ketiga parameter ini, kemudian memfilter piksel yang berada dalam rentang tersebut.

Transformasi ruang warna ini penting karena manusia mengenali objek lebih berdasarkan hue daripada intensitas cahaya. Oleh karena itu, HSV mencerminkan persepsi warna yang lebih alami dan memberikan akurasi yang lebih tinggi dalam proses segmentasi dibandingkan dengan RGB.

- Penjelasan Ruang warna HSV :
  - a) Hue (H): Menentukan tipe warna (merah, kuning, hijau, biru, dll) dan dinyatakan dalam derajat (0–179 di OpenCV).
  - b) Saturation (S): Menunjukkan seberapa murni atau jenuh warna tersebut (semakin tinggi semakin mencolok).
  - c) Value (V): Intensitas cahaya atau kecerahan.
- Keunggulan HSV dalam segmentasi :
  - a) Hue stabil terhadap perubahan pencahayaan, sehingga lebih akurat untuk deteksi warna di kondisi nyata.
  - b) RGB bisa berubah drastis hanya karena pencahayaan berubah, tapi hue relatif konsisten.
- Teknik Segmentasi
  - a) Transformasi RGB → HSV
  - b) Tentukan rentang H, S, dan V untuk warna target
  - c) Gunakan fungsi seperti `cv2.inRange()` untuk menghasilkan mask biner dari warna tersebut[2].

### 2.3. Teknik Thresholding dalam segmentasi

Thresholding adalah teknik dasar untuk mengubah citra ke dalam bentuk biner, yang membantu dalam memisahkan objek dari latar belakang. Teknik ini bekerja dengan membandingkan nilai intensitas atau komponen warna dengan nilai ambang tertentu. Jika nilai piksel berada di dalam rentang threshold, maka dianggap sebagai bagian dari objek; jika tidak, maka dianggap sebagai latar belakang.

Dalam konteks deteksi warna, threshold diterapkan pada setiap kanal HSV secara bersamaan. Hal ini memerlukan pemahaman yang baik tentang bagaimana nilai-nilai HSV berinteraksi agar segmentasi berhasil. menyatakan bahwa metode threshold adaptif yang mempertimbangkan histogram distribusi warna dapat meningkatkan performa segmentasi terutama pada citra dengan kondisi pencahayaan yang kompleks.

Teknik threshold juga menjadi dasar bagi metode segmentasi lanjutan seperti Otsu, k-means, dan metode berbasis deep learning. Namun, dalam konteks praktikum ini, threshold manual berbasis uji coba tetap menjadi pendekatan yang efektif untuk citra sederhana dan memperkuat pemahaman terhadap karakteristik warna dan intensitas.

- Jenis therasholding :
  - a. Global Thresholding: Menggunakan satu nilai ambang untuk seluruh gambar
  - b. Adaptive Thresholding: Menghitung threshold berdasarkan area lokal, cocok untuk gambar dengan pencahayaan tidak merata
  - c. Otsu's Method: Menentukan nilai threshold optimal secara otomatis berdasarkan histogram intensitas[3].

#### 2.4. Perbaikan Citra Backlight

Citra dengan efek backlight sering kali memiliki objek utama yang tampak gelap karena cahaya kuat datang dari belakang objek. Hal ini menurunkan kualitas informasi visual dan menyulitkan proses pengenalan objek. Salah satu solusi utama adalah konversi ke citra grayscale untuk mengeliminasi informasi warna dan memfokuskan pada intensitas cahaya.

Setelah citra dikonversi ke grayscale, langkah berikutnya adalah peningkatan kontras. Teknik seperti histogram equalization atau contrast stretching digunakan untuk memperlebar distribusi intensitas, membuat bagian gelap menjadi lebih terang. menunjukkan bahwa penggunaan histogram lokal (adaptive histogram equalization) mampu meningkatkan detail dalam area gelap tanpa merusak bagian terang.

Selain metode tersebut, teknik perbaikan lainnya adalah pengaplikasian transformasi logaritmik atau power-law (gamma correction) yang bertujuan memperkuat detail dalam area dengan pencahayaan lemah. Implementasi metode ini dalam praktikum memberikan pemahaman langsung kepada mahasiswa mengenai bagaimana kondisi citra dapat ditingkatkan dan informasi yang tersembunyi dapat ditampilkan kembali secara efektif[4].

#### 2.5. Peran Python dan OpenCV

Python adalah bahasa pemrograman yang banyak digunakan dalam pengolahan citra karena sintaksnya yang sederhana dan pustaka yang lengkap. Salah satu pustaka yang paling populer adalah OpenCV (Open Source Computer Vision Library), yang menyediakan berbagai fungsi untuk membaca gambar, mengubah ruang warna, menerapkan filter, melakukan transformasi morfologi, dan segmentasi.

Dengan menggunakan OpenCV, proses transformasi RGB ke HSV, pengaplikasian threshold, serta perbaikan kontras dapat dilakukan hanya dalam beberapa baris kode. Hal ini memungkinkan proses eksperimen dan evaluasi dilakukan dengan cepat, sekaligus memberi fleksibilitas tinggi bagi pengguna untuk menyesuaikan parameter yang digunakan.

Kombinasi Python dan OpenCV membuat praktikum ini tidak hanya fokus pada teori, tetapi juga memberikan pengalaman praktis yang relevan dengan kebutuhan industri modern, di mana pengolahan citra digunakan untuk berbagai aplikasi seperti kamera pintar, sistem keamanan, dan kendaraan otonom.

- Keunggulan OpenCV:
  - a. Mendukung banyak format gambar dan video
  - b. Efisien dalam operasi pixel-wise, filter, transformasi ruang warna, dan segmentasi
  - c. Integrasi mudah dengan pustaka lain seperti NumPy, SciPy, matplotlib, dan TensorFlow (untuk deep learning)[5].



## BAB III

### HASIL

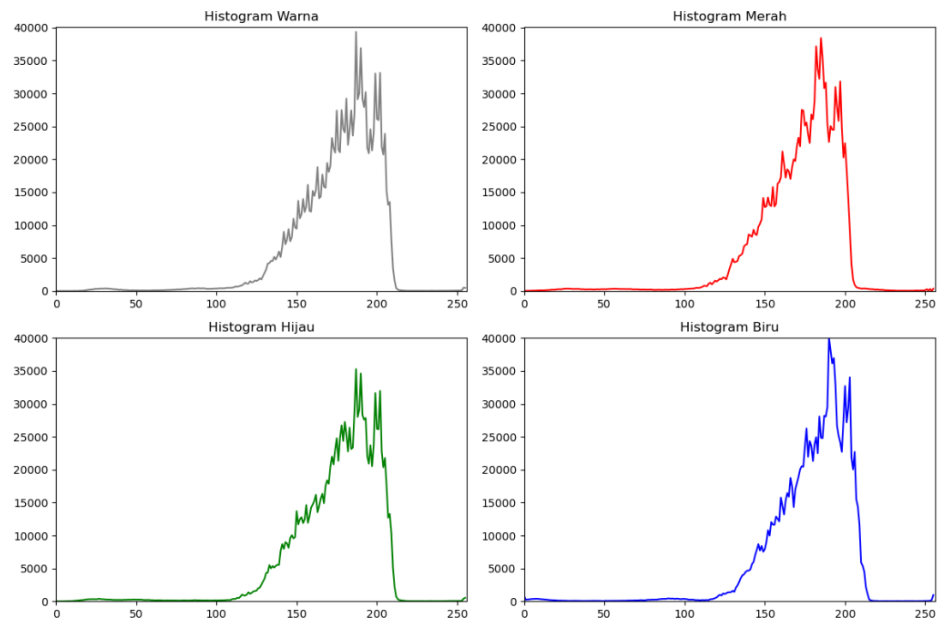
#### 3.1 Deteksi Warna Pada Citra

- Hasil Output



Penjelasan :

- **CITRA KONTRAS (RGB)**: Gambar asli menunjukkan teks dengan warna hijau (PRAMESTI), biru (ANGGREINI), dan merah (ROMADONA)
- **BIRU**: Hanya teks "ANGGREINI" yang terlihat terang (intensitas warna tinggi di channel itu) karena warnanya biru
- **MERAH**: Hanya teks "ROMADONA" yang terang karena mengandung warna merah.
- **HIJAU**: Hanya teks "PRAMESTI" yang terang karena warnanya hijau.
- Setiap channel menampilkan intensitas warna masing-masing dalam bentuk grayscale semakin terang, semakin tinggi kandungan warnanya

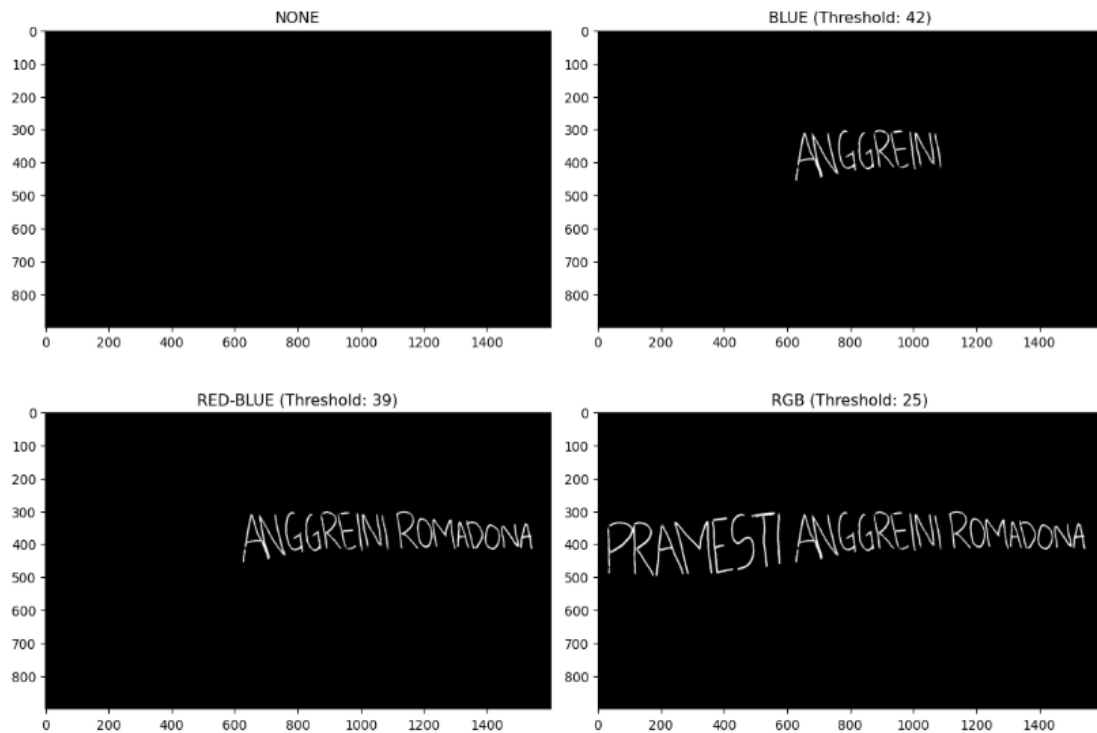


#### Penjelasan :

- Histogram Warna (Grayscale) – Kiri Atas
  - Warna: Abu-abu
  - ini adalah histogram dari versi grayscale gambar.
  - Sumbu X (0–255): intensitas piksel (0 = hitam, 255 = putih).
  - Sumbu Y: jumlah piksel yang memiliki intensitas tertentu.
  - Pola: Terlihat lonjakan besar di rentang intensitas 150–200, menandakan gambar ini cenderung terang (banyak piksel berwarna cerah).
- Histogram merah -kanan atas
  - Warna: Merah
  - Ini adalah histogram dari saluran warna merah.
  - Terlihat distribusi sangat tinggi antara 150–200, menunjukkan banyak piksel mengandung intensitas merah tinggi.
  - Sangat mirip dengan grayscale karena merah mungkin dominan dalam gambar tersebut.
- Histogram Hijau – Kiri Bawah
  - Warna: Hijau
  - Pola distribusi mirip seperti merah, tapi sedikit lebih lebar dan mungkin sedikit lebih rendah puncaknya.
  - Ini menunjukkan kontribusi saluran hijau cukup besar tapi tidak setinggi merah.
- Histogram Biru – Kanan Bawah
  - Warna: Biru
  - Distribusinya mirip, tapi intensitasnya sedikit lebih rendah dibandingkan merah dan hijau.
  - Masih menunjukkan puncak yang kuat pada area 150–200, tapi puncaknya lebih "ramping".

### 3.2 Mencari Ambang Batas Terkecil Samapai Dengan Terbesar

- Hasil output

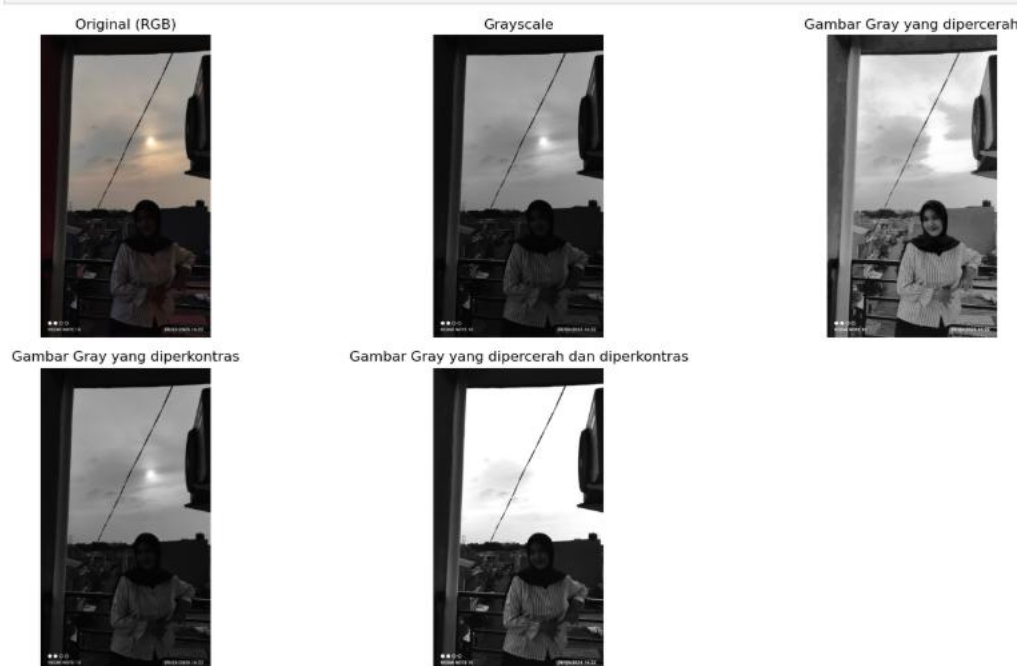


Penjelasan :

- **NONE:**  
Tidak ada warna yang dipilih, gambar sepenuhnya hitam.
- **BLUE (Threshold: 42):**  
Hanya tulisan "ANGGREINI" muncul terdeteksi sebagai warna biru.
- **RED-BLUE (Threshold: 39):**  
Tulisan "ANGGREINI ROMADONA" muncul gabungan dari biru dan merah.
- **RGB (Threshold: 25):**  
Semua tulisan "PRAMESTI ANGGREINI ROMADONA" muncul berhasil mendeteksi hijau, biru, dan merah.

### 3.3 Memperbaiki Gambar Backlight

- Hasil Output



Penjelasan :

- Original (RGB)
  - Ini adalah gambar asli yang dimuat dan dikonversi dari format BGR (yang digunakan oleh OpenCV) ke RGB (yang digunakan oleh matplotlib) agar warna tampil dengan benar.
  - Tidak ada perubahan apa pun di sini.
- Grayscale
  - Gambar ini adalah hasil konversi dari gambar RGB ke skala abu-abu (grayscale).
  - Warna dihilangkan sehingga hanya intensitas terang-gelap (brightness) yang dipertahankan.
- Gambar gray yang dipercerah (*Histogram Equalization*)
  - Ini hasil dari metode Histogram Equalization, yaitu teknik peningkatan kontras otomatis untuk gambar grayscale.
  - Tujuannya agar distribusi intensitas piksel lebih merata, membuat detail di area gelap dan terang lebih terlihat jelas.
- Gambar yang diperkontras
  - Ini adalah hasil kontras manual (contrast stretching).
  - Nilai piksel dikonversi agar intensitas terkecil menjadi 0 dan yang terbesar menjadi 255. Ini memperlebar rentang intensitas dan meningkatkan kontras secara manual.
- Gambar yang dipercerah dan diperkontras

- Ini hasil dari operasi dengan parameter  $\alpha = 1.5$  (meningkatkan kontras) dan  $\beta = 40$  (meningkatkan kecerahan).
- Secara visual, gambar tampak lebih terang dan lebih tajam, cocok untuk meningkatkan visibilitas objek pada gambar yang gelap atau kusam.

## **BAB IV**

### **PENUTUP**

Berdasarkan landasan teori dan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa pemahaman terhadap konsep dasar pengolahan citra digital sangat penting dalam proses analisis dan manipulasi gambar secara komputasional. Transformasi ruang warna dari RGB ke HSV terbukti lebih efektif dalam mendeteksi objek berdasarkan warna karena mampu memisahkan informasi warna (hue), kejenuhan (saturation), dan kecerahan (value), sehingga memudahkan proses segmentasi meskipun citra memiliki pencahayaan yang tidak seragam.

Penerapan teknik thresholding pada kanal HSV memungkinkan segmentasi objek warna tertentu dengan akurasi yang lebih baik. Penentuan nilai ambang batas yang tepat sangat krusial agar warna yang ditargetkan dapat diekstraksi secara optimal dari latar belakang. Hal ini diperkuat oleh penggunaan histogram sebagai alat bantu analisis distribusi warna dalam gambar.

Selain itu, pada kasus citra dengan kondisi pencahayaan yang buruk seperti backlight, teknik perbaikan citra melalui konversi ke grayscale dan peningkatan kontras mampu menampilkan detail yang semula tersembunyi. Peningkatan kualitas ini berkontribusi besar dalam membuat citra menjadi lebih informatif dan layak untuk dianalisis lebih lanjut.

Secara keseluruhan, praktikum ini memberikan wawasan praktis tentang bagaimana teori-teori pengolahan citra digital dapat diterapkan untuk menyelesaikan masalah nyata. Melalui bahasa pemrograman Python dan pustaka OpenCV, mahasiswa dapat memahami alur kerja pengolahan citra secara menyeluruh, mulai dari pembacaan gambar, transformasi ruang warna, segmentasi, hingga perbaikan kualitas visual. Praktikum ini juga menekankan pentingnya eksperimen dan penyesuaian parameter untuk mencapai hasil optimal dalam pemrosesan citra digital.

## DAFTAR PUSTAKA

- [1] F. Marpaung, F. Aulia, N. Suryani SKom, and R. Cyra Nabila SKom, *COMPUTER VISION DAN PENGOLAHAN CITRA DIGITAL*. [Online]. Available: [www.pustakaaksara.co.id](http://www.pustakaaksara.co.id)
- [2] M. Oni, B. Kanata, and D. Ratnasari, "MENENTUKAN LUAS OBJEK CITRA DENGAN TEKNIK SEGMENTASI BERDASARKAN WARNA PADA RUANG WARNA HSV Determining the Image Object Area Using Color-Based Segmentation Technique in HSV Color Space," 2021.
- [3] S. Bhahri, "Transformasi Citra Biner Menggunakan Metode Thresholding Dan Otsu Thresholding," 2018.
- [4] *MDP STUDENT CONFERENCE (MSC) 2022*.
- [5] I. Sulistiyowati, H. M. Ichsan, and I. Anshory, "Object Sorting Conveyor with Detection Color Using ESP-32 Camera Python Based on Open-CV," *JEECS (Journal of Electrical Engineering and Computer Sciences)*, vol. 9, no. 1, pp. 61–68, Jun. 2024, doi: 10.54732/jeecs.v9i1.7.