

# Laporan Praktikum Fisika Komputasi

Tugas 12 : Senin, 09 Desember 2024

Disusun oleh : Najlah Rupaidah

NIM : 1227030025

1. Sebelum warna dapat di prediksi dengan tepat, langkah pertama yang dilakukan adalah membuat database terlebih dahulu, dimana setiap warna memiliki kode angkanya tersendiri, yaitu: 1 Merah, 2 Hijau, 3 Biru, 4 Hitam, 5 Kuning, dan 6 Putih. Maka untuk membuat database dari warna hijau, kita harus mengklik nomor 2, dan webcam akan mengklasifikasikannya kedalam database warna hijau, begitupun untuk warna hitam, kita harus mengklik nomor 4, dan webcam akan mengklasifikasikannya kedalam database warna hitam. Adapun database yang telah dibuat untuk kedua warna tersebut, adalah sebagai berikut:

B,G,R,Target
99,163,98,hijau
100,165,99,hijau
101,165,99,hijau
101,165,99,hijau
101,165,100,hijau
101,165,99,hijau
101,165,99,hijau
101,165,99,hijau
101,165,100,hijau
102,166,100,hijau
102,166,99,hijau
102,166,100,hijau
102,166,100,hijau
102,166,99,hijau
102,166,100,hijau
102,166,100,hijau
102,166,100,hijau
102,166,100,hijau
109,174,107,hijau
101,163,100,hijau
114,180,112,hijau
108,175,105,hijau
107,174,104,hijau
107,174,104,hijau
107,174,104,hijau
107,175,104,hijau
107,175,104,hijau
107,175,104,hijau
108,174,104,hijau
108,174,104,hijau
107,174,104,hijau
107,175,104,hijau
108,175,105,hijau
108,175,104,hijau

108,175,105,hijau
107,175,105,hijau
107,175,104,hijau
177,164,85,hitam
34,30,30,hitam
29,25,26,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,23,24,hitam
29,23,24,hitam
29,23,24,hitam
29,23,24,hitam
29,23,24,hitam
29,23,24,hitam
29,23,24,hitam
29,23,24,hitam
29,24,25,hitam
30,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,24,25,hitam
29,23,24,hitam
30,23,24,hitam
30,23,24,hitam
30,23,24,hitam
29,23,24,hitam
29,23,24,hitam

2. Setelah membuat database, maka selanjutnya adalah memprediksi warna, berikut adalah prediksinya:



3. Deteksi warna berbasis OpenCV memiliki tiga kode program yang harus dijalankan, yaitu:

- Pengecekan 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()
```

Kode program tersebut berfungsi untuk mengecek apakah kamera laptop atau webcam dapat berfungsi dengan baik. Program ini menggunakan *library* cv2 sebagai *library* OpenCV untuk membuka kamera laptop atau webcam dan menampilkan video secara langsung.

- Pengambilan Database

```
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
    color = [
        int(np.mean(region[:, :, 0])),
```

```

        int(np.mean(region[:, :, 1])),
        int(np.mean(region[:, :, 2]))
    ]
    color = [color[0], color[1], color[2]]

    # Tampilkan area analisis dan warna rata-rata
    cv2.rectangle(img, (300, 220), (340, 260), (0, 255, 0), 2)
    cv2.putText(img, f'B: {color[0]}, G: {color[1]}, R: {color[2]}',
    (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
    cv2.imshow("Kamera dengan Warna Capture", img)

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

    cap.release()
cv2.destroyAllWindows()

```

Kode program ini berfungsi untuk mengumpulkan data warna menggunakan kamera. Kamera akan menangkap gambar, lalu program mengambil warna rata-rata dari sebuah kotak kecil di layar. Kotak ini ditandai dengan garis hijau agar tahu area mana yang sedang dianalisis.

Lalu, tekan tombol angka pada keyboard untuk memberi label pada warna yang terdeteksi, misalnya 1, 2, 3, 4, 5, 6 berturut-turut untuk warna merah, hijau, biru, hitam, kuning, atau putih. Data warna berupa nilai rata-rata dari warna biru (B), hijau (G), dan merah (R), beserta label warna yang dipilih, akan disimpan ke file bernama **DatabaseWarna.txt**. Jika file belum ada, program akan membuatnya otomatis.

Proses ini berjalan terus jika belum menekan tombol ESC untuk keluar. Program ini berguna untuk membuat kumpulan data warna yang bisa dipakai untuk selanjutnya dianalisis pada kode program prediksi warna.

- 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
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, 330:370] # 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), (370, 260), (0, 255, 0), 2) # Area
    analisis
    cv2.imshow("Color Tracking", img)

    # Tombol keluar
    k = cv2.waitKey(80) & 0xff
    if k == 27: # Tekan ESC untuk keluar
        break

    cap.release()
cv2.destroyAllWindows()

```

Kode program ini dibuat untuk mendeteksi dan mengenali warna secara langsung menggunakan kamera, dengan memanfaatkan algoritma *Support Vector Machine* (SVM). Pertama, program mengimpor pustaka seperti **OpenCV** untuk mengolah video dari kamera, **NumPy** untuk perhitungan angka, **Pandas** untuk membaca data warna dari file, dan **sklearn** untuk melatih model SVM.

Kamera diatur untuk merekam video dengan resolusi tertentu, kemudian program membaca data warna dari file CSV yang berisi nilai warna dalam format BGR (biru, hijau, merah) dan label warna.

Program ini memiliki fungsi untuk memprediksi warna berdasarkan nilai BGR. Saat kamera merekam, program mengambil area kecil dari gambar dan menghitung rata-rata warnanya. Nilai rata-rata ini digunakan untuk memprediksi warna, dan hasil prediksi ditampilkan di layar secara langsung, dengan area yang dianalisis diberi tanda kotak warna hijau.