



Deteksi Warna Dasar Menggunakan Metode Thresholding HSV dengan OpenCV

Zidanul Akbar^{1*}, Asrul Suwondo², Rizky Ramadhan³, Abdul Halim Hasugian⁴

¹⁻⁴ Sains dan Teknologi, Ilmu Komputer, Universitas Islam Negeri Sumatera Utara, Indonesia

Email: ¹*1zidanulakbar69@gmail.com, ²2asrulsuwondo02@gmail.com, ³3rezky65@gmail.com,
⁴4abdulhalimhasugian@uinsu.ac.id

Alamat: Jl. William Iskandar Ps. V, Medan Estate, Kec. Percut Sei Tuan,
Kabupaten Deli Serdang, Sumatera Utara 20371
Korespondensi penulis: zidanulakbar69@gmail.com

Abstract Digital image processing is a rapidly developing branch of computer science and has many applications in everyday life. One of the fields that most often utilizes this technique is object detection and color identification in images and videos. This study specifically aims to implement the thresholding method in the HSV (Hue, Saturation, Value) color space to detect three basic colors, namely red, green, and blue, in digital images. The research process begins with uploading images using the Google Colab platform, a cloud-based computing environment that makes it easy for users to run Python programs without requiring additional software installation. After the image is uploaded, the next step is to convert it from the RGB (Red, Green, Blue) color space to the HSV color space. This conversion is important because the HSV color space is more suitable for use in the color segmentation process. The Hue value represents the type of color, Saturation shows the level of saturation, while Value describes the level of brightness. Once the image is in the HSV color space, the next step is to determine the HSV value range for each basic color. This range is determined based on experimental results and references from related literature. Using this range, masking is performed to extract the appropriate pixels so that only the red, green, or blue portions of the image are visible, while the other colors are reduced. The results show that the thresholding method in the HSV color space is capable of detecting primary colors with a good level of visual accuracy, especially in simple images with contrasting backgrounds. The implementation of this program is relatively lightweight, easy to run directly in Google Colab, and does not require high-spec hardware. Therefore, this method is very suitable for use as basic learning material for digital image processing, both for students and novice researchers.

Keywords: Color Detection, HSV, Image Processing, OpenCV, Thresholding

Abstrak Pengolahan citra digital merupakan salah satu cabang ilmu komputer yang berkembang pesat dan memiliki banyak aplikasi dalam kehidupan sehari-hari. Salah satu bidang yang paling sering memanfaatkan teknik ini adalah deteksi objek dan identifikasi warna pada gambar maupun video. Penelitian ini secara khusus bertujuan untuk mengimplementasikan metode thresholding pada ruang warna HSV (Hue, Saturation, Value) guna mendeteksi tiga warna dasar, yaitu merah, hijau, dan biru, pada citra digital. Proses penelitian diawali dengan pengunggahan citra menggunakan platform Google Colab, sebuah lingkungan komputasi berbasis cloud yang memudahkan pengguna dalam menjalankan program Python tanpa memerlukan instalasi perangkat lunak tambahan. Setelah citra diunggah, tahap berikutnya adalah konversi dari ruang warna RGB (Red, Green, Blue) ke ruang warna HSV. Konversi ini penting dilakukan karena ruang warna HSV lebih sesuai digunakan dalam proses segmentasi warna. Nilai Hue merepresentasikan jenis warna, Saturation menunjukkan tingkat kejenuhan, sedangkan Value menggambarkan tingkat kecerahan. Setelah citra berada dalam ruang warna HSV, langkah selanjutnya adalah menentukan rentang nilai HSV untuk masing-masing warna dasar. Rentang ini ditentukan berdasarkan hasil percobaan serta acuan dari literatur terkait. Dengan rentang tersebut, dilakukan masking untuk mengekstraksi piksel yang sesuai sehingga hanya bagian citra dengan warna merah, hijau, atau biru yang terlihat, sedangkan warna lainnya tereduksi. Hasil penelitian menunjukkan bahwa metode thresholding pada ruang warna HSV mampu mendeteksi warna dasar dengan tingkat akurasi visual yang baik, khususnya pada citra sederhana dengan latar belakang kontras. Implementasi program ini relatif ringan, mudah dijalankan secara langsung di Google Colab, serta tidak membutuhkan perangkat keras dengan spesifikasi tinggi. Dengan demikian, metode ini sangat sesuai digunakan sebagai materi pembelajaran dasar pengolahan citra digital, baik bagi mahasiswa maupun peneliti pemula.

Kata kunci: Deteksi Warna, HSV, OpenCV, Pengolahan Citra, Thresholding

1. PENDAHULUAN

Pengolahan citra digital telah menjadi salah satu bidang yang berkembang pesat dalam ilmu komputer dan teknik informatika (Sulistiyowati et al., 2024). Perkembangan teknologi ini memungkinkan komputer untuk melakukan analisis visual terhadap gambar atau video, sehingga banyak dimanfaatkan di berbagai sektor seperti keamanan, industri, pertanian, hingga pendidikan. Salah satu penerapan yang relatif sederhana namun bermanfaat adalah deteksi warna pada citra digital (Fitriyah et al., 2021). Deteksi warna dapat digunakan untuk berbagai tujuan, misalnya dalam sistem penyortiran otomatis, identifikasi objek, maupun pengolahan sinyal visual untuk robotika. Keunggulan metode deteksi warna adalah prosesnya yang relatif cepat, mudah diimplementasikan, dan tidak memerlukan model pembelajaran yang kompleks, sehingga dapat dijadikan proyek awal untuk pembelajaran pengolahan citra (Wibowo, 2020).

Salah satu tantangan dalam deteksi warna adalah pemilihan model ruang warna yang tepat. Secara default, banyak gambar digital direpresentasikan dalam ruang warna RGB (Red, Green, Blue) (Erza et al., 2022). Namun, pada praktiknya, ruang warna RGB seringkali kurang efektif untuk proses segmentasi berdasarkan warna, terutama karena ketiga komponen warnanya saling bergantung pada intensitas cahaya (Putri et al., 2018). Oleh karena itu, ruang warna HSV (Hue, Saturation, Value) lebih banyak digunakan pada kasus deteksi warna, karena mampu memisahkan informasi warna (hue) dari tingkat kejenuhan (saturation) dan kecerahan (value). Pemisahan ini memudahkan proses thresholding untuk menentukan batas nilai yang digunakan dalam mendeteksi suatu warna tertentu (Sari et al., 2020).

Metode thresholding dalam ruang warna HSV dilakukan dengan menentukan batas bawah (lower bound) dan batas atas (upper bound) nilai HSV untuk setiap warna yang ingin dideteksi. Sebagai contoh, warna merah dapat didefinisikan dalam dua rentang hue yang berbeda, karena nilai hue bersifat melingkar (0° – 180° pada OpenCV) (Amrozi et al., 2024). Begitu pula dengan warna hijau dan biru, masing-masing memiliki rentang HSV tertentu yang dapat diatur berdasarkan percobaan atau acuan dari literatur (Suradi et al., 2023). Dengan menggunakan metode ini, setiap piksel yang berada dalam rentang tersebut akan diberi nilai “mask” atau biner, sehingga dapat dipisahkan dari latar belakang citra (Mulyana et al., 2024).

Implementasi metode ini menggunakan bahasa pemrograman Python dengan pustaka OpenCV. OpenCV menyediakan fungsi bawaan untuk konversi warna dari RGB ke HSV dan untuk membuat mask berdasarkan rentang nilai HSV yang telah ditentukan (Purwoko et al., 2021). Selain itu, integrasi OpenCV dengan Google Colab membuat proses pengujian menjadi lebih fleksibel dan praktis, karena pengguna tidak perlu melakukan instalasi manual pada

komputer lokal. Cukup dengan mengunggah gambar ke Colab, proses deteksi warna dapat dilakukan secara langsung dan hasilnya dapat divisualisasikan dengan pustaka Matplotlib (Ahadi et al., 2024).

Penelitian ini difokuskan pada deteksi tiga warna dasar, yaitu merah, hijau, dan biru, karena warna-warna ini merupakan komponen utama dalam model RGB dan banyak digunakan sebagai indikator dalam berbagai aplikasi sederhana (Nur Rochim et al., 2024). Hasil yang diharapkan dari penelitian ini adalah sebuah implementasi kode yang mudah dijalankan, ringan, dan dapat digunakan sebagai bahan ajar pengenalan pengolahan citra digital (Fajar Wibowo et al., 2024). Dengan metode yang sederhana ini, mahasiswa atau peneliti pemula dapat memahami konsep dasar representasi warna, konversi ruang warna, dan proses segmentasi citra berbasis warna, yang nantinya dapat dikembangkan lebih lanjut untuk aplikasi yang lebih kompleks (Setiyani et al., 2022).

2. METODE PENELITIAN

Metode penelitian ini meliputi beberapa tahap sebagai berikut: 1). Persiapan Data: Dua gambar apel digunakan sebagai sampel: red apple dan green apple. Kedua gambar memiliki latar belakang putih polos untuk memudahkan segmentasi. 2). Pengunggahan dan Pembacaan Gambar: Gambar diunggah ke Google Colab menggunakan modul files dari Google Colab. OpenCV digunakan untuk membaca gambar dalam format BGR dan kemudian dikonversi ke RGB untuk keperluan visualisasi, 3). Konversi Ruang Warna: Gambar yang telah dibaca dikonversi dari ruang warna RGB ke HSV menggunakan fungsi `cv2.cvtColor()`. HSV dipilih karena memisahkan informasi warna (hue) dari tingkat kejenuhan (saturation) dan kecerahan (value), sehingga mempermudah proses thresholding, 4). Penentuan Rentang Warna (Threshold) Merah: Hue 0–10 dan 160–179, Saturation 100–255, Value 100–255 Hijau: Hue 40–80, Saturation 40–255, Value 40–255 Biru: Hue 100–140, Saturation 150–255, Value 0–255 Rentang ini digunakan dalam fungsi `cv2.inRange()` untuk membuat mask biner, 5). Pembuatan Mask dan Visualisasi: Mask biner untuk masing-masing warna dihasilkan, di mana piksel dalam rentang warna tertentu bernilai putih (255) dan sisanya hitam (0). Hasil deteksi divisualisasikan menggunakan Matplotlib, menampilkan gambar asli dan hasil deteksi untuk merah, hijau, dan biru (Islami, 2021).

3. HASIL DAN PEMBAHASAN

Pengujian dilakukan menggunakan dua citra sampel, yaitu green apple dan red apple. Proses deteksi warna dilakukan menggunakan bahasa pemrograman Python di Google Colab dengan pustaka OpenCV. Kode program yang digunakan untuk melakukan deteksi warna dasar merah, hijau, dan biru adalah sebagai berikut:

```
# Install library yang dibutuhkan
!pip install opencv-python matplotlib

import cv2
import numpy as np
import matplotlib.pyplot as plt

# Upload gambar
from google.colab import files
uploaded = files.upload()

# Ambil nama file yang diunggah
image_path = list(uploaded.keys())[0]

# Baca gambar dan konversi ke RGB
image = cv2.imread(image_path)
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Konversi RGB -> HSV
image_hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

# Rentang warna Merah
lower_red1 = np.array([0, 100, 100])
upper_red1 = np.array([10, 255, 255])
mask_red1 = cv2.inRange(image_hsv, lower_red1, upper_red1)

lower_red2 = np.array([160, 100, 100])
upper_red2 = np.array([179, 255, 255])
```

```
mask_red2 = cv2.inRange(image_hsv, lower_red2, upper_red2)
mask_red = mask_red1 | mask_red2

# Rentang warna Hijau
lower_green = np.array([40, 40, 40])
upper_green = np.array([80, 255, 255])
mask_green = cv2.inRange(image_hsv, lower_green, upper_green)

# Rentang warna Biru
lower_blue = np.array([100, 150, 0])
upper_blue = np.array([140, 255, 255])
mask_blue = cv2.inRange(image_hsv, lower_blue, upper_blue)

# Visualisasi hasil
fig, axs = plt.subplots(2, 2, figsize=(10, 8))
axs[0, 0].imshow(image_rgb)
axs[0, 0].set_title("Gambar Asli")
axs[0, 0].axis("off")

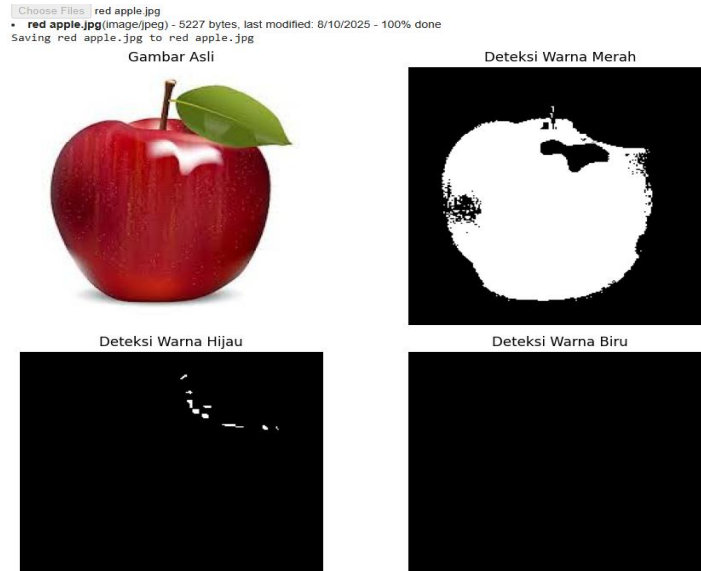
axs[0, 1].imshow(mask_red, cmap='gray')
axs[0, 1].set_title("Deteksi Warna Merah")
axs[0, 1].axis("off")

axs[1, 0].imshow(mask_green, cmap='gray')
axs[1, 0].set_title("Deteksi Warna Hijau")
axs[1, 0].axis("off")

axs[1, 1].imshow(mask_blue, cmap='gray')
axs[1, 1].set_title("Deteksi Warna Biru")
axs[1, 1].axis("off")

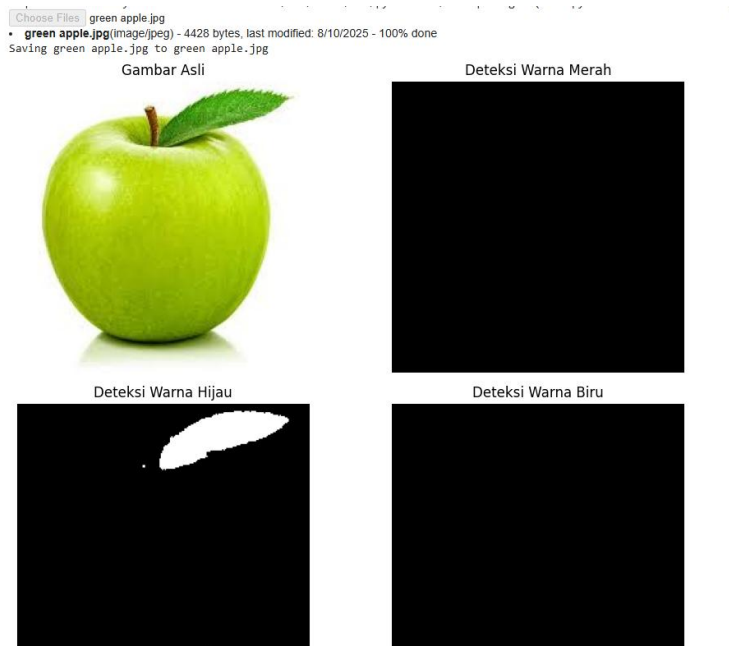
plt.tight_layout()
plt.show()
```

Kode program yang digunakan telah ditampilkan sebelumnya dan berfungsi untuk: 1). Mengunggah citra dari perangkat pengguna, 2). Mengonversi citra dari RGB ke HSV, 3). Membuat mask biner berdasarkan rentang HSV untuk warna merah, hijau, dan biru, 4). Menampilkan hasil deteksi setiap warna dalam bentuk citra biner (hitam-putih).



Gambar 1 Output red apple

Pada citra red apple, hasil Deteksi Warna Merah menunjukkan mask putih yang menutupi hampir seluruh permukaan buah. Hal ini menunjukkan bahwa hue (0–10 dan 160–179) berhasil menangkap rona merah dominan pada objek. Pada Deteksi Warna Hijau, hanya sedikit area daun yang terdeteksi, sesuai dengan keberadaan daun berwarna hijau pada citra asli. Untuk Deteksi Warna Biru, tidak ada area yang terdeteksi, mask tetap hitam sepenuhnya.



Gambar 2 Output green apple

Berdasarkan hasil eksekusi program, pada citra green apple bagian Deteksi Warna Hijau menghasilkan mask putih yang hampir menutupi seluruh permukaan apel, menunjukkan bahwa warna hijau terdeteksi dominan. Pada Deteksi Warna Merah dan Deteksi Warna Biru, mask yang dihasilkan berwarna hitam penuh, menandakan tidak ada area signifikan yang masuk ke dalam rentang HSV untuk warna merah atau biru. Keberhasilan ini disebabkan oleh rentang hue (40–80) yang digunakan untuk mendeteksi warna hijau sesuai dengan komposisi warna apel tersebut.

Hasil deteksi untuk kedua citra memperlihatkan bahwa metode thresholding HSV efektif dalam membedakan warna dominan dengan latar belakang kontras putih. Rentang HSV yang tepat mampu memisahkan objek dari latar belakang tanpa kesalahan segmentasi signifikan. Namun, pada bagian tepian objek terlihat sedikit area yang tidak terdeteksi secara penuh akibat pengaruh pencahayaan dan gradasi warna. Misalnya, pada apel merah terdapat area refleksi cahaya yang memengaruhi nilai hue dan saturation sehingga tidak masuk ke rentang yang didefinisikan.

Tabel 1. Hasil Deteksi Warna pada Sampel Gambar

Gambar	Deteksi Warna Merah	Deteksi Warna Hijau	Deteksi Warna Biru
Green Apple	Tidak terdeteksi	Teridentifikasi pada area buah	Tidak terdeteksi
Red Apple	Teridentifikasi pada area buah	Teridentifikasi sedikit pada area daun	Tidak terdeteksi

Secara keseluruhan, hasil ini membuktikan bahwa metode thresholding HSV dapat menjadi teknik sederhana namun efektif untuk mendeteksi warna dasar pada citra digital, terutama bila latar belakang memiliki kontras yang tinggi terhadap objek. Implementasi di Google Colab membuat metode ini praktis dijalankan tanpa konfigurasi rumit, sehingga cocok untuk pembelajaran dasar pengolahan citra.

4. KESIMPULAN

Metode thresholding pada ruang warna HSV (Hue, Saturation, Value) terbukti efektif untuk mendeteksi warna dasar—merah, hijau, dan biru—dengan cukup akurat di citra dengan latar belakang kontras seperti red apple dan green apple. Apel merah berhasil terdeteksi dominan sebagai mask merah yang hampir menutupi seluruh buah; sedangkan apel hijau menampilkan mask hijau dominan yang serupa. Area daun hijau pada apel merah juga secara tepat terdeteksi. Meskipun begitu, terdapat sedikit keterbatasan pada bagian tepi objek akibat variasi pencahayaan dan refleksi yang memengaruhi konsistensi nilai HSV. Implementasi di

Google Colab dengan Python dan OpenCV sangat ringan dijalankan dan ideal sebagai materi pembelajaran awal dalam pengolahan citra digital. Untuk aplikasi lanjutan, metode thresholding adaptif atau teknik segmentasi berbasis pembelajaran mesin bisa dipertimbangkan untuk meningkatkan ketahanan terhadap kondisi pencahayaan yang bervariasi.

DAFTAR PUSTAKA

- Ahadi, A. H., Gustina, G., Syawal, M. F., Aminuddin, F. H., & Anzari, Y. (2024). Implementasi sistem pendeteksi warna objek dengan OpenCV-Python. *SENTRI: Jurnal Riset Ilmiah*, 3(7), 3573–3578. <https://doi.org/10.55681/sentri.v3i7.3185>
- Amrozi, M. A., Figo, S. W. D., & Wahyusari, R. (2024). Perbandingan segmentasi ruang warna HSV dan YCbCr untuk deteksi objek. *Infomatek*, 26(2), 217–222. <https://doi.org/10.23969/infomatek.v26i2.19025>
- Erza, F., Fitriyah, H., & Setiawan, E. (2022). Sistem object tracking pada quadcopter menggunakan segmentasi citra dengan deteksi warna HSV dan metode regresi linier berbasis Raspberry Pi. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 9(7), 1733–1740. <https://doi.org/10.25126/jtiik.2022976808>
- Fajar Wibowo, C., Izzul Haq, F., Maulana Ansaris, F., & Agustin, S. (2024). Analisis persentase warna blue gem pada skin case hardened di Counter Strike 2 menggunakan OpenCV dan Python. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 7241–7247. <https://doi.org/10.36040/jati.v8i4.10140>
- Fitriyah, H., Maulana, R., Komputer, F. I., Malang, U. B., & Korespondensi, P. (2021). Weed detection in HSV color-space based on shape features using. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(5), 929–938. <https://doi.org/10.25126/jtiik.202184719>
- Islami, F. (2021). Implementation of HSV-based thresholding method for iris detection. *Journal of Computer Networks, Architecture, and High-Performance Computing*, 3(1), 98–104. <https://doi.org/10.47709/cnahpc.v3i1.939>
- Mulyana, D. I., Arinal, V., & Akbarulloh, F. (2024). Efektivitas penggunaan ruang warna HSV untuk klasifikasi daging sapi segar dan busuk dalam industri pangan. *Jurnal JTik (Jurnal Teknologi Informasi dan Komunikasi)*, 9(1), 244–254. <https://doi.org/10.35870/jtik.v9i1.3129>
- Nur Rochim, F., Desky Sompie, G., Mua'mmar, Imam Saputra, R., & Rosyani, P. (2024). Perancangan sistem deteksi warna real-time menggunakan metode Gaussian blur dan ruang warna HSV. *Biner: Jurnal Ilmu Komputer, Teknik dan Multimedia*, 2(2), 178–183. <https://journal.mediapublikasi.id/index.php/Biner>
- Purwoko, J. D. G., Pratama, J. Y., & Hartanto, A. D. (2021). Mendeteksi objek berdasarkan warna menggunakan HSV color space secara realtime. *Intechno Journal (Information Technology Journal)*, 3(2), 44–48. <https://doi.org/10.24076/intechnojournal.2021v3i2.1554>
- Putri, A. S., Setyawan, G. E., & Tibyani. (2018). Sistem deteksi warna pada quadcopter AR.Drone menggunakan metode color filtering hue saturation and value (HSV). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(9), 3202–3207. <http://j-tiik.ub.ac.id>

- Sari, D. A. L., Mulyadi, A., & Pratama, A. (2020). Deteksi objek berwarna real-time berdasarkan visualisasi webcam. *Zetroem: Jurnal Ilmiah Mahasiswa Teknik Elektro*, 2(1), 21–24. <https://ejournal.unibabwi.ac.id/index.php/Zetroem/article/view/1336>
- Setiyani, A., Maison, M., & Fuady, S. (2022). Perancangan sistem deteksi objek bola dengan metode coloring HSV berbasis VB.Net untuk robot sepak bola beroda. *Jurnal Engineering*, 4(2), 67–73. <https://doi.org/10.22437/jurnalengineering.v4i2.19835>
- Sulistiyowati, I., Maulana Ichsan, H., & Anshory, I. (2024). Konveyor penyortir objek dengan deteksi warna menggunakan kamera ESP-32 berbasis OpenCV Python. *Prosiding TAU SNARS-TEK Seminar Nasional Rekayasa dan Teknologi*, 4(1), 35–41. <https://doi.org/10.47970/snarstek.v2i1.711>
- Suradi, A. A. M., Rasyid, M. F., Mushaf, M., & Rizal, M. (2023). Deteksi tingkat kematangan buah apel menggunakan segmentasi ruang warna HSV. *Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, 12(1), 19–26.
- Wibowo, J. S. (2020). Deteksi dan klasifikasi citra berdasarkan warna kulit menggunakan HSV. *Jurnal Teknologi Informasi DINAMIK*, 16(2), 118–123.