**PRAKTIKUM FISIKA KOMPUTASI**

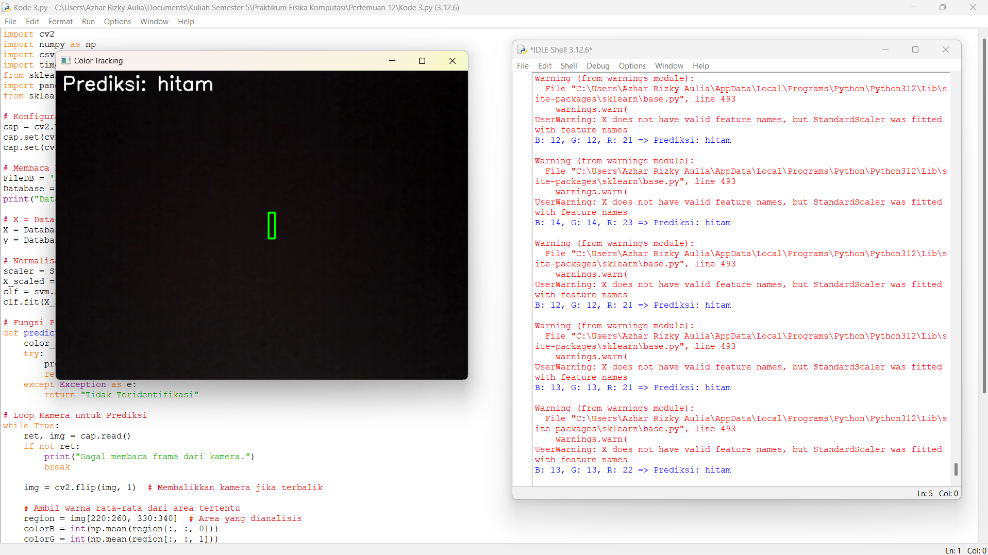
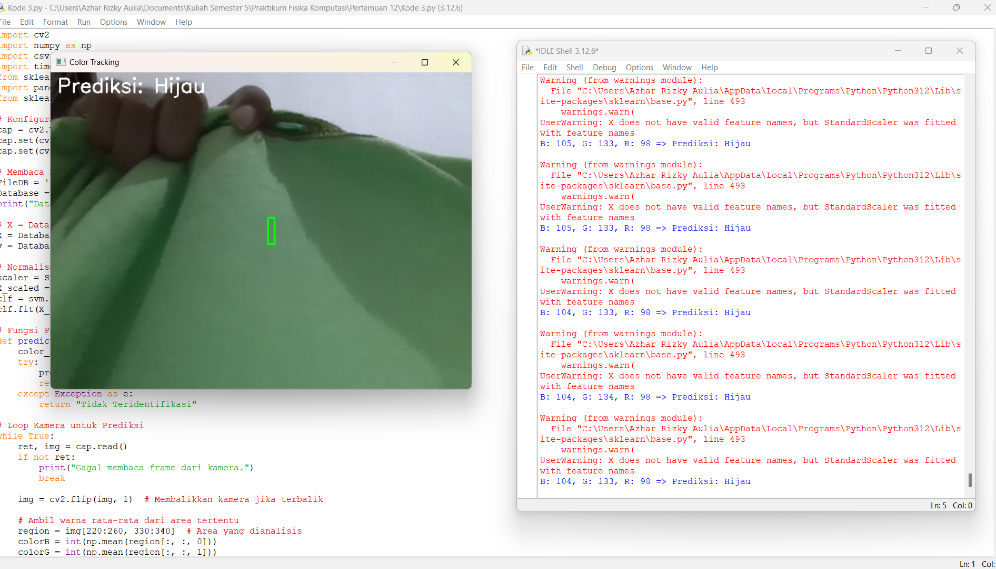
**MODUL 12 OPENCV PREDIKSI WARNA**

**Nama:**

**AZHAR RIZKY AULIA (1227030008)**

1. Buatlah database dengan target warna hijau dan hitam

|  |  |
| --- | --- |
| Database warna hijau 91,144,85,Hijau  86,143,84,Hijau  85,143,85,Hijau  86,143,85,Hijau  86,142,85,Hijau  85,142,84,Hijau  85,142,84,Hijau  85,142,84,Hijau  85,142,84,Hijau  85,142,84,Hijau  85,142,84,Hijau  85,142,84,Hijau  85,142,84,Hijau  86,142,84,Hijau  85,142,85,Hijau  85,142,84,Hijau  85,142,83,Hijau  84,142,83,Hijau  84,142,84,Hijau  85,142,85,Hijau  88,144,87,Hijau  89,145,88,Hijau  89,145,88,Hijau  90,145,88,Hijau  90,145,88,Hijau  89,145,88,Hijau  89,145,88,Hijau  89,144,88,Hijau  90,145,88,Hijau  89,145,88,Hijau  90,145,88,Hijau  89,145,88,Hijau  88,144,87,Hijau  83,138,82,Hijau  78,137,79,Hijau  76,137,79,Hijau  76,137,80,Hijau | Database warna hitam 93,86,93,hitam  92,86,92,hitam  92,86,92,hitam  94,87,94,hitam  94,88,94,hitam  94,88,94,hitam  95,88,94,hitam  95,89,95,hitam  96,89,96,hitam  96,90,96,hitam  97,90,97,hitam  98,91,97,hitam  98,91,97,hitam  98,91,98,hitam  99,92,97,hitam  99,92,97,hitam  99,92,97,hitam  99,92,97,hitam  99,92,97,hitam  99,92,97,hitam  100,92,97,hitam  100,92,96,hitam  100,92,96,hitam  100,92,96,hitam  101,92,96,hitam  102,92,95,hitam  102,93,95,hitam  102,93,95,hitam  101,92,94,hitam  100,93,94,hitam  100,94,95,hitam  101,95,96,hitam  71,64,68,hitam  73,66,70,hitam  74,67,72,hitam  73,67,72,hitam  74,67,72,hitam |

1. Prediksi warna hijau dan hitam
2. 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  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() |

Library cv2 digunakan untuk mengakses kamera dan library numpy digunakan untuk membuat array. Kemudian variael cap berfungsi untuk membuka kamera bawaan laptop dengan ID (0). Kemudian dibuat sebuah perulangan untuk membaca frame dari kamera dan menampilkan frame yang telah di proses, dan jika tombol esc ditekan maka perulangan akan dihentikan dan program akan menutup kamera.

Kode program 2

|  |
| --- |
| import csv  import cv2  import numpy as np  # 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"File {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 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 == 27:  # ESC          break      elif key == ord('1'):          label = "Merah"      elif key == ord('2'):          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() |

Dimulai dengan import library yang diperlukan seperti cv2 untuk mengakses kamera, numpy untuk membuat array, dan csv supaya file disimpan dalam csv. Kemudian dilanjutkan dengan membuka kamera bawaan laptop dengan variabel cap menggunakan ID (0) dan mengatur resolusi menjadi 480x360 piksel. Kemudian variabel FileDB berisi nama file dan format untuk database warna dan variabel header adalah nama kolom data didalam file. Kemudian ditentukan area analisis warna pada koordinat tertentu dan menghitung rata rata warna biru, hijau, dan merah pada area. Kemudian area analisis warna ditampilkan dan ditampilkan juga nilai rata-rata warna pada layer. Kemudian dibuat logika sesuai dengan input yang diberikan saat kode program dijalankan, misalnya 1 untuk merah kemudian data warna (B,G,R) dan label ditambahkan ke file database. Kemudian jika tombol esc ditekan maka pengambilan data berakhir dan kamera akan langsung ditutup.

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  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()  # Normalisasi data  X\_scaled = scaler.fit\_transform(X)  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() |

Dimulai dengan import library yang diperlukan seperti cv2 untuk mengakses kamera, numpy untuk membuat array, svm untuk melatih dan menjalankan model SVM, dan pandas untuk membaca file CSV. Kemudian terdapat variabel dengan nama Database yang berfungsi untuk membaca database warna dari file CSV, lalu variabel X merupakan data input yaitu B,G,R dan variabel y adalah data target atau nama warna. Kemudian model SVM dilatih dengan kernel linear. Kemudian terdapat fungsi dengan nama predict\_color untuk memprediksi warna berdasarkan nilai B, G, R. Kemudian dilakukan perulangan untuk membuat area analisis warna pada daerah tertentu dan dilakukan prediksi warna menggunakan fungsi predict\_color. Kemudian hasil prediksi tersebut ditampilkan di frame kamera.

Proses prediksi warna diawali dengan kode program membaca data dari kamera, dan mengambil area kecil sebagai area untuk menganalisis warna. Area ini dibuat untuk menghitung nilai rata-rata dari biru, merah, dan hijau. Setelah mendapatkan nilai rata-rata warna, data tersebut kemudian diolah menggunakan StandardScaler supaya format data lebih mudah dibaca saat model SVM dilatih. Kemudian data yang telah diolah ini dimasukkan ke model SVM yang telah dilatih menggunakan database warna. Model SVM akan membandingkan data baru dengan nilai-nilai warna yang ada dalam database untuk menentukan warna yang paling mendekati. Kemudian hasil prediksi ditampilkan secara langsung didalam frame kamera.