

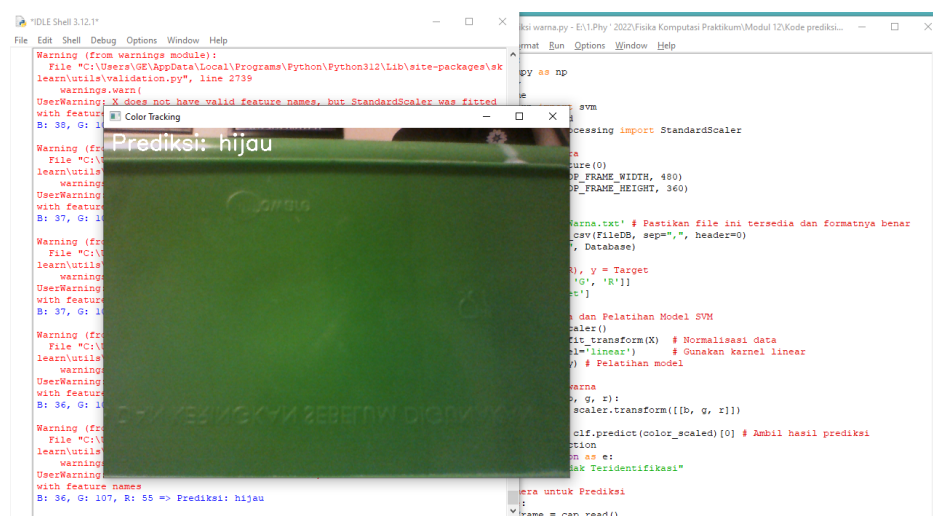
# 1. Buatlah database dengan target warna hijau dan hitam!

berikut adalah database warna yang dibuat hijau dan hitam.

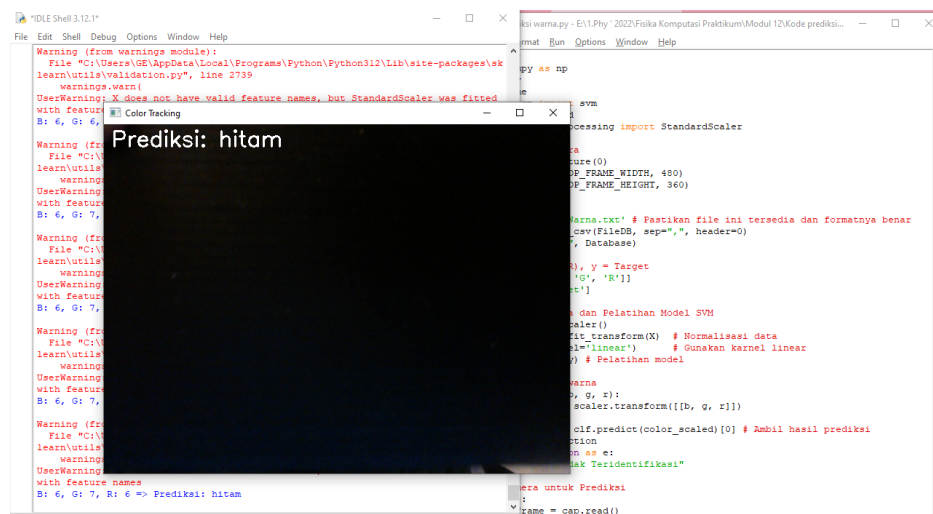
35,31,25,hitam	
32,29,26,hitam	
28,28,26,hitam	
27,28,22,hitam	
30,28,24,hitam	
34,29,22,hitam	52,78,51,hijau
32,27,28,hitam	51,68,43,hijau
29,26,26,hitam	53,110,80,hijau
30,26,27,hitam	41,99,53,hijau
45,36,30,hitam	48,81,46,hijau
46,36,32,hitam	69,91,64,hijau
47,36,31,hitam	43,68,48,hijau
44,36,31,hitam	74,97,68,hijau
37,34,31,hitam	74,97,65,hijau
34,33,29,hitam	57,78,49,hijau
33,33,28,hitam	95,128,102,hijau
34,35,29,hitam	
34,35,29,hitam	
37,33,30,hitam	
41,37,30,hitam	

## 2. Prediksi warna hijau dan hitam.

### 1. Prediksi warna Hijau



## 2. Prediksi warna Hitam



## 3. Jelaskan kode program dan prosesnya hingga diperoleh hasil prediksi dengan Bahasa sendiri singkat, padat dan jelas!

Kode program menggunakan beberapa library, yaitu cv2 untuk menangani kamera dan pemrosesan gambar, numpy untuk operasi array dan komputasi, pandas untuk membaca file database warna, serta sklearn untuk membangun model SVM (Support Vector Machine) dan normalisasi data.

Kamera diatur menggunakan cv2.VideoCapture(0), yang menunjukkan kamera utama atau bukan menggunakan web cam. Resolusi kamera diatur menjadi 480x360 piksel agar sesuai dengan kebutuhan aplikasi.

File database DatabaseWarna.txt dibaca menggunakan pandas. File ini berisi data nilai warna biru (B), hijau (G), dan merah (R) sebagai fitur input, serta label warna yang diinginkan sebagai target keluaran.

Setelah database dibaca, data warna dinormalisasi menggunakan StandardScaler untuk memastikan distribusi data seragam. Model SVM dengan kernel linear kemudian dilatih menggunakan data hasil normalisasi untuk memetakan hubungan antara warna dan labelnya.

Fungsi `predict_color` didefinisikan untuk menerima nilai warna (B, G, R), melakukan normalisasi ulang, dan menggunakan model SVM yang telah dilatih untuk memprediksi warna berdasarkan nilai input tersebut.

Program masuk ke dalam loop kamera yang berjalan terus-menerus untuk menangkap gambar secara real-time. Dalam setiap frame yang ditangkap, area tertentu (220:260, 330:340) dipilih untuk dianalisis. Nilai rata-rata warna hijau, dan hitam dari area tersebut dihitung dan digunakan sebagai input ke fungsi prediksi.

Prediksi warna yang dihasilkan oleh model ditampilkan di layar kamera secara langsung menggunakan fungsi cv2.putText. Informasi warna yang terdeteksi diperbarui secara real-time pada setiap frame.

Program terus berjalan hingga pengguna menekan tombol ESC, yang akan menghentikan loop, melepaskan kamera, dan menutup semua jendela aplikasi.