

PRAKTIKUM FISIKA KOMPUTASI

OPENCV PREDIKSI WARNA

Darniel Trio Apriliansyah

NIM. 1227030009

1. Buatlah database dengan target warna hijau dan hitam!

Hitam	Hijau
B, G, R, Target	B, G, R, Target
B: 56, G: 55, R: 72, Prediksi: hitam	B: 34, G: 177, R: 76, Prediksi: Hijau
B: 56, G: 55, R: 72, Prediksi: hitam	B: 50, G: 205, R: 50, Prediksi: Hijau
B: 56, G: 55, R: 75, Prediksi: hitam	B: 0, G: 128, R: 0, Prediksi: Hijau
B: 56, G: 55, R: 73, Prediksi: hitam	B: 85, G: 107, R: 47, Prediksi: Hijau
B: 55, G: 55, R: 73, Prediksi: hitam	B: 46, G: 139, R: 87, Prediksi: Hijau
B: 53, G: 53, R: 70, Prediksi: hitam	B: 60, G: 179, R: 113, Prediksi: Hijau
B: 53, G: 52, R: 68, Prediksi: hitam	B: 0, G: 255, R: 127, Prediksi: Hijau
B: 55, G: 54, R: 72, Prediksi: hitam	B: 127, G: 255, R: 0, Prediksi: Hijau
B: 56, G: 55, R: 74, Prediksi: hitam	B: 144, G: 238, R: 144, Prediksi: Hijau
B: 55, G: 55, R: 72, Prediksi: hitam	B: 0, G: 100, R: 0, Prediksi: Hijau
B: 56, G: 55, R: 72, Prediksi: hitam	B: 0, G: 250, R: 154, Prediksi: Hijau
B: 56, G: 56, R: 75, Prediksi: hitam	B: 0, G: 255, R: 0, Prediksi: Hijau
B: 54, G: 59, R: 81, Prediksi: hitam	B: 173, G: 255, R: 47, Prediksi: Hijau
B: 54, G: 56, R: 78, Prediksi: hitam	B: 0, G: 201, R: 87, Prediksi: Hijau
B: 46, G: 45, R: 56, Prediksi: hitam	B: 32, G: 178, R: 170, Prediksi: Hijau
B: 41, G: 41, R: 50, Prediksi: hitam	B: 64, G: 224, R: 208, Prediksi: Hijau
B: 44, G: 44, R: 54, Prediksi: hitam	B: 152, G: 251, R: 152, Prediksi: Hijau
B: 44, G: 44, R: 55, Prediksi: hitam	B: 34, G: 139, R: 34, Prediksi: Hijau
B: 40, G: 41, R: 47, Prediksi: hitam	B: 107, G: 142, R: 35, Prediksi: Hijau
B: 34, G: 36, R: 39, Prediksi: hitam	B: 124, G: 252, R: 0, Prediksi: Hijau

2. Prediksi warna hijau dan hitam!



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

Kode Program 1:

```
import cv2
import numpy as np

# Membuka kamera
cap = cv2.VideoCapture(0)

while True:
    # Membaca frame dari kamera
    ret, frame = cap.read()

    if not ret:
        print("Gagal membaca frame dari kamera.")
        break

    # Membalik frame secara horizontal
    frame = cv2.flip(frame, 1)

    # Menampilkan frame
    cv2.imshow("Camera", frame)

    # Menunggu tombol tekan, keluar jika tombol ESC ditekan
    key = cv2.waitKey(1)
    if key == 27: # Tombol ESC
        break

# Melepaskan kamera dan menutup jendela
cap.release()
cv2.destroyAllWindows()
```

Kode Program 2:

```
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.")

cap.release()
cv2.destroyAllWindows()

```

Kode Program 3

```

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()

```

Ketiga program bekerja secara berurutan untuk menangkap, memproses, dan memprediksi warna secara real-time. Program pertama digunakan untuk menampilkan video dari kamera dan memastikan kamera berfungsi dengan baik. Program kedua bertujuan untuk mengumpulkan data warna dengan menangkap nilai rata-rata biru (B), hijau (G), dan merah (R) dari area tertentu pada frame kamera, memberi label warna berdasarkan target, dan menyimpannya ke file Databasewarna.txt sebagai dataset pelatihan. Program ketiga membaca file database tersebut, menormalisasi data menggunakan StandardScaler, melatih model Support Vector Machine (SVM), dan melakukan prediksi warna berdasarkan data real-time dari kamera. Prediksi warna yang dihasilkan ditampilkan langsung pada layar,

memungkinkan pengguna mendeteksi warna secara otomatis sesuai dengan model yang telah dilatih.