

Praktikum Fisika Komputasi

Modul 12

Septian Tri Laksono

1227030032

Cek Kamera

```
import cv2
import numpy as np
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    frame = cv2.flip (frame, 1)
    cv2.imshow("camera", frame)
    key = cv2.waitKey (1)
    if key == 27:
        break
cap.release()
cv2.destroyAllWindows ()
```

Pengambilan data base warna hitam dan hijau

```
import cv2
import numpy as np
import csv

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

# Nama file database
FileDB = 'DatabaseWarna.txt'

# Header untuk file CSV
header = ['B', 'G', 'R', 'Target']

# Buat file CSV jika belum ada
try:
    with open(FileDB, 'x', newline="") as f:
        writer = csv.writer(f)
        writer.writerow(header)
except FileExistsError:
```

```

print(f'{FileDB} sudah ada, melanjutkan penambahan data.")

print("Tekan tombol berikut untuk menambahkan data warna:")
print("1: Merah, 2: Hijau, 3: Biru, 4: Hitam, 5: Kuning, 6: Putih, [ESC]: Keluar")

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]

    # Tampilkan area analisis dan warna rata-rata
    cv2.rectangle(img, (330, 220), (340, 260), (0, 255, 0), 2)
    cv2.putText(img, f'B: {colorB}, G: {colorG}, R: {colorR}', (10, 30),
                cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
    cv2.imshow("Database Color Capture", img)

    # Deteksi tombol untuk menentukan warna
    key = cv2.waitKey(30) & 0xff

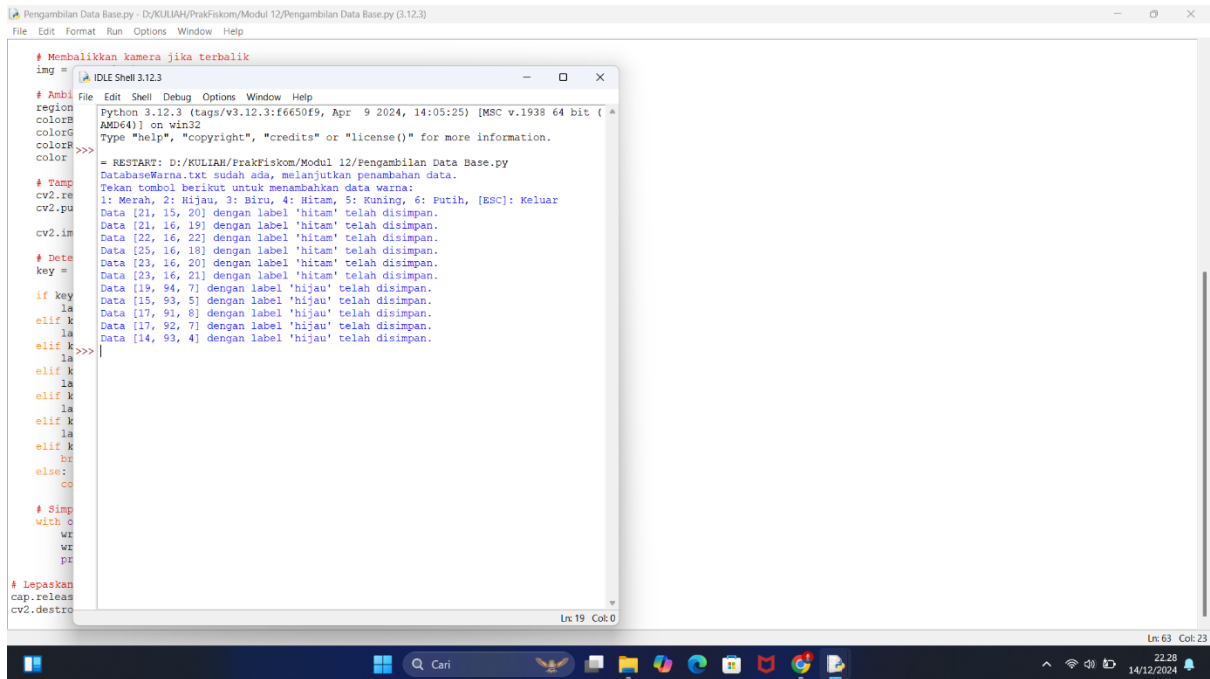
    if key == ord('1'): # Merah
        label = "merah"
    elif key == ord('2'): # Hijau
        label = "hijau"
    elif key == ord('3'): # Biru
        label = "biru"
    elif key == ord('4'): # Hitam
        label = "hitam"
    elif key == ord('5'): # Kuning
        label = "kuning"
    elif key == ord('6'): # Putih
        label = "putih"
    elif key == 27: # ESC untuk keluar
        break
    else:
        continue

    # Simpan data ke file CSV
    with open(FileDB, 'a', newline="") as f:
        writer = csv.writer(f)

```

```
writer.writerow(color + [label])
print(f'Data {color} dengan label '{label}' telah disimpan.")
```

```
# Lepaskan kamera dan tutup semua jendela
cap.release()
cv2.destroyAllWindows()
```



Prediksi Warna

```
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)

# Gunakan kernel linear untuk model SVM
clf = svm.SVC(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

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

    # 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]))

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

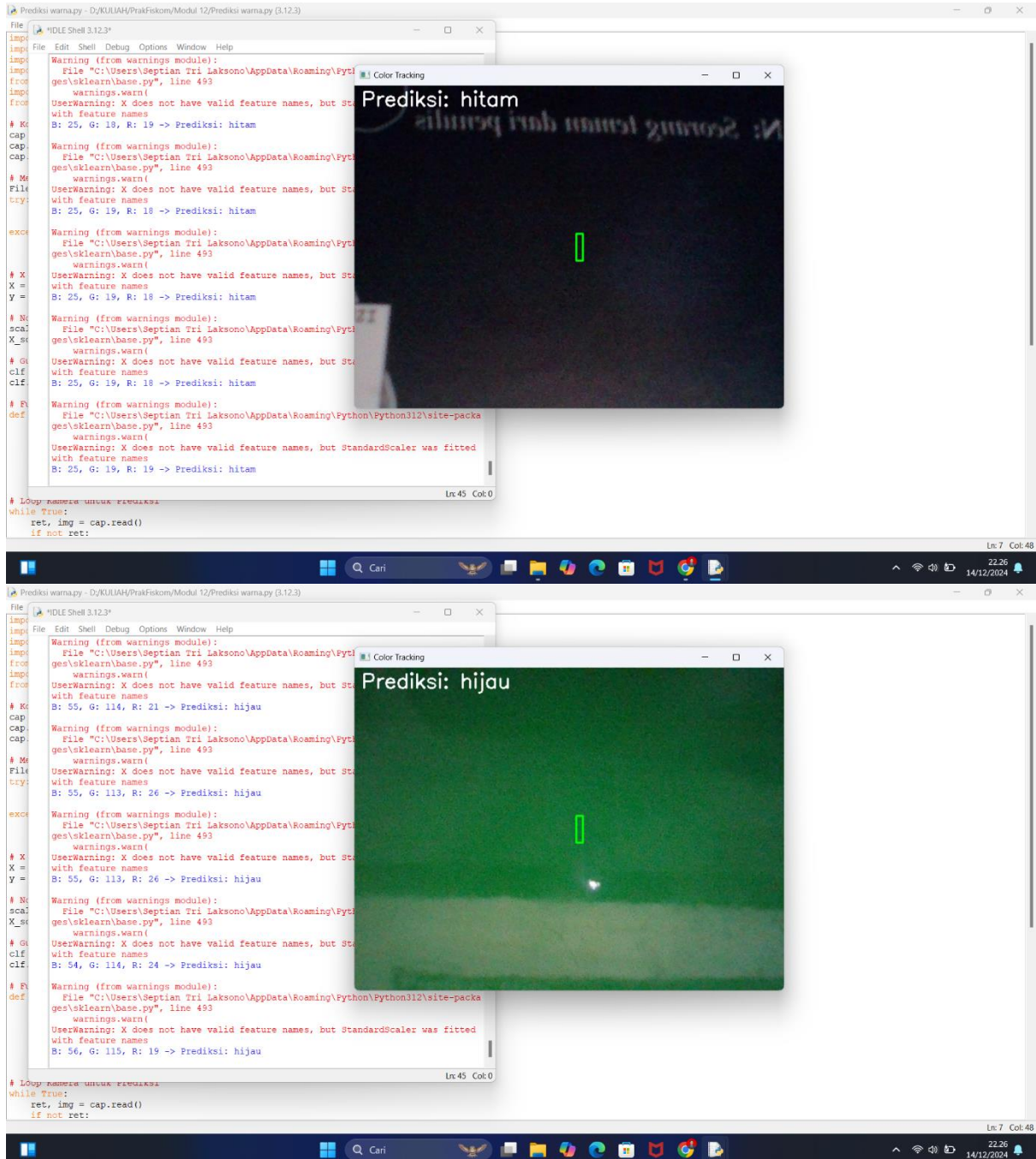
    # Tampilkan hasil di jendela kamera
    cv2.putText(img, f'Prediksi: {prediction}', (10, 30),
                cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
    cv2.rectangle(img, (330, 220), (340, 260), (0, 255, 0), 2) # Area analisis
    cv2.imshow("Color Tracking", img)

    # Tombol keluar (ESC)
    k = cv2.waitKey(30) & 0xff

```

```
if k == 27: # Tekan ESC untuk keluar
    break
```

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



Pada modul OpenCV prediksi warna ini menggunakan 3 kode program utama, yang pertama yaitu kode program cek kamera dengan menggunakan cv2.VideoCapture(0) untuk

membuka kamera default atau webcam yang nantinya digunakan untuk mendeteksi warna. Kode program kedua yaitu pengambilan data base warna, jadi sebelum kita mendeteksi warna, sebelumnya kita harus mengambil data base terlebih dahulu agar warna yang ingin kita deteksi dapat dikenali seperti warna merah, hitam, biru, hijau, dll. Data base yang kita ambil akan tersimpan dalam bentuk txt. Lalu kode yang ketiga yaitu kode prediksi warna, kode ini membaca data warna dari file database yang telah kita buat sebelumnya, melatih model SVM, dan menggunakan kamera untuk memprediksi warna secara real-time.