

TUGAS 9 – *Color Object Detection*
DATA VISUALISASI & IMAGE PROCESSING
Muhammad Naufaldi Dzakwan – 10024107

1. *Color Object Segmentation*

KODE PROGRAM

```
import cv2
import numpy as np

image = cv2.imread("apple_pict.jpg")

if image is None:
    print("Error gagal membaca gambar")
    exit()

hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

lower_red1 = np.array([0,120, 70])
upper_red1 = np.array([10,255,255])
mask1 = cv2.inRange(hsv, lower_red1,upper_red1)

lower_red2 = np.array([170,120,70])
upper_red2 = np.array([180,255,255])
mask2 = cv2.inRange(hsv, lower_red2,upper_red2)

mask = mask1+mask2
kernel = np.ones((5,5),np.uint8)
mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
mask = cv2.morphologyEx(mask, cv2.MORPH_DILATE, kernel)

result= cv2.bitwise_and(image, image, mask=mask)

cv2.imwrite("masked_object.jpg",result)
cv2.imwrite("mask.jpg", mask)
```

OUTPUT PROGRAM



Gambar 1.1 Input Gambar



Gambar 1.2 Output Gambar masking segmentasi warna merah

2. Color Object Detection (apel kuning)

KODE PROGRAM

```
# Coba detect warna apel kuning
import cv2
import numpy as np

image = cv2.imread("apple_pict.jpg")

if image is None:
    print("Error gagal membaca gambar")
    exit()

hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

# Ganti ke rentang HSV untuk warna yellow (apel kuning)
lower_yellow = np.array([20, 100, 50])
upper_yellow = np.array([28, 200, 255])
mask = cv2.inRange(hsv, lower_yellow, upper_yellow)

kernel = np.ones((5,5),np.uint8)
mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
mask = cv2.morphologyEx(mask, cv2.MORPH_DILATE, kernel)

contours,_ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

for contour in contours :
    if cv2.contourArea(contour) > 500 :
        x,y,w,h = cv2.boundingRect(contour)
        cv2.rectangle(image, (x,y), (x+w, y+h), (0,255,0),2)

        M = cv2.moments(contour)
        if M["m00"] != 0:
            cX = int(M["m10"] / M["m00"])
            cY = int(M["m01"] / M["m00"])
            cv2.circle(image, (cX,cY), 5, (0,0,255),-1)
            cv2.putText(image, f"({cX}, {cY})", (cX - 20, cY - 20),
                        cv2.FONT_HERSHEY_COMPLEX, 0.5, (255,255,255),2)

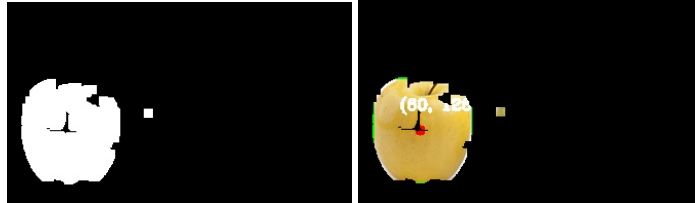
result= cv2.bitwise_and(image, image, mask=mask)

cv2.imwrite("masked_VisDat.jpg",result)
cv2.imwrite("mask_VisDat.jpg", mask)
```

OUTPUT PROGRAM



Gambar 1.3 Input Gambar



Gambar 1.4 Output Gambar Deteksi apel warna kuning

3. Color Object Video Detection (Baju merah)

KODE PROGRAM

```
# Video Object Color Detection baju Manchester United (merah)
import cv2
import numpy as np

# Inisialisasi webcam
cap = cv2.VideoCapture(0)

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

    # Konversi ke HSV
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # Rentang warna merah
    lower_red = np.array([0, 120, 70])
    upper_red = np.array([10, 255, 255])
    mask1 = cv2.inRange(hsv, lower_red, upper_red)
    lower_red = np.array([170, 120, 70])
    upper_red = np.array([180, 255, 255])
    mask2 = cv2.inRange(hsv, lower_red, upper_red)
    mask = mask1 + mask2

    # Operasi morfologi
    kernel = np.ones((5, 5), np.uint8)
    mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)

    # Temukan kontur
    contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

```

# Lacak centroid
for contour in contours:
    if cv2.contourArea(contour) > 500:
        # Hitung centroid
        M = cv2.moments(contour)
        if M["m00"] != 0:
            cx = int(M["m10"] / M["m00"])
            cy = int(M["m01"] / M["m00"])
            # Gambar centroid
            cv2.circle(frame, (cx, cy), 5, (0, 255, 0), -1)
            # Gambar bounding box
            x, y, w, h = cv2.boundingRect(contour)
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
            # Tampilkan koordinat
            cv2.putText(frame, f"({cx}, {cy})", (cx+10, cy), cv2.FONT_HERSHEY_SIMPLEX,
0.5, (0, 255, 0), 2)

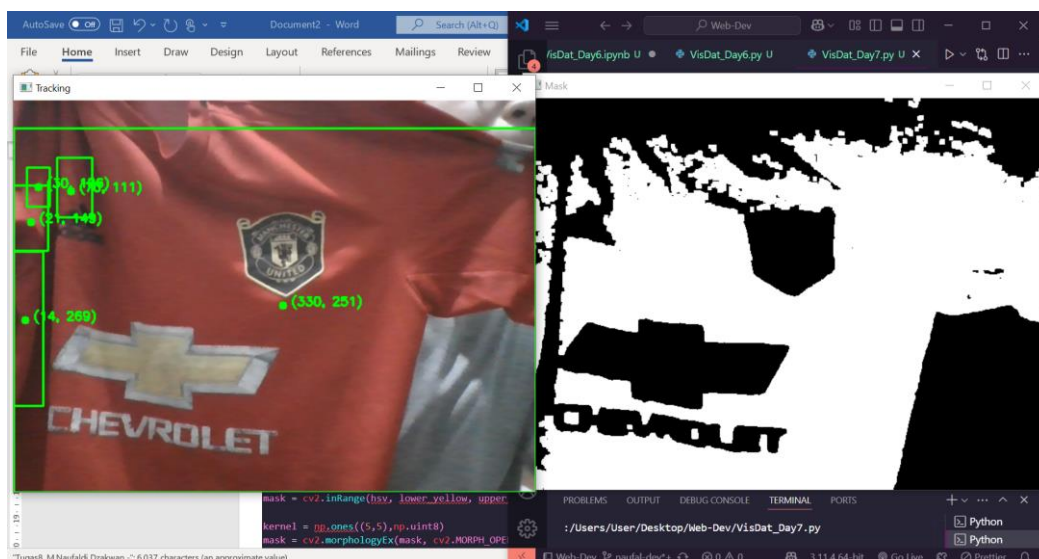
# Tampilkan frame
cv2.imshow('Tracking', frame)
cv2.imshow('Mask', mask)

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

# Bersihkan
cap.release()
cv2.destroyAllWindows()

```

OUTPUT PROGRAM



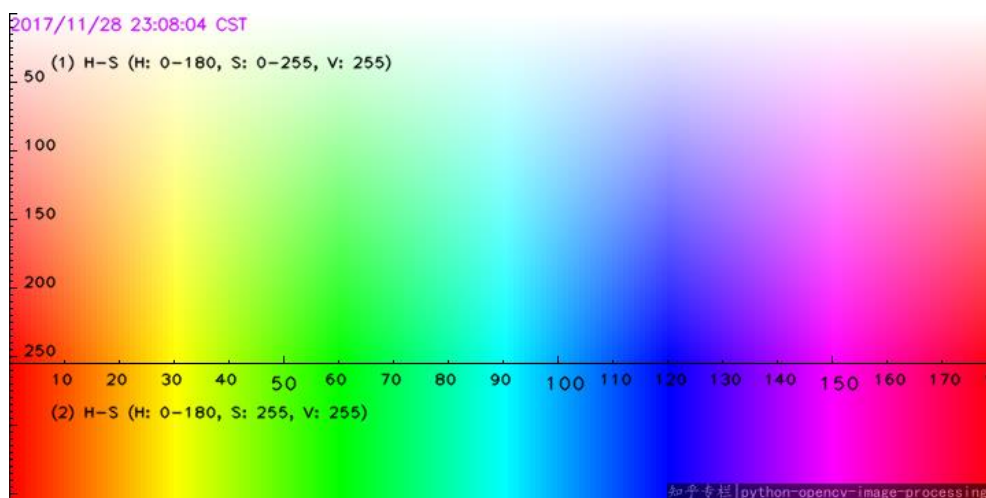
Gambar 1.5 Output Gambar (sebelah kiri - Object Detection, Sebelah kanan – Masking Object)

ANALISA

Semua kode python mencakup pipeline pemrosesan warna dari

- **Segmentasi warna statis** (dengan masking),
- **Deteksi objek dan koordinat titik pusatnya (*centroid*)**,
- **Tracking objek berwarna secara real-time.**

berisi rangkaian proses **deteksi dan pelacakan objek berbasis warna** menggunakan OpenCV, dimulai dari segmentasi warna pada gambar statis hingga pelacakan *real-time* lewat video. Proses utamanya melibatkan konversi citra dari format BGR ke HSV, karena ruang warna HSV memisahkan informasi warna (*hue*) dari intensitas, sehingga lebih stabil untuk segmentasi. Warna target seperti merah atau kuning diidentifikasi dengan menentukan rentang nilai *hue* yang sesuai, seperti yang digambarkan secara visual pada gambar HSV terlampir.



Gambar 1.6 Range Warna beserta kode HSV dalam 8-bit (0-255)

Setelah mask biner terbentuk dari threshold HSV, dilakukan filter morfologi untuk mengurangi *noise*, lalu kontur objek dianalisis untuk menggambar *bounding box* dan menentukan koordinat pusat (*centroid*). Semua teknik ini kemudian diterapkan ke video streaming dari webcam laptop untuk mendeteksi dan melacak objek dengan warna tertentu secara *real-time*. Maka dari itu, gambar HSV menjadi panduan penting untuk memilih dan menyetel nilai hue yang tepat dalam proses segmentasi dan tracking warna.