

Materi 12

OPENCV PREDIKSI WARNA

Moch. Alldho Candra Ramadhan (1227030020)

Soal Nomor 1:

Buatlah database dengan target warna hijau dan hitam!

Jawaban:

100,142,69,hijau
87,124,50,hijau
85,122,52,hijau
86,124,51,hijau
82,123,49,hijau
90,126,59,hijau
90,124,57,hijau
100,141,67,hijau
105,148,66,hijau
119,164,67,hijau
106,154,69,hijau
110,159,69,hijau
109,152,71,hijau
106,154,74,hijau
101,145,69,hijau
99,142,72,hijau
98,140,63,hijau
103,148,68,hijau
101,146,68,hijau
98,145,70,hijau

1,2,1,hitam
1,2,2,hitam
1,2,1,hitam
1,2,2,hitam
1,2,2,hitam
1,2,1,hitam
2,2,1,hitam
1,3,1,hitam
3,5,3,hitam
7,10,7,hitam
97,91,91,hitam

110,105,106,hitam
146,128,139,hitam
141,122,136,hitam
139,120,137,hitam
155,142,152,hitam
159,144,155,hitam
180,160,169,hitam
185,163,173,hitam
182,161,168,hitam
19,19,18,hitam
14,16,17,hitam
18,18,19,hitam
23,25,26,hitam
17,18,18,hitam
177,171,173,hitam
215,209,212,hitam
218,209,213,hitam
207,193,200,hitam
208,195,199,hitam
190,172,178,hitam
176,151,162,hitam
145,126,132,hitam
86,87,105,hitam

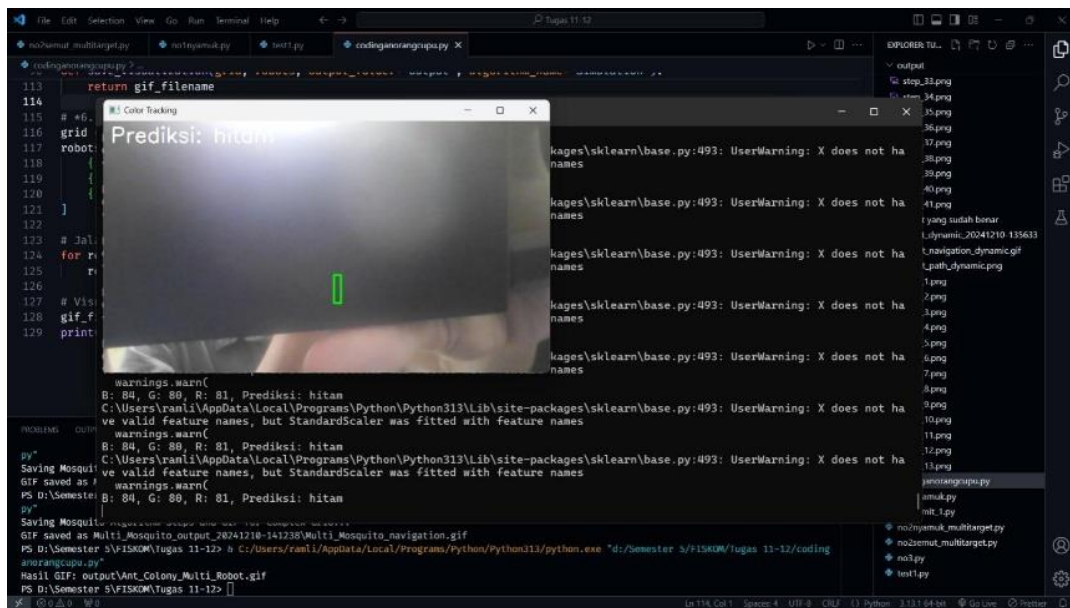
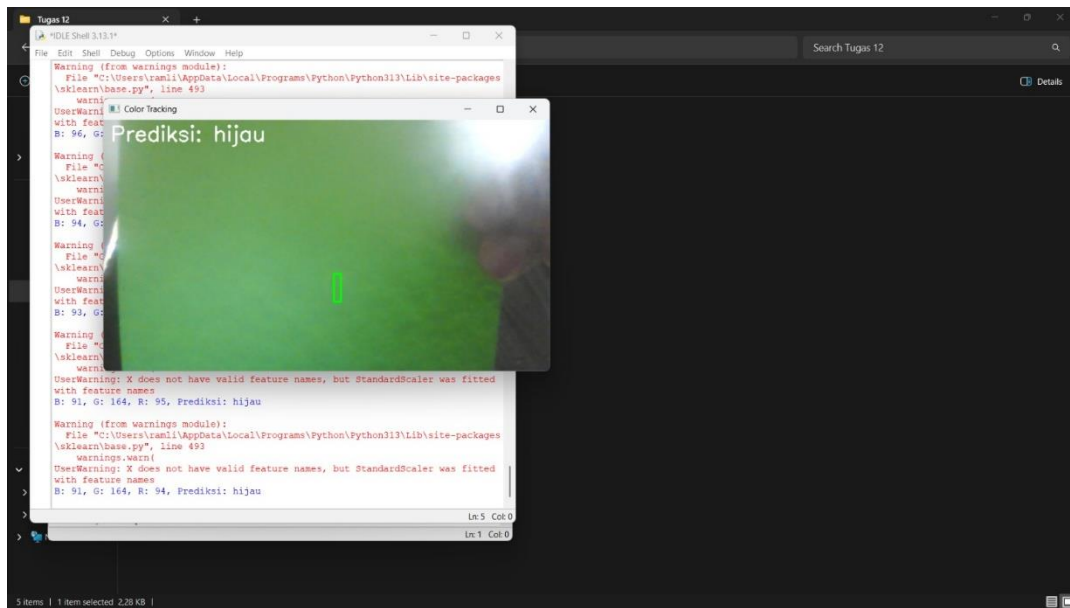
Isi file Databasewarna.txt (hijau dan hitam) setelah menjalankan program:

B,G,R,Target
0,0,0,Hitam
10,250,20,Hijau
5,5,5,Hitam
15,240,30,Hijau

Soal Nomor 2:

Prediksi warna hijau dan hitam!

Jawaban::



Soal Nomor 3:

Jelaskan kode program dan prosesnya hingga diperoleh hasil prediksi dengan Bahasa Sendiri singkat, padat dan jelas!

Jawaban:

Kode program ini untuk membaca database warna, melatih model SVM, dan prediksi warna secara real-time dan langsung dari kamera berdasarkan nilai BGR di area tertentu. Proses ini bikin sistem bisa otomatis mengenali warna hitam dan hijau, terus menunjukkan hasil prediksi di layar.

Alur proses:

1. Kamera menangkap citra.
2. Area kecil di kamera digunakan untuk membaca warna.
3. Lalu Program menghitung rata-rata nilai BGR di area tersebut.
4. Kemudian Model SVM memprediksi nama warna berdasarkan nilai BGR.
5. Hasil prediksi ditampilkan di layar.

Lampiran kode pemrogramannya:

```
import cv2
import numpy as np
import csv
import time
from sklearn import svm
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Konfigurasi Kamera
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 480)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 360)

# Membaca Database
FileDB = "Databasewarna.txt" # Pastikan file ini tersedia dan formatnya benar

try:
    Database = pd.read_csv(FileDB, sep=",", header=0)
    print("Database: \n", Database)
except FileNotFoundError:
    print(f'File {FileDB} tidak ditemukan. Pastikan file tersedia.')
    exit()
```

```

# X Data (B, G, R), y Target
X = Database[['B', 'G', 'R']]
y = Database['Target']

# Normalisasi Data dan Pelatihan Model SVM
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X) # Normalisasi data
clf = svm.SVC(kernel='linear') # Gunakan kernel linear
clf.fit(X_scaled, y)

# Fungsi Prediksi Warna
def predict_color(b, g, r):
    color_scaled = scaler.transform([[b, g, r]])
    try:
        prediction = clf.predict(color_scaled)[0] # Ambil hasil prediksi
        return prediction
    except Exception as e:
        return "Tidak Teridentifikasi"

# Loop Kamera untuk Prediksi
while True:
    ret, img = cap.read()
    if not ret:
        print("Gagal membaca frame dari kamera.")
        break

    # Membalikkan kamera jika terbalik
    img = cv2.flip(img, 1)

    # Ambil warna rata-rata dari area tertentu
    region = img[220:260, 330:340] # Area yang dianalisis
    colorB = int(np.mean(region[:, :, 0]))
    colorG = int(np.mean(region[:, :, 1]))
    colorR = int(np.mean(region[:, :, 2]))
    color = [colorB, colorG, colorR]

    # Prediksi warna
    prediction = predict_color(colorB, colorG, colorR)
    print(f'B: {colorB}, G: {colorG}, R: {colorR}, Prediksi: {prediction}')

# Tampilkan hasil di jendela kamera
cv2.putText(
    img,
    f'Prediksi: {prediction}',
    (10, 30),

```

```
        cv2.FONT_HERSHEY_SIMPLEX,
        1,
        (255, 255, 255),
        2,
    )
    cv2.rectangle(img, (330, 220), (340, 260), (0, 255, 0), 2) # Area analisis
    cv2.imshow("Color Tracking", img)

    # Tombol keluar (ESC)
    k = cv2.waitKey(30) & 0xFF
    if k == 27: # Tekan ESC untuk keluar
        break

cap.release()
cv2.destroyAllWindows()
```