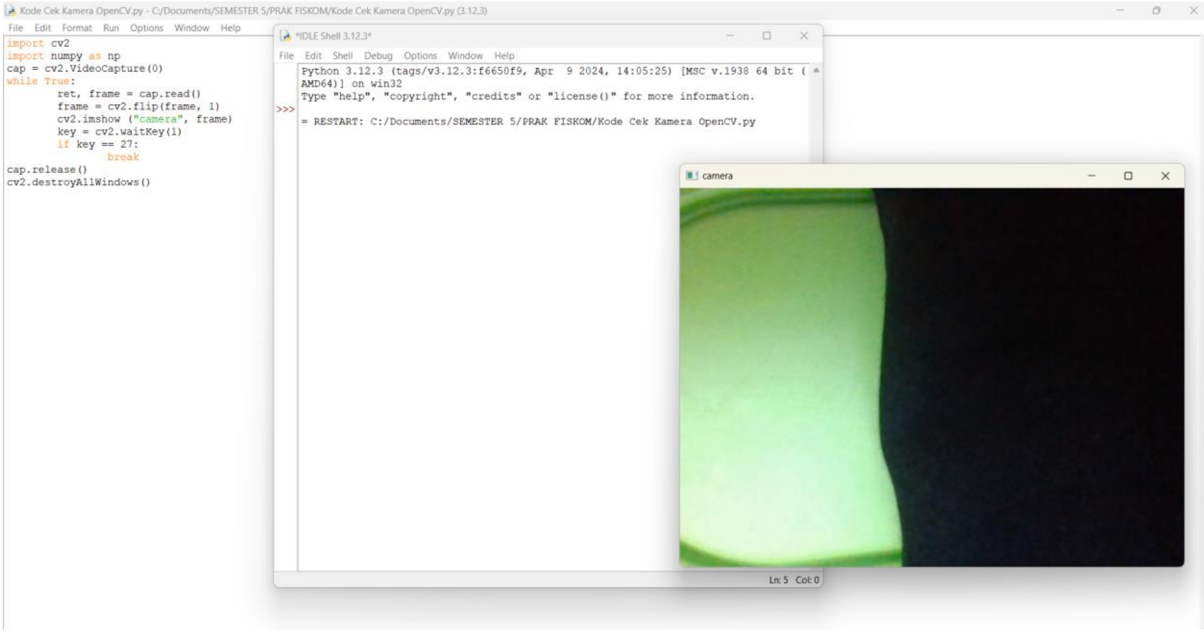
PRAKTIKUM FISIKA KOMPUTASI
OPENCV PREDIKSI WARNA

Oleh:
Eneng Yulia Pebryanti
NIM 1227030013

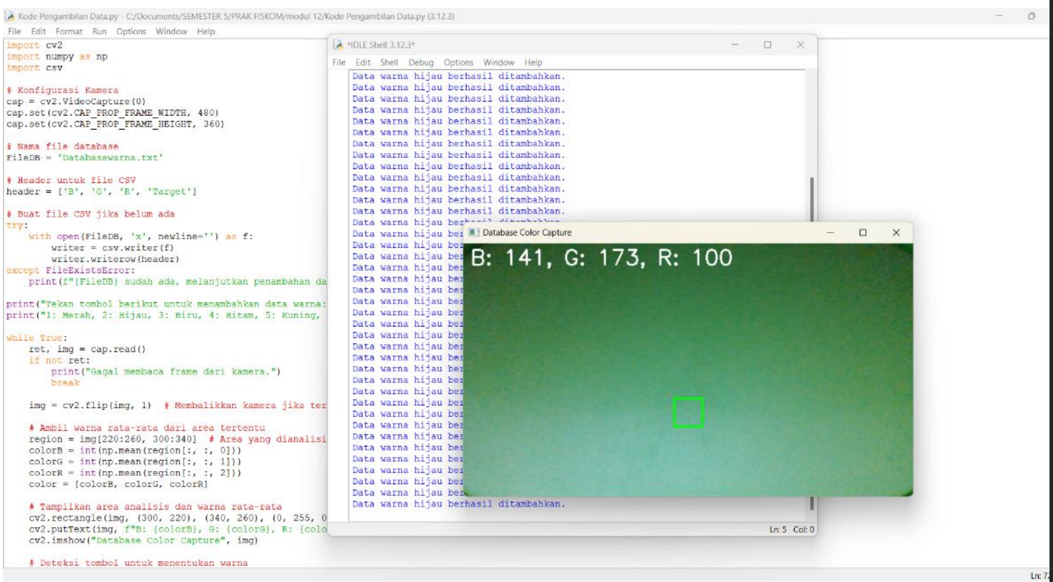
1. Pengambilan Data

a. Kode cek kamera



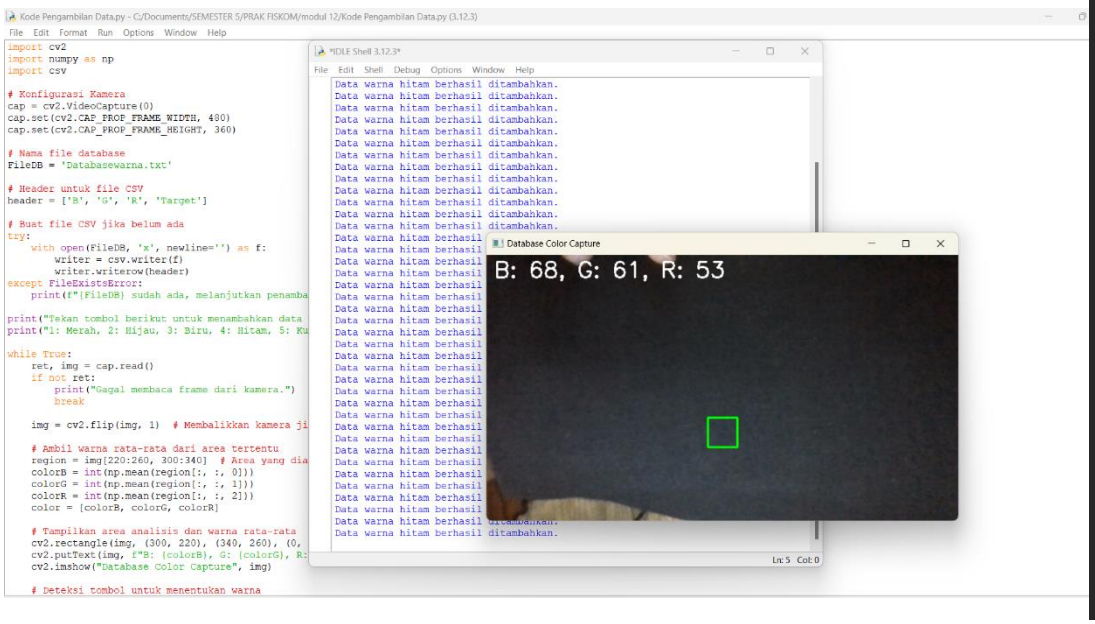
b. Kode pengambilan data

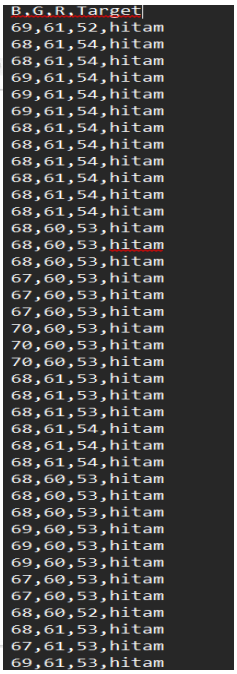
➢ Hijau



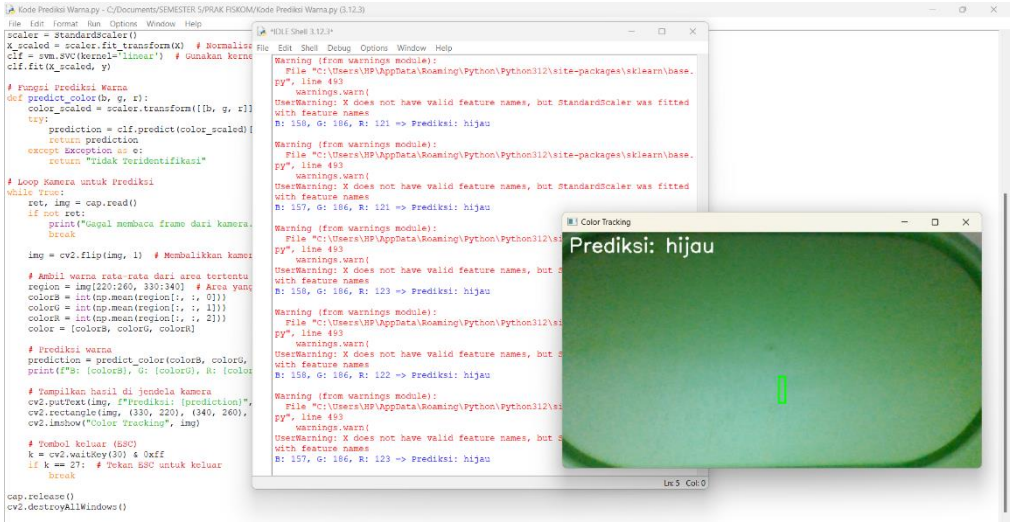
```
B,G,R,Target
145,174,103,hijau
146,175,103,hijau
144,175,103,hijau
145,175,103,hijau
144,174,103,hijau
144,174,103,hijau
144,174,103,hijau
143,174,101,hijau
143,173,101,hijau
142,173,100,hijau
142,173,101,hijau
142,173,101,hijau
143,173,100,hijau
143,173,101,hijau
142,173,100,hijau
142,173,101,hijau
141,173,101,hijau
142,173,100,hijau
141,172,100,hijau
141,172,100,hijau
141,173,101,hijau
141,172,100,hijau
142,173,101,hijau
142,173,101,hijau
142,173,101,hijau
142,173,102,hijau
142,173,100,hijau
142,173,101,hijau
143,173,100,hijau
142,173,101,hijau
142,173,101,hijau
141,172,99,hijau
141,172,100,hijau
141,172,99,hijau
141,172,101,hijau
140,172,99,hijau
140,172,99,hijau
141,172,99,hijau
140,172,100,hijau
```

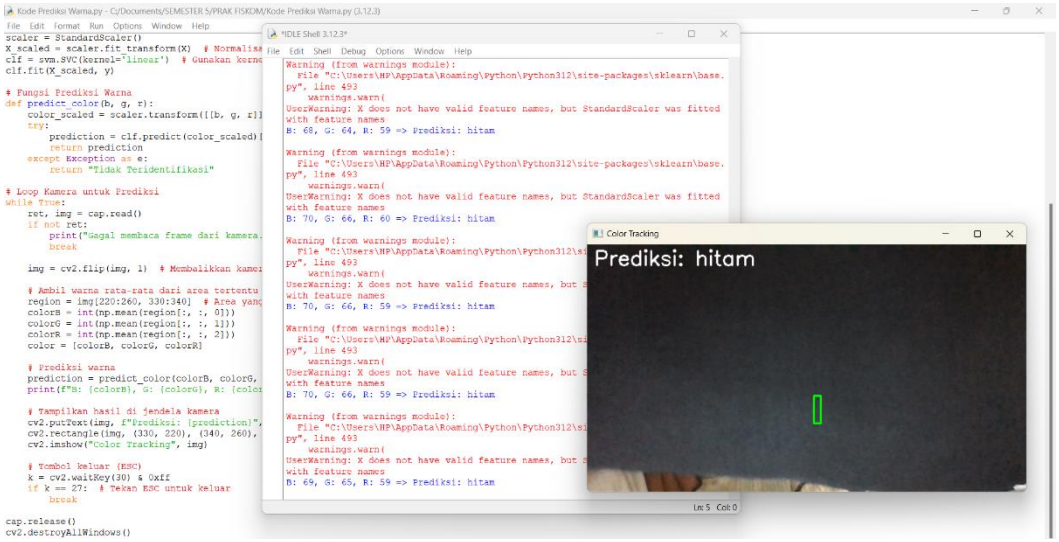
- Hitam





2. Prediksi Warna





3. **Penjelasan kode program dan proses hingga diperoleh hasil prediksi**

Langkah pertama adalah membuat database warna dengan menangkap warna dari kamera dan menyimpannya ke dalam file. Berikut adalah kode program yang digunakan:

Kode Pengambilan DataBase.py - C:/Documents/SEMESTER 5/PRAK FISKOM/Kode Pengambilan DataBase.py (3.12.3)

File Edit Format Run Options Window Help

```
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

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

    # Ambil warna rata-rata dari area tertentu
    region = img[220:260, 300:340]  # 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, (300, 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(20) & 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'
    else:
        continue

    with open(FileDB, 'a', newline='') as f:
        writer = csv.writer(f)
        writer.writerow([colorB, colorG, colorR, label])
        print(f"Data warna {label} berhasil ditambahkan.")

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

Kode di atas bekerja dengan membuka kamera menggunakan fungsi cv2.VideoCapture(0). Kamera menangkap frame video secara real-time, dan warna piksel di tengah layar diambil untuk dianalisis dalam format BGR. Fungsi cv2.putText() digunakan untuk menampilkan nilai BGR tersebut di layar. Warna yang ingin disimpan dapat ditambahkan ke dalam database dan databese akan tersimpan di file, database_warna.txt menggunakan np.savetxt().

Setelah database selesai dibuat, langkah berikutnya adalah menggunakan database ini untuk melatih model prediksi menggunakan algoritma Support Vector Machine (SVM).

Kode Prediksi Warna.py - C:/Documents/SEMESTER 5/PRAK FISKOM/Kode Prediksi Warna.py (3.12.3)

File Edit Format Run Options Window Help

```
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
Database = pd.read_csv(FileDB, sep=",", header=0)
print("Database:\n", Database)

# 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

    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]))
    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()
```

Pada kode ini, database yang telah dibuat sebelumnya dimuat menggunakan np.loadtxt() untuk mendapatkan nilai-nilai BGR. Label warna diberikan berdasarkan kondisi nilai BGR. Model Support Vector Machine (SVM) dilatih menggunakan model.fit() untuk mengenali pola hubungan antara nilai BGR dan label warna. Ketika kamera menangkap warna baru, nilai BGR dari piksel tengah layar dianalisis dan diprediksi oleh model untuk menentukan apakah warna tersebut hijau atau hitam. Prediksi ini ditampilkan di layar menggunakan cv2.putText. Dengan langkah-langkah

ini, program mampu membuat database warna hijau dan hitam, melatih model, serta memprediksi warna secara real-time dengan tingkat akurasi yang baik.