**Praktikum Fisika Komputasi**

**Tugas 12 : OpenCV Prediksi Warna**

Abdan shiddiq mubarok (1227030001)

|  |  |
| --- | --- |
| **Warna** | |
| Hijau  B, G, R, Target  B: 35, G: 180, R: 77, Prediksi: Hijau  B: 51, G: 208, R: 51, Prediksi: Hijau  B: 1, G: 130, R: 1, Prediksi: Hijau  B: 86, G: 110, R: 48, Prediksi: Hijau  B: 47, G: 142, R: 88, Prediksi: Hijau  B: 61, G: 182, R: 114, Prediksi: Hijau  B: 1, G: 258, R: 128, Prediksi: Hijau  B: 128, G: 258, R: 1, Prediksi: Hijau  B: 145, G: 241, R: 145, Prediksi: Hijau  B: 1, G: 103, R: 1, Prediksi: Hijau  B: 1, G: 253, R: 155, Prediksi: Hijau  B: 1, G: 258, R: 1, Prediksi: Hijau  B: 174, G: 258, R: 48, Prediksi: Hijau  B: 1, G: 204, R: 88, Prediksi: Hijau  B: 33, G: 181, R: 171, Prediksi: Hijau  B: 65, G: 227, R: 209, Prediksi: Hijau  B: 153, G: 254, R: 153, Prediksi: Hijau  B: 35, G: 142, R: 35, Prediksi: Hijau  B: 108, G: 145, R: 36, Prediksi: Hijau | Hitam  B, G, R, Target  B: 57, G: 58, R: 73, Prediksi: hitam  B: 57, G: 58, R: 73, Prediksi: hitam  B: 57, G: 58, R: 76, Prediksi: hitam  B: 57, G: 58, R: 74, Prediksi: hitam  B: 56, G: 58, R: 74, Prediksi: hitam  B: 54, G: 56, R: 71, Prediksi: hitam  B: 54, G: 55, R: 69, Prediksi: hitam  B: 56, G: 57, R: 73, Prediksi: hitam  B: 57, G: 58, R: 75, Prediksi: hitam  B: 56, G: 58, R: 73, Prediksi: hitam  B: 57, G: 58, R: 73, Prediksi: hitam  B: 57, G: 59, R: 76, Prediksi: hitam  B: 55, G: 60, R: 82, Prediksi: hitam  B: 55, G: 57, R: 79, Prediksi: hitam  B: 47, G: 46, R: 57, Prediksi: hitam  B: 42, G: 42, R: 51, Prediksi: hitam  B: 45, G: 45, R: 55, Prediksi: hitam  B: 45, G: 45, R: 56, Prediksi: hitam  B: 41, G: 42, R: 48, Prediksi: hitam |

|  |
| --- |
| **Kode 1**  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 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  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]  # Tampilkan 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 3**  import cv2  import numpy as np  import csv  import time  from sklearn import svm  from sklearn.preprocessing import StandardScaler  import pandas as pd  # 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  except Exception as e:  return "Tidak Teridentifikasi"  return prediction  # 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() |

Program ini terdiri dari tiga kode pemograman, yang pertama ada pembacaan kamera, trs ada pembuatan database warna, dan yang terakhir ada prediksi warna. kode pertama menggunakan OpenCV untuk menangkap dan menampilkan feed kamera secara real-time. Terus di kode yang kedua, sistem menganalisis nilai RGB rata-rata dari area tertentu dalam frame kamera, kemudian menyimpannya ke dalam file database (Databasewarna.txt) berdasarkan label warna yang ditentukan pengguna melalui input keyboard, contohnya kya 2 untuk hijau dan 4 untuk hitam. Yang terakhir kode ketiga memanfaatkan database tersebut untuk melatih model klasifikasi **Support Vector Machine (SVM)** dengan kernel linear, setelah data dinormalisasi menggunakan StandardScaler. Model ini kemudian digunakan untuk memprediksi warna secara real-time berdasarkan input RGB dari kamera.