

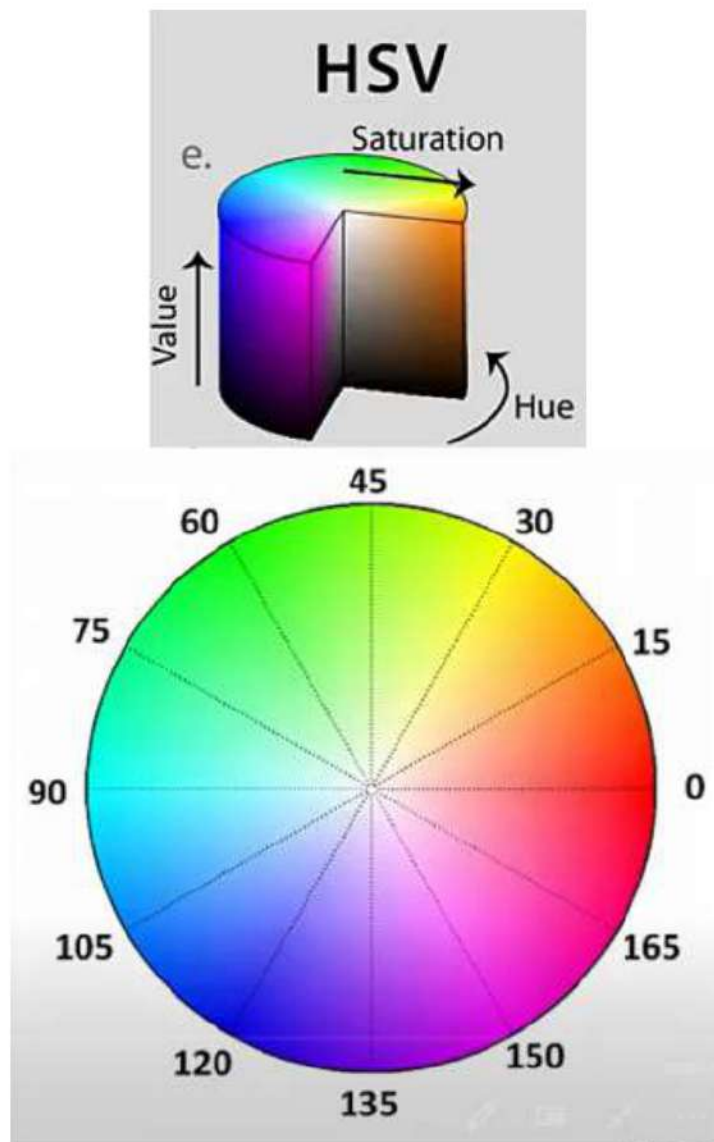
I. COLOR PROCESSING

Judul

PRAKTEK I1-COLOR TRACKING

Deskripsi

Proses mendeteksi objek berdasarkan identitas warna yang telah ditentukan. Objek berwarna RGB dikonversikan ke dalam atribut warna HSI. Pendeteksian warna dilakukan dengan menentukan range warna jenuh. Dengan menggunakan HSV lebih mudah untuk mengetahui tingkatan warna berdasarkan warna jenuh (*hue*), tingkat kecerahan warna jenuh berdasarkan penggunaan warna putih (*saturation*) dan nilai kecerahan warna berdasarkan intensitas cahaya (*Value/Brightness/intensity*)



Filter by Hue, terdiri dari range 0 sampai 180 dengan color range filter:

Red: -165 sampai 15

Green: -45 sampai 75

Blue: -90 sampai 120

Filter by saturation and *Value/Brightness/intensity*:

Biasanya mengatur range filter dari 50-255. Range saturation dari 0-60 akan mendekati warna putih dan Tange value 0-60 mendekati warna hitam

Estimasi waktu 15 menit

Prerequisite Objek berwarna
Web camera

Algoritma

1. Mengaktifkan kamera
2. Konversikan citra RGB ke HSI
3. Menentukan range warna yang akan dideteksi
4. Menampilkan hasil deteksi warna

Listing Program

```
import cv2
import numpy as np
cam = cv2.VideoCapture(0)
while True:
    _ frame = cam.read()
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    lower_color = np.array([66, 98, 100])
    upper_color = np.array([156, 232, 255])
    mask = cv2.inRange(hsv, lower_color, upper_color)
    result = cv2.bitwise_and(frame, frame, mask=mask)
    cv2.imshow("frame", frame)
    cv2.imshow("mask", mask)
    cv2.imshow("result", result)
```

```
    ey = cv2.waitKey(1)
    if key == 27:
        break
cam.release()
cv2.destroyAllWindows()
```

Hasil

Pada skenario percobaan ini sistem mencoba mendeteksi objek secara realtime yang berwarna biru dengan hasil sebagai berikut:



Gambar 1. Hasil color detection

PRAKTEK I2-COLOR PICKER	
Judul	PRAKTEK I2-COLOR PICKER
Deskripsi	Proses mendeteksi objek berdasarkan identitas warna dengan mengatur trackbar color space HSV (<i>Hue, Saturation and Value/Brightness/intensity</i>).
Estimasi waktu	15 menit
Prerequisite	Objek berwarna Web camera
Algoritma	<ol style="list-style-type: none">1. Mengaktifkan kamera2. Konversikan citra RGB ke HSI3. Membuat trackbar color

4. Menentukan range warna yang akan dideteksi berdasarkan color trackbar
5. Menampilkan hasil deteksi warna

Listing Program

```
import cv2
import numpy as np

def nothing(x):
    pass

cam = cv2.VideoCapture(0)
cv2.namedWindow("Trackbars")

cv2.createTrackbar("L - H", "Trackbars", 0, 179, nothing)
cv2.createTrackbar("L - S", "Trackbars", 0, 255, nothing)
cv2.createTrackbar("L - V", "Trackbars", 0, 255, nothing)
cv2.createTrackbar("U - H", "Trackbars", 179, 179, nothing)
cv2.createTrackbar("U - S", "Trackbars", 255, 255, nothing)
cv2.createTrackbar("U - V", "Trackbars", 255, 255, nothing)

while True:
    _, frame = cam.read()
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    l_h = cv2.getTrackbarPos("L - H", "Trackbars")
    l_s = cv2.getTrackbarPos("L - S", "Trackbars")
    l_v = cv2.getTrackbarPos("L - V", "Trackbars")
    u_h = cv2.getTrackbarPos("U - H", "Trackbars")
    u_s = cv2.getTrackbarPos("U - S", "Trackbars")
    u_v = cv2.getTrackbarPos("U - V", "Trackbars")
```

```
lower_color = np.array([l_h, l_s, l_v])
upper_color = np.array([u_h, u_s, u_v])
mask = cv2.inRange(hsv, lower_color, upper_color)
result = cv2.bitwise_and(frame, frame, mask=mask)

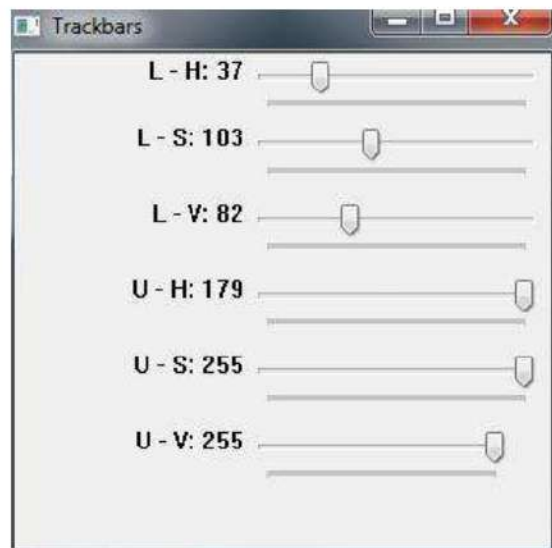
cv2.imshow("frame", frame)
cv2.imshow("mask", mask)
cv2.imshow("result", result)

key = cv2.waitKey(1)
if key == 27:
    break

cam.release()
cv2.destroyAllWindows()
```

Hasil

Skenario percobaan ini dilakukan untuk menentukan range warna yang akan dideteksi



Gambar 2. Color Trackbar



Gambar 3. Hasil color detection berdasarkan color picker