PRAKTIKUM FISIKA KOMPUTASI

MODUL -12 OPENCV PREDIKSI WARNA

Disusun Oleh:

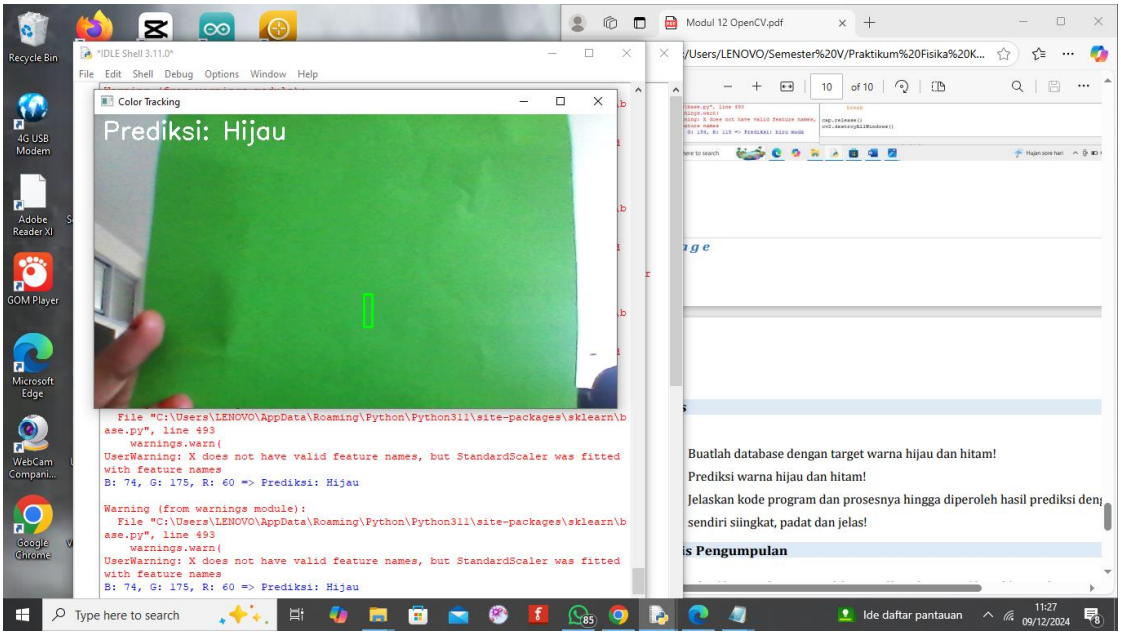
Dewi Rahmawati (1227030010)

1. Pada kode program data dibuat data foto untuk warna hitam dan hijau sehingga diproleh datanya yaitu:

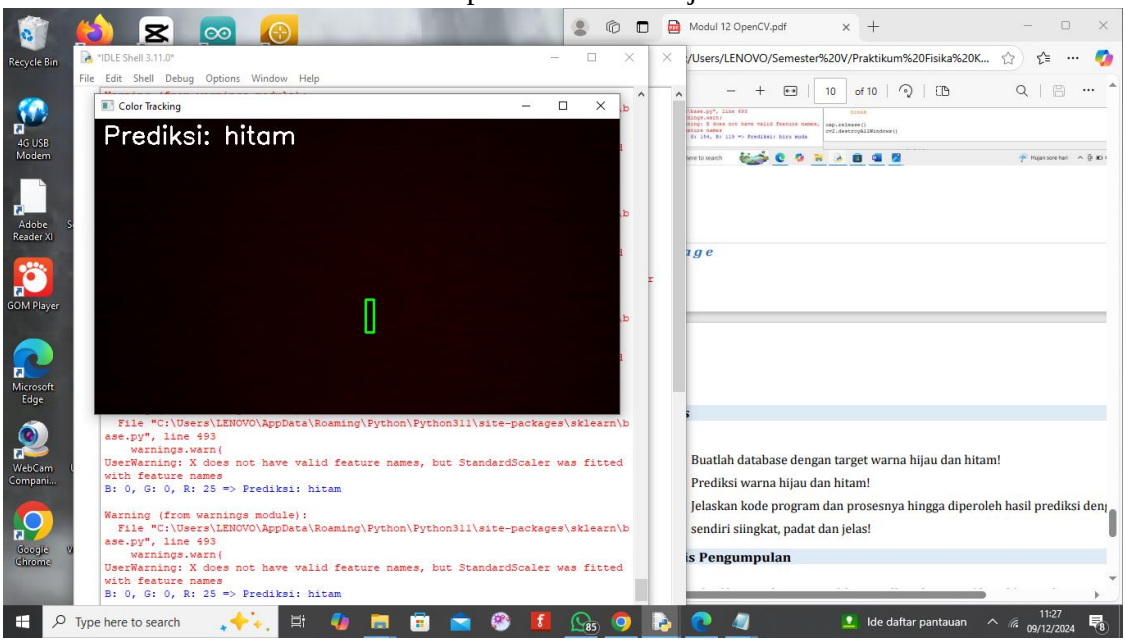
B,G,R,Target
50,128,37,Hijau
47,125,35,Hijau
50,130,38,Hijau
50,130,38,Hijau
50,129,37,Hijau
50,131,38,Hijau
54,135,40,Hijau
56,139,42,Hijau
55,139,42,Hijau
56,139,43,Hijau
57,139,43,Hijau
56,139,41,Hijau
57,138,42,Hijau
58,141,42,Hijau
54,139,41,Hijau
57,140,42,Hijau
55,138,40,Hijau
56,138,40,Hijau
56,139,41,Hijau
55,137,40,Hijau
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam
0,0,0,hitam

0,0,0,hitam 0,0,0,hitam 0,0,0,hitam 0,0,0,hitam 0,0,0,hitam 0,0,0,hitam

2. Pada kode program prediksi diuji hasil dari data yang dibuat sehingga hasilnya dapat dilihat:



hasil prediksi warna hijau



hasil prediksi warna hitam

3. Kode program untuk pengambilan data

```
import csv
import cv2
import numpy as np
```

Pada kode program pengambilan data langkah pertama yaitu megimport library yang digunkan. library csv digunakan untuk menulis dan membaca CSV. Library cv2 berfungsi untuk akses kamera dan manipulasi gambar dengan data terdekat. kemudian ibrary numpy digunakan untuk memperhitungkan rata-rata nilai komposisi warna yang dihasilkan.

```
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 480)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 360)
```

Bagian kode program ini merupakan bagian untuk menkonfigurasi kamera. Kamera dikonfigurasi dengan diakses menggunakan **`cv2.VideoCapture`**. untuk nilai 0 nya menunjukkan kamera default dan cap.set merupakan pengaturan untuk resolusi kamera yaitu 400x360.

```
FileDB = 'DatabaseWarna.txt'

# Header untuk file CSV
header = ['B', 'G', 'R', 'Target']
```

Kode ini enunjukkan nama file data yang diambil dalam bentuk txt dan judul pada data yang disimpan.

```
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")
```

Kode proram ini mennjukkan jika file data warna yang akan disimpan ternyata sudah tersedia, maka akan melakuan penambahan data pada warna yang belum ada. Pada tampilan akan diprint kode angka beserta perintah warnanya. Angka 1 untuk menyimpan data berwarna merah, angka 2 untuk warna hijau, 3 untuk warna biru, 4 untuk warna hitam, 5 untuk warna kuning, 6 ntuk warna putih, dan ESC untuk keluar dari pindaian kamera untuk menginput data warna.

```
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
```

Kode program ini menunjukkan kamera akan memindai warna secara real-time. Akan tetapi, jika kamera gagal membaca frame, maka program keluar dari loop atau akan menghentikan pemindaian.

```
        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]
```

Kode program ini menunjukkan bahwa saat pemindaian warna akan ada kotak kecil yang menjadi titik pemindaian warna . Koordinat pemindaian warna ini berada di koordinat (330, 220) hingga (340, 260). Kemudian pada proses pemindaian akan dilihat nilai rata-rata warna biru (B), hijau (G), dan merah (R) menggunakan **np.mean.**

```
        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)
```

Kode program ini akan membantu menganalisis data warna yang terekam oleh kamera di area kotak koordinat.

```
        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
```

Kode program ini berisi kodeuntuk setiap warna yang akan disimpan.]Angka 1 untuk menyimpan data berwarna merah, angka 2 untuk warna hijau, 3 untuk warna biru, 4 untuk warna hitam, 5 untuk warna kuning, 6 ntuk warna putih, dan ESC untuk keluar dari pindaian kamera untuk menginput data warna.

```
        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 diatas menunjukkan perintah untuk menyimpan data yang dihasilkan dari pemindaian dan menutup kamera ketika pemindaian sudah diakhiri.

Selanjutnya yaitu kode program untuk memindai atau memprediksi warna.

```
import cv2
import numpy as np
import csv
import time
from sklearn import svm
import pandas as pd
from sklearn.preprocessing import StandardScaler
```

Pada bagian pertama yaitu library yang digunakan. Library cv2 digunakan untuk mengakses kamera. Library numpy digunakan untuk operasi numerik seperti menghitung rata-rata nilai warna. Library csv digunakan untuk membaca file CSV. Libray time digunakan untuk menambahkan jeda untuk memindai warna yang sesuai dengan data. Svm ari library sklearn untuk menormalisasikan data warna. Library pandas digunakan untuk membaca database dari file CSV yang telah tersedia.

```
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 480)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 360)
```

Bagian kode program ini merupakan bagian untuk menkonfigurasi kamera. Kamera dikonfigurasi dengan diakses menggunakan **`cv2.VideoCapture`**. untuk nilai 0 nya menunjukkan kamera default dan cap.set merupakan pengaturan untuk resolusi kamera yaitu 400x360.

```
FileDB = 'DatabaseWarna.txt'  # Pastikan file ini tersedia dan formatnya benar
Database = pd.read_csv(FileDB, sep=",", header=0)
print("Database:\n", Database)
```

Kode program ini digunakan untuk membaca data yang yang telah disimpan dalam Database warna.txt.

```
# X = Data (B, G, R), y = Target
X = Database[['B', 'G', 'R']]
y = Database['Target']
```

Kode program ini menunjukkan input nilai x sebagai database koordinat data dan y sebagai target data warna.

```
scaler = StandardScaler()  # Normalisasi data
X_scaled = scaler.fit_transform(X)
clf = svm.SVC(kernel='linear')  # Gunakan kernel linear
clf.fit(X_scaled, y)
```

Kode program ini menunjukan normalisasi data warna RGB menggunakan **StandardScaler**, sehingga setiap fitur memiliki rata-rata 0 dan standar deviasi 1. Kernel linear digunakan untuk klasifikasi.

```
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"
```

Kode program ini berisi fungsi untuk menerima nilai RGB, menormalkan data menggunakan scaler, dan memprediksi warna menggunakan model **SVM**. Jika terjadi error (misalnya input tidak valid), fungsi mengembalikan "Tidak Teridentifikasi".

```
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
```

Kode program ini menunjukkan bahwa kamera membaca pemindaian secara real-time pada frame. Jika pembacaan gagal, maka kode program keluar dari loop.

```
    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]
```

Kode program ini menunjukkan bahwa saat pemindaian warna akan ada kotak kecil yang menjadi titik pemindaian warna . Koordinat pemindaian warna ini berada di koordinat (330, 220) hingga (340, 260). Kemudian pada proses pemindaian akan dilihat nilai rata-rata warna biru (B), hijau (G), dan merah (R) menggunakan **np.mean.**

```
   prediction = predict_color(colorB, colorG, colorR)

  print(f"B: {colorB}, G: {colorG}, R: {colorR} => Prediksi: {prediction}")
```

Kode program ini menunjukkan hasil prediksi yang dihasilkan pada saat pemindaian warna.

```
    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)
```

Kode program ini akan membantu menganalisis data warna yang terekam oleh kamera di area kotak koordinat.

```
k = cv2.waitKey(30) & 0xff

    if k == 27:  # Tekan ESC untuk keluar

        break

cap.release()

cv2.destroyAllWindows()
```

Kode program diatas menunjukkan perintah untuk keluar dari pemindaian setelah menekan ESC.

Proses dalam prediksi warna ini diawali dengan pemindaian data yang dilakukan dengan memposisikan warna pada kamera sehingga ada warna pada rectangle dan dikli kode angka untuk warna yang ditambahkan pada data yang dipindai hanya hijau dan hitam sehingga diklik angka 2 untuk warna hijau dan angka 4 untuk warna hitam. Data warna tersebut akan disimpan pada format txt dengan nama Database warna dengan header BGRT. Pada kode

program prediksi kamera akan memindai warna pada kamera diarea rectangle. Lalu pada roses pemindaian akan dimbil hasil data yang telah dibuat sebelumnya dan akan dicocokan untuk menampikan warna yang disesuaikan dengan data.

Kode Program Database

```
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()
```

Kode program Prediksi

```
        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()
```