

Laporan Praktikum Kontrol Cerdas

Nama : Muhamad Azril Insanul Huda

NIM : 224308012

Kelas : TKA-6A

Akun Github (Tautan) : <https://github.com/azl1254>

Student Lab Assistant : Muhammad Mahirul Faiq (214308043)

1. Judul Percobaan

Deteksi Warna Real Time Dengan Implementasi KNN

2. Tujuan Percobaan

Tujuan dari praktikum ini yaitu :

1. Mahasiswa dapat mengimplementasikan KNN untuk deteksi Warna dengan menggunakan VS code atau python.
2. Mahasiswa dapat mengintegrasikan machine learning dengan computer vision

3. Landasan Teori

Istilah *Machine Learning* pertama kali diperkenalkan oleh Arthur Samuel pada tahun 1959 dalam konteks permainan catur dengan mesin. Secara definisi, *Machine Learning* merupakan model komputasi statistik yang berfokus pada prediksi dengan menggunakan komputer. Algoritma dalam *Machine Learning* membangun model matematika berdasarkan data sampel, yang dikenal sebagai *data training* atau data pelatihan, untuk membuat prediksi atau mengambil keputusan tanpa perlu diprogram secara eksplisit. Alur kerja utama dari *Machine Learning* melibatkan pemrosesan data mentah sebagai input untuk menghasilkan prediksi sebagai output. Secara umum, algoritma *Machine Learning* dikategorikan ke dalam tiga jenis utama, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*.

Citra dapat diartikan sebagai gambar, foto, atau tampilan dua dimensi yang merepresentasikan visualisasi suatu objek. Citra bisa berbentuk fisik (tercetak) maupun digital. Citra digital sendiri merupakan susunan angka-angka dalam dua dimensi. Citra digital disimpan dalam bentuk array angka digital yang berasal dari proses kuantisasi tingkat kecerahan setiap piksel penyusunnya. Dari sudut pandang matematika, citra adalah fungsi kontinu yang merepresentasikan intensitas cahaya pada bidang dua dimensi. Ketika sumber cahaya menyinari objek, sebagian cahaya dipantulkan oleh objek tersebut. Pantulan cahaya ini kemudian ditangkap oleh alat optik seperti mata manusia, kamera, scanner, dan lain-lain, sehingga terbentuklah bayangan objek yang disebut citra. Citra digital

adalah gambar dua dimensi $f(x,y)$ yang diperoleh melalui proses diskritisasi koordinat spasial (sampling) dan tingkat kuantisasi (kecemerlangan atau keabuan). Citra ini dapat ditampilkan di layar komputer dalam bentuk nilai digital diskrit yang disebut piksel. Pengolahan Citra Digital adalah proses pengolahan dan analisis gambar yang menggunakan teknik untuk mengolah, memanipulasi, dan memodifikasi citra, dengan fokus pada persepsi visual. Dalam proses ini, data masukan dan informasi keluarannya berbentuk citra. Secara umum, pengolahan citra digital diartikan sebagai pemrosesan gambar dua dimensi menggunakan komputer. Tujuan utamanya adalah meningkatkan kualitas gambar agar terlihat lebih jelas.

Model warna HSV (Hue, Saturation, Value) merupakan salah satu model warna selain RGB. Hue merepresentasikan warna dasar, seperti merah, ungu, dan kuning. Hue digunakan untuk membedakan warna serta menentukan nuansa seperti kemerahan atau kehijauan pada cahaya. Hue berkaitan dengan panjang gelombang cahaya. Saturation menunjukkan tingkat kemurnian warna, yang menggambarkan seberapa banyak warna putih yang dicampurkan ke dalam warna tersebut. Sementara itu, Value merepresentasikan tingkat kecerahan yang terlihat oleh mata, tanpa memandang warnanya. Seperti saturation, value memiliki rentang dari 0 hingga 1. Jika nilai value adalah 0, warna tersebut akan menjadi hitam. Sebaliknya, jika value bernilai 1, tidak ada kandungan warna hitam dalam warna tersebut.

Python cv2 dan numpy adalah pustaka pemrograman yang digunakan untuk visi komputer secara waktu nyata. Cv2 berfungsi sebagai pustaka pengolahan citra untuk berbagai jenis objek, baik objek manusia maupun objek benda lainnya, sedangkan numpy digunakan untuk komputasi dan perhitungan numerik. Dengan menggabungkan kedua pustaka ini, proses deteksi objek pada citra dapat dilakukan dengan lebih cepat dan akurat melalui iterasi numerik yang efisien.

4. Analisis dan Diskusi

Analisis:

Terdapat 2 program dalam praktikum ini yang pertama digunakan untuk melatih model K-Nearest Neighbors (KNN) agar dapat mengenali warna berdasarkan nilai RGB. Data warna dibaca dari dataset, kemudian dinormalisasi agar lebih seimbang. Setelah itu, model dilatih dan diuji sebelum akhirnya disimpan untuk digunakan dalam aplikasi lain. Akurasi model sangat bergantung pada kualitas dan jumlah data warna yang tersedia.

Program kedua menggunakan model KNN yang telah dilatih untuk mendeteksi warna secara real-time melalui kamera. Program ini mengambil warna dari titik tengah layar, melakukan normalisasi, lalu memprediksi warna menggunakan model KNN. Hasil warna ditampilkan pada layar, dan program berjalan hingga pengguna menekan tombol 'q' untuk keluar.

Diskusi:

1. Kelebihan:

- kesederhanaannya dan kemampuannya untuk mengenali warna secara langsung dari kamera.

Kekurangan:

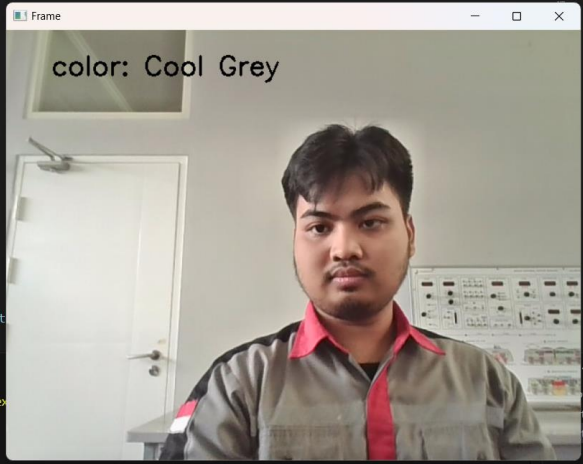
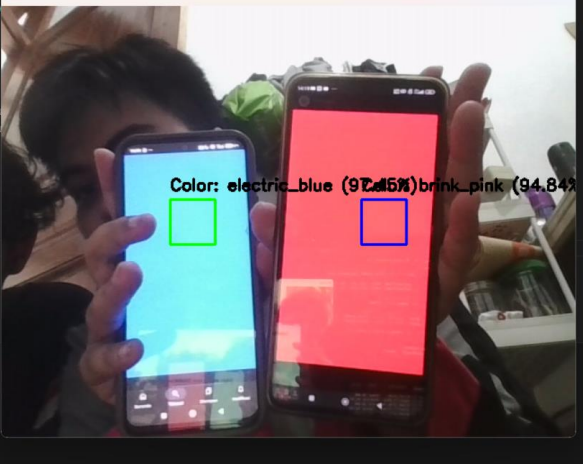
- ketergantungan pada dataset warna, pengaruh pencahayaan terhadap hasil deteksi.

2. Untuk meningkatkan program ini, beberapa pengembangan yang dapat dilakukan adalah menambah dataset warna agar prediksi lebih akurat, menggunakan metode lain seperti SVM atau Deep Learning, serta membuat tampilan GUI yang lebih menarik.

5. Assignment

Pada assignment ini merubah program yang awalnya mendeteksi warna pada Tengah frame dang menggunakan KNN dirubah menjadi SVM serta menambahkan fitur pada program yaitu penambahan 2 pendeteksi warna berbeda dalam waktu yang sama dan akurasi warna yang tertangkap oleh frame dan ada perubahan Dimana dataset yang digunakan juga berbeda yang di implementasikan pada program ini .

6. Data dan Output Hasil Pengamatan

N0	Variabel	Hasil Pengamatan
1	KNN	
2	SVM	

7. Kesimpulan

Dari percobaan yang telah dilakukan dapat disimpulkan bahwa:

1. Program ini berhasil mengenali warna berdasarkan nilai RGB menggunakan K-Nearest Neighbors (KNN).
2. Model yang telah dilatih dapat digunakan untuk deteksi warna real-time dengan kamera.
3. Hasil deteksi masih dipengaruhi oleh beberapa faktor, seperti pencahayaan yang bisa mengubah warna yang tertangkap kamera.

8. Saran

1. Menambah dataset warna agar model bisa mengenali lebih banyak variasi warna dengan akurasi lebih tinggi.
2. Mengatasi pengaruh pencahayaan dengan teknik color correction atau preprocessing tambahan.

9. Daftar Pustaka

Dedy Agung Prabowo et al. (2018). *DETEKSI DAN PERHITUNGAN OBJEK BERDASARKAN WARNA MENGGUNAKAN COLOR OBJECT TRACKING*

Ahmad Husna Ahadi et al. (2024). *IMPLEMENTASI SISTEM PENDETEKSI WARNA OBJEK DENGAN OPENCVPYTHON*

Resky Ramadhandi Santoso et al. (2020) *IMPLEMENTASI METODE MACHINELEARNINGMENGGUNAKAN ALGORITMA EVOLVINGARTIFICIALNEURAL NETWORK PADA KASUS PREDIKSIDIAGNOSIS DIABETES*
