

Praktikum Fisika Komputasi

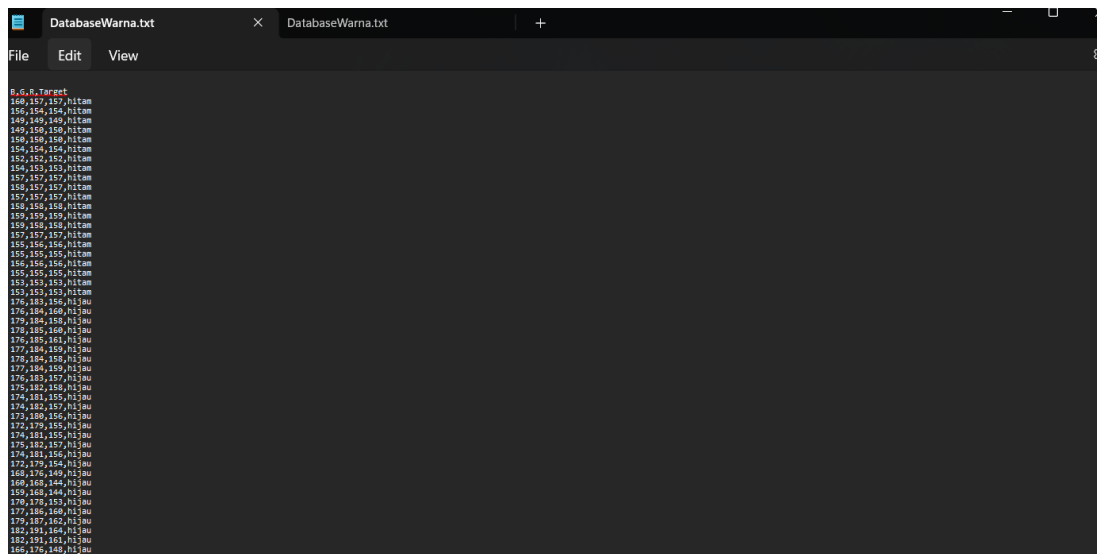
Opencv Prediksi Warna (Tugas 12)

Ramli Zhafran Amarillo (1227030027)

1.) Database warna Hijau dan Hitam

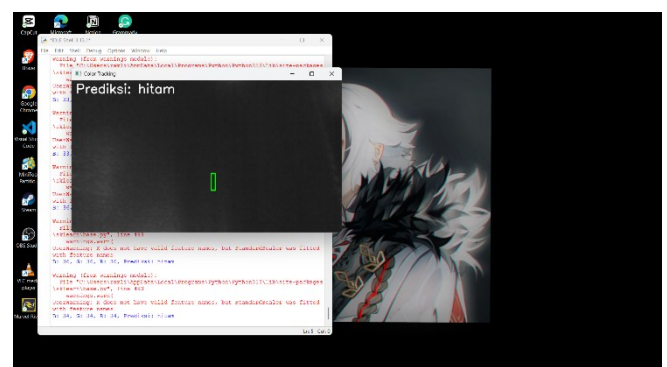
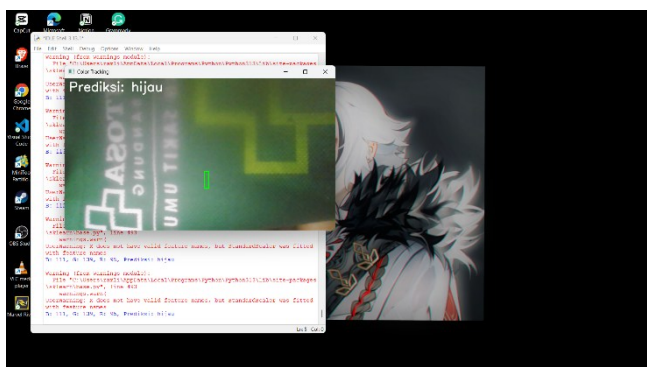


DatabaseWarna.txt



2.) Prediksi warna Hitam dan Hijau

Sesuai untuk mengkategorikan warna saat dicoba



3.) Penjelasan Code dan Prosesnya

Code 1 (Membuat 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

    # 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()

```

Code 2 (Test Database)

```

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()

```

Program dimulai dengan menginisialisasi kamera menggunakan `cv2.VideoCapture(0)` untuk mengambil gambar. Data warna yang telah dilabeli dibaca dari file CSV menggunakan **Pandas** (`pd.read_csv()`), yang berisi kolom BGR (Blue, Green, Red) sebagai fitur dan target warna yang telah dilabeli. Data BGR kemudian diolah dengan `StandardScaler()` untuk memastikan nilainya pada skala yang sama. Setelah itu, model **SVM dengan kernel linear** dilatih menggunakan data yang telah diolah mempelajari model ini dengan cara mengenali pola warna berdasarkan nilai BGR dan label yang sesuai.

Selanjutnya, program terus berjalan untuk membaca frame dari kamera. Setiap frame dianalisis dengan menghitung rata-rata nilai BGR dari area tertentu dan di sini pada code programnya koordinatnya adalah antara (220:260, 330:340)). Nilai rata-rata BGR ini dikirim ke fungsi `predict_color()`, yang akan melakukan prediksi menggunakan model SVM yang telah dilatih. Hasil prediksi warna ditampilkan pada gambar yang ditangkap dengan kotak yang menunjukkan area analisis dan teks yang menunjukkan warna yang diprediksi. Proses ini berulang terus hingga pengguna menekan tombol ESC untuk keluar dari program.