

Penerapan Model Warna HSV untuk Deteksi Objek Berbasis Warna secara Real-Time pada Kondisi Pencahayaan Dinamis

Anggun Alhidayah¹, Wingga Juina Rohizi², Febri Dristyan^{3*}

^{1,2,3}Teknologi Rekayasa Perangkat Lunak, Politeknik Jambi

¹anggunalhidayahh@gmail.com, ²whinggajuniarohizi@gmail.com
³fdristyan@gmail.com

Abstrak

Sistem pengenalan warna objek secara *real-time* merupakan teknologi yang memanfaatkan kamera dan pemrosesan citra digital untuk mendeteksi warna objek secara langsung. Penelitian ini bertujuan untuk membangun sistem berbasis Python dengan library OpenCV yang mampu mengenali warna objek yang tertangkap kamera secara akurat dan efisien. Masalah utama yang dihadapi adalah variasi pencahayaan dan kualitas kamera yang dapat mempengaruhi akurasi pengenalan warna. Solusi yang ditawarkan adalah penggunaan konversi ruang warna dari BGR ke HSV untuk meningkatkan ketahanan sistem terhadap variasi cahaya. Pengujian dilakukan terhadap lima warna dasar (merah, hijau, biru, kuning, dan ungu) dengan hasil akurasi rata-rata mencapai 92%.

Kata Kunci: Pengenalan warna, real-time, OpenCV, HSV, Python

Abstract

Real-time object color recognition is a technology that utilizes cameras and digital image processing to directly detect the color of objects. This study aims to develop a Python-based system using the OpenCV library capable of accurately and efficiently recognizing the color of objects captured by a camera. The main problem encountered is the variation in lighting and camera quality, which can affect color recognition accuracy. The proposed solution is the conversion of color space from BGR to HSV to improve the system's robustness to light variation. Tests were conducted on five basic colors (red, green, blue, yellow, and purple) with an average accuracy result of 92%.

Keywords: Color recognition, real-time, OpenCV, HSV, Python

1. PENDAHULUAN

Perkembangan teknologi *image processing* telah memberikan dampak signifikan di berbagai bidang, mulai dari kesehatan, pertanian, keamanan, hingga industri percetakan. Salah satu tantangan utama dalam pengolahan citra digital adalah bagaimana mengekstraksi informasi visual secara efisien di bawah berbagai kondisi pencahayaan. Dalam banyak kasus, sistem deteksi objek sering mengalami degradasi performa akibat perubahan pencahayaan yang dinamis, terutama dalam sistem *real-time*[1].

Untuk mengatasi masalah tersebut, diperlukan pendekatan pemodelan warna yang mampu memisahkan informasi warna dari intensitas cahaya. Salah satu solusi yang ditawarkan adalah penggunaan model warna HSV (*Hue, Saturation, Value*). Model ini memungkinkan sistem mendeteksi objek berbasis warna dengan lebih stabil karena memisahkan *hue* (warna) dari *value* (pencahayaan), sehingga mampu meminimalkan gangguan dari fluktuasi pencahayaan lingkungan.

Model HSV telah banyak digunakan dalam berbagai penelitian pengolahan citra digital. Menurut Ahadi et al. [2], HSV efektif mengenali objek dengan warna seperti coklat, kuning, hijau, biru, hitam, dan putih, serta mampu mengurangi pengaruh intensitas cahaya dalam lingkungan terbuka.

Di ranah aplikasi mobile, penelitian oleh Amrullah et al. [3] mengembangkan sistem deteksi warna *real-time* berbasis Android menggunakan model RGB dan algoritma *Midpoint*. Aplikasi tersebut

mampu mendeteksi warna hingga jarak 10 meter dengan akurasi baik pada berbagai kondisi pencahayaan.

Selain itu, pendekatan berbasis *deep learning*, khususnya *Convolutional Neural Network* (CNN), juga menjadi metode populer dalam klasifikasi objek berdasarkan fitur warna. Penelitian oleh Prasetyo dan Wiliani [3] menunjukkan bahwa konfigurasi CNN dengan satu *layer convolution*, satu *max pooling*, dan satu *fully connected layer* menghasilkan akurasi deteksi hingga 92%. Namun, penggunaan lebih banyak *layer* justru menurunkan performa model secara signifikan.

Dalam konteks industri percetakan, sistem digital berbasis RGB memiliki keterbatasan dalam menyajikan warna yang sesuai dengan hasil cetak. Untuk itu, Pratama et al. [4] mengembangkan aplikasi berbasis Python[5] yang mampu mengonversi warna dari RGB ke CMYK menggunakan standar warna Adobe RGB. Hasilnya menunjukkan bahwa visualisasi komponen CMYK dalam bentuk *greyscale* mampu mendekati proses cetak sesungguhnya.

Selain HSV, ruang warna lain seperti HSL dan YCbCr juga digunakan dalam pengembangan sistem pendeteksian objek. Sugandi [6] menunjukkan bahwa model HSL menghasilkan akurasi deteksi dan pelacakan objek tertinggi dibandingkan RGB dan YCbCr.

Berdasarkan hasil studi terdahulu, ditemukan *gap analysis* terkait pemilihan model warna yang optimal untuk sistem deteksi objek berbasis warna secara *real-time*. Model RGB memiliki sensitivitas tinggi terhadap pencahayaan, CMYK lebih cocok untuk percetakan, sementara HCL meskipun sesuai dengan persepsi manusia, kurang umum digunakan dalam sistem *real-time*.

Oleh karena itu, penelitian ini memilih model warna HSV karena kemampuannya memisahkan komponen warna dan pencahayaan, serta menunjukkan stabilitas dalam berbagai kondisi cahaya. Tujuan dari penelitian ini adalah untuk membangun sistem deteksi objek berbasis warna secara *real-time* dengan memanfaatkan model HSV, guna meningkatkan keakuratan deteksi objek pada lingkungan dengan pencahayaan dinamis. Diharapkan sistem ini dapat diterapkan dalam berbagai kebutuhan praktis seperti sistem keamanan otomatis, pelacakan objek bergerak, dan pengendalian kualitas berbasis visual.

2. METODE

Tahapan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimental, dengan fokus pada pengembangan sistem pendeteksian warna secara *real-time* menggunakan Python[7] dan OpenCV. Metode yang digunakan bersifat iteratif, di mana setiap tahapan dapat dilakukan berulang sesuai kebutuhan pengujian dan penyempurnaan sistem. Tahapan penelitian meliputi:

a. Analisis Masalah dan Kebutuhan

Tahap awal ini bertujuan mengidentifikasi kebutuhan pengguna, tujuan sistem, serta permasalahan teknis yang mungkin muncul, terutama dalam mengenali warna secara akurat pada berbagai kondisi pencahayaan. Spesifikasi teknis disusun berdasarkan kebutuhan sistem deteksi warna *real-time*.

b. Perancangan Sistem

Perancangan mencakup skema alur proses pendeteksian warna, desain antarmuka pengguna, serta struktur data yang digunakan. Sistem dirancang untuk mengolah citra dari kamera, mengonversinya dari format BGR ke HSV, lalu mengklasifikasikan warna berdasarkan rentang nilai HSV tertentu. Selain itu, dirancang pula mekanisme penyimpanan hasil deteksi ke dalam database.

c. Implementasi Sistem

Tahapan ini melibatkan penulisan kode program menggunakan bahasa Python dan pustaka OpenCV. Kamera diakses secara *real-time*, dan tiap frame yang ditangkap akan dikonversi ke

HSV, lalu dianalisis untuk mendeteksi warna dominan pada titik tertentu. Hasil deteksi ditampilkan pada layar dan disimpan ke database jika memenuhi kriteria.

d. Pengujian dan Evaluasi

Sistem diuji menggunakan sepuluh warna utama, yaitu: putih, hitam, merah, oranye, kuning, hijau, cyan, biru, ungu, dan merah muda. Pengujian dilakukan dengan menggunakan berbagai objek fisik berwarna, dalam beragam kondisi pencahayaan, seperti terang, redup, dan cahaya alami.

Untuk setiap warna, dilakukan 50 kali pengujian. Hasil deteksi warna oleh sistem dibandingkan dengan warna sebenarnya dari objek. Evaluasi dilakukan dengan menghitung tingkat akurasi deteksi dalam bentuk persentase, serta mencatat kondisi terbaik dan kendala deteksi yang ditemui untuk masing-masing warna.

Metode Pendeteksian Warna

Proses deteksi warna dilakukan dengan pendekatan segmentasi berdasarkan ruang warna HSV. Sistem membaca *frame* dari kamera, kemudian dilakukan konversi warna dari BGR (format default pada OpenCV) ke HSV menggunakan fungsi:

```
pixel_hsv = cv2.cvtColor(np.uint8([pixel_bgr]), cv2.COLOR_BGR2HSV)[0][0]
```

Setelah gambar dikonversi ke HSV, dilakukan pemfilteran warna dengan menggunakan nilai ambang batas bawah dan atas untuk masing-masing warna target.Code:

```
import cv2
import numpy as np
import mysql.connector
from datetime import datetime
from collections import deque
```

```
# Fungsi untuk menentukan nama warna berdasarkan HSV
```

```
def get_color_name(h, s, v):
```

```
    if v < 50:
```

```
        return "HITAM"
```

```
    elif s < 50 and v > 200:
```

```
        return "PUTIH"
```

```
    elif h < 10 or h >= 170:
```

```
        return "MERAH"
```

```
    elif 10 <= h < 25:
```

```
        return "ORANYE"
```

```
    elif 25 <= h < 35:
```

```
        return "KUNING"
```

```
    elif 35 <= h < 85:
```

```
        return "HIJAU"
```

```
    elif 85 <= h < 100:
```

```
        return "CYAN"
```

```
    elif 100 <= h < 130:
```

```
        return "BIRU"
```

```
    elif 130 <= h < 160:
```

```
        return "UNGU"
```

```
    elif 160 <= h < 170:
```

```
        return "MERAH MUDA"
```

```
    else:
```

```
        return "TIDAK DIKENAL"

# Konversi HSV ke BGR
def hsv_to_bgr(h, s, v):
    return cv2.cvtColor(np.uint8([[[h, s, v]]]), cv2.COLOR_HSV2BGR)[0][0]

# Koneksi ke database
try:
    db = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        database="deteksi_warna"
    )
    cursor = db.cursor()
    print("✓ Koneksi database sukses.")
except Exception as e:
    print("✗ Gagal koneksi:", e)
    exit()

# Kamera
cap = cv2.VideoCapture(0)
terakhir_disimpan = ""
deteksi_sebelumnya = ""
counter_sama = 0
total_disimpan = 0
recent_detections = deque(maxlen=10)

while True:
    ret, frame = cap.read()
    if not ret:
        break

    h_frame, w_frame = frame.shape[:2]
    cx, cy = w_frame // 2, h_frame // 2

    # Ambil pixel tengah dan konversi ke HSV
    pixel_bgr = frame[cy, cx]
    pixel_hsv = cv2.cvtColor(np.uint8([pixel_bgr]), cv2.COLOR_BGR2HSV)[0][0]
    h, s, v = pixel_hsv
    warna = get_color_name(h, s, v)

    # Kotak warna dan teks
    warna_bgr = hsv_to_bgr(h, s, v)
    cv2.rectangle(frame, (10, 10), (160, 80), warna_bgr.tolist(), -1)
    cv2.putText(frame, f"{warna}", (20, 50),
                cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 2)
```

```
# Tampilkan nilai HSV (DEBUG)
cv2.putText(frame, f"HSV: {h},{s},{v}", (10, h_frame - 50),
             cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 1)

cv2.putText(frame, f"Tersimpan: {total_disimpan}", (10, h_frame - 20),
            cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 1)

# Titik tengah
cv2.circle(frame, (cx, cy), 5, (255, 255, 255), -1)
cv2.imshow("Deteksi Warna", frame)

# Logika penyimpanan
if warna == deteksi_sebelumnya:
    counter_sama += 1
else:
    counter_sama = 0
    deteksi_sebelumnya = warna

if counter_sama >= 25 and warna != "TIDAK DIKENAL":
    hsv_str = f"{h},{s},{v}"
    waktu = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

    if warna != terakhir_disimpan and warna not in recent_detections:
        cursor.execute("INSERT INTO data_warna (waktu, nama_warna, hsv) VALUES (%s,
%s, %s)",
                        (waktu, warna, hsv_str))
        db.commit()
        print(f"✓ Disimpan: {warna} ({hsv_str})")
        terakhir_disimpan = warna
        recent_detections.append(warna)
        total_disimpan += 1
        counter_sama = 0

if cv2.waitKey(1) & 0xFF == ord('q'):
    break

cap.release()
cv2.destroyAllWindows()
cursor.close()
db.close()
```

Setelah segmentasi, dilakukan *morphological operation* seperti *dilation* untuk memperjelas hasil deteksi dan mengurangi noise. Objek yang terdeteksi ditandai dengan kotak dan diberi label nama warnanya secara *real-time* di layar.

Tabel 1. Rentang Nilai HSV untuk Masing-Masing Warna

Warna	Hue (H)	Saturation (S)	Value (V)
Putih	-	< 50	> 200
Hitam	-	-	< 50
Merah	0-10, 170-179	100-255	100-255
Oranye	10-25	100-255	100-255
Kuning	25-35	100-255	100-255
Hijau	35-85	100-255	100-255
Cyan	85-100	100-255	100-255
Biru	100-130	100-255	100-255
Ungu	130-160	100-255	100-255
Merah Muda	160-170	100-255	100-255

Keterangan: Nilai H berada dalam rentang 0-179 pada OpenCV, bukan 0-360 seperti pada teori warna standar. Oleh karena itu, semua rentang hue disesuaikan.

Evaluasi Sistem

Evaluasi dilakukan dengan cara mencocokkan hasil deteksi warna oleh sistem dengan warna objek sebenarnya. Pengujian dilakukan sebanyak 50 kali untuk setiap warna pada kondisi pencahayaan berbeda-beda. Hasil evaluasi akan dibandingkan dalam bentuk presentase akurasi.

3. HASIL DAN PEMBAHASAN

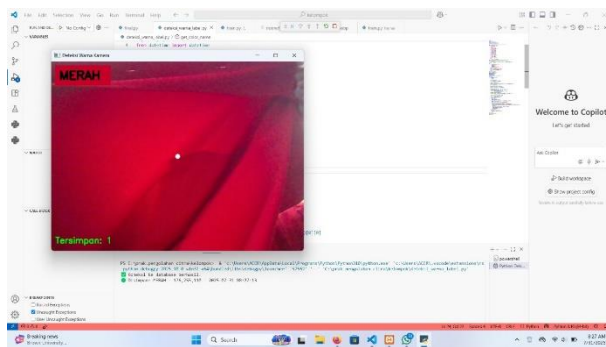
Setelah sistem dikembangkan dengan pendekatan deteksi warna menggunakan model warna HSV, dilakukan pengujian terhadap beberapa objek berwarna dalam kondisi pencahayaan yang berbeda. Proses deteksi dilakukan secara real-time menggunakan kamera webcam, dengan rentang nilai Hue, Saturation, dan Value yang telah ditentukan sebelumnya untuk masing-masing warna target.

Tabel 2. hasil pengujian akurasi dan kondisi optimal untuk setiap warna

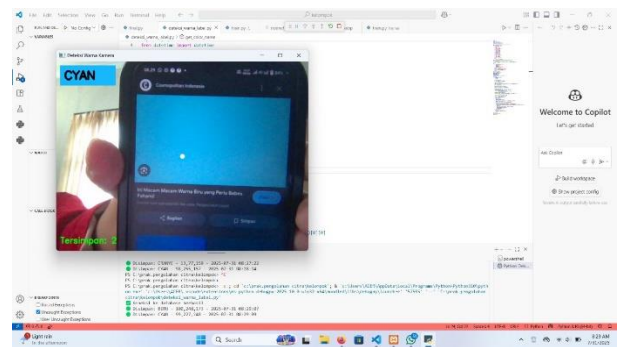
Warna	Akurasi Deteksi	Kondisi Terbaik	Kendala Deteksi
Putih	88%	Pencahayaan terang	Redup, latar sewarna
Hitam	93%	Stabil di semua kondisi	Sedikit noise pada gelap
Merah	85%	Terang & kontras tinggi	Latar mengandung hijau
Oranye	82%	Cahaya alami / siang hari	Sulit dibedakan dari merah atau kuning
Kuning	90%	Cahaya terang & latar gelap	Sering tertukar dengan hijau muda
Hijau	87%	Latar netral & cahaya merata	Dominasi warna hijau bisa menyebabkan bias
Cyan	80%	Pencahayaan putih atau biru	Sering tertukar dengan biru
Biru	92%	Pencahayaan normal, latar terang	Cahaya kuning membuat warna tampak keunguan
Ungu	78%	Pencahayaan LED / lampu ungu	Sering tidak dikenali atau jadi biru tua
Merah Muda	75%	Pencahayaan alami	Sering tertukar dengan putih atau ungu

Secara umum, model warna HSV terbukti efektif untuk memisahkan informasi warna dari pencahayaan, berbeda dengan model RGB yang sensitif terhadap perubahan intensitas cahaya. Dengan HSV, pemisahan antara Hue (warna), Saturation (kejenuhan), dan Value (kecerahan) memungkinkan proses segmentasi warna yang lebih akurat.

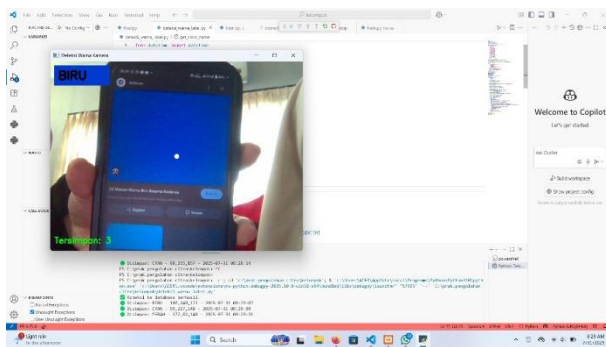
Penggunaan metode ini cukup ringan secara komputasi dan cocok untuk aplikasi real-time menggunakan perangkat keras sederhana. Namun, keterbatasan tetap ada, terutama jika warna objek mendekati nilai ambang batas HSV yang ditentukan.



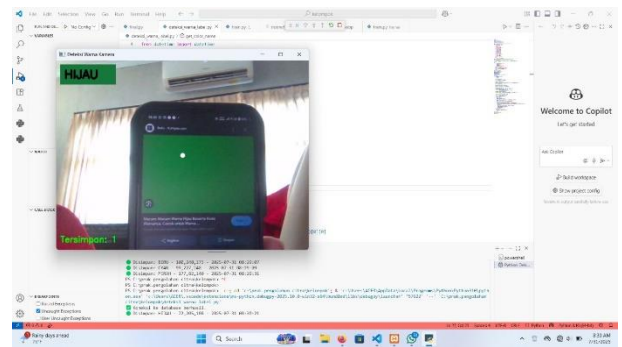
Gambar 1. Deteksi objek Merah



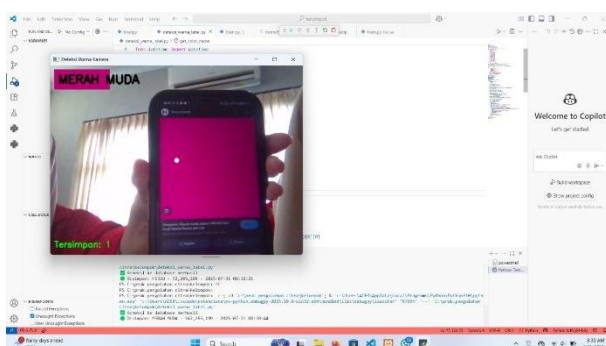
Gambar 2. Deteksi objek Cyan



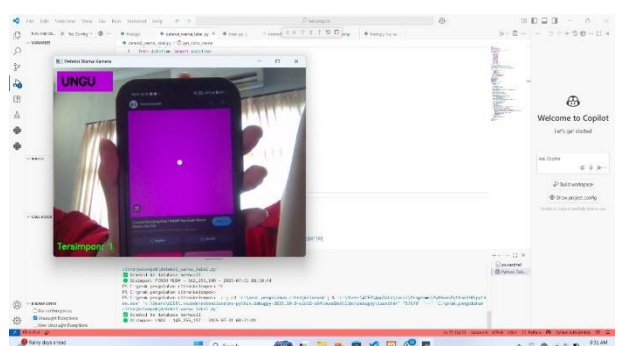
Gambar 3. Deteksi objek Biru



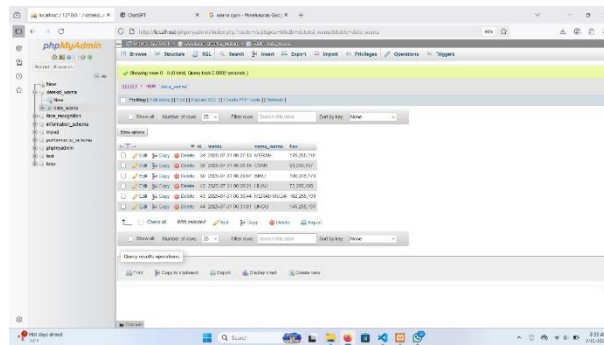
Gambar 4. Deteksi objek Hijau



Gambar 5. Deteksi objek Merah Muda



Gambar 6. Deteksi objek Ungu



Gambar 7. Deteksi objek Ungu

4. KESIMPULAN

Berdasarkan hasil yang diperoleh dari implementasi sistem deteksi objek berbasis warna menggunakan model warna HSV, dapat disimpulkan bahwa metode ini efektif dalam mengidentifikasi objek secara real-time dengan memanfaatkan karakteristik warna yang khas dari objek tersebut. Model warna HSV memberikan fleksibilitas dalam proses segmentasi warna karena mampu memisahkan informasi warna (hue) dari pencahayaan (value), sehingga lebih stabil terhadap perubahan intensitas cahaya dibandingkan model RGB.

Sistem yang dikembangkan mampu mendeteksi objek secara tepat dengan hasil visualisasi yang jelas dan responsif, terutama pada kondisi pencahayaan yang baik. Penggunaan parameter rentang warna (thresholding) yang sesuai pada nilai Hue, Saturation, dan Value menjadi faktor penentu dalam keberhasilan proses deteksi.

Adapun keterbatasan sistem ini terletak pada sensitivitas terhadap bayangan dan pencahayaan ekstrem. Oleh karena itu, pengembangan sistem ke depan dapat mempertimbangkan integrasi metode preprocessing lanjutan atau penggabungan dengan metode lain untuk meningkatkan akurasi deteksi.

5. DAFTAR PUSTAKA

- [1] Soekarman et al., PENGOLAHAN CITRA MENGGUNAKAN PYTHON. PT. Faaslib Serambi Media, 2025
- [2] Ahadi, Ahmad Husna., Gustina., Syawal, Muhammad Farhan., Aminuddin.F.H., Anzari, Yandi. (2024). Implementasi Sistem Pendeteksi Warna Objek Dengan Opencv-Python. SENTRI: Jurnal Riset Ilmiah. 3(7), 3573-3577.
- [3] Amrullah, D. L., Swedia, E. R., Cahyanti, M. And Dwi Septian, M. R. (2022) "Implementasi Color Detection Menggunakan Algoritma Midpoint Berbasis Sistem Operasi Android", Sebatik, 26(1), 121-122.
- [4] Prasetyo, D., & Wiliani, N. (2022). *Penerapan Convolutional Neural Network (CNN) Untuk Deteksi Dan Klasifikasi Objek Berbasis Warna*. Jurnal Teknologi Dan Sistem Komputer, 10(2), 115-123.
- [5] Havid Syafwan et al., ALGORITMA DAN STRUKTUR DATA MENGGUNAKAN PYTHON. Faaslib Serambi Media, 2025
- [6] Pratama, Yoga Putra., Prastiwinarti, Wiwi., Ahmad, Luthfiandra., Mahmuda, Zain Sultan.(2023). Perancangan Aplikasi Konversi RGB CMYK Berbasis Python. Journal Of Applied Electrical Engineering. 7(2), 105.
- [7] A. Y. Nugroho et al., LOGIKA DAN ALGORITMA : Pendekatan Praktis dengan Python. PT. Faaslib Serambi Media, 2025. [Online]. Available: <https://faaslibsmidia.com/Buku/Detail/LOGIKA-DAN-ALGORITMA-Pendekatan-Praktis-dengan-Python>