

# Laporan Praktikum Kontrol Cerdas

Nama : Widhi Widya Hastuti

NIM : 224308025

Kelas : TKA 6A

Akun Github (Tautan) : widhiwidya

Student Lab Assistant : Yulia Brilianty

## 1. Judul Percobaan

Penerapan Machine Learning untuk Deteksi Warna Real-Time Menggunakan KNN dan SVM dengan OpenCV.

## 2. Tujuan Percobaan

1. Memahami dasar-dasar Machine Learning dalam sistem kendali
2. Mengimplementasikan model ML sederhana untuk klasifikasi objek
3. Menggunakan Scikit-learn untuk membuat model ML dasar
4. Mengintegrasikan model ML dengan Computer Vision untuk deteksi objek

## 3. Landasan Teori

Machine Learning (ML) adalah teknologi yang memungkinkan komputer belajar dari data tanpa harus diprogram secara langsung. Dalam sistem kendali, ML membantu dalam membuat keputusan yang lebih baik, menyesuaikan respons sistem secara otomatis, dan mengatasi masalah yang sulit diselesaikan dengan cara biasa. Dengan ML, sistem kendali bisa menjadi lebih cerdas dan efisien dalam mengelola berbagai tugas.

Salah satu penggunaan dasar ML adalah untuk mengenali dan mengklasifikasikan objek. Misalnya, sebuah sistem bisa dilatih untuk membedakan antara mobil dan sepeda berdasarkan bentuk dan warnanya. Model sederhana seperti K-Nearest Neighbors (KNN) atau Decision Tree sering digunakan untuk tugas ini karena mudah dipahami dan diterapkan.

Scikit-learn adalah pustaka (library) dalam bahasa pemrograman Python yang sangat berguna untuk membangun model ML dengan mudah. Dengan Scikit-learn, kita bisa mengolah data, melatih model, dan mengukur seberapa baik model tersebut bekerja. Pustaka ini

menyediakan berbagai alat yang memungkinkan pengguna, termasuk pemula, untuk melakukan eksperimen dengan berbagai algoritma ML tanpa harus membangun semuanya dari awal.

ML juga dapat dikombinasikan dengan Computer Vision untuk mengenali objek dalam gambar atau video. Contohnya, sebuah kamera keamanan bisa menggunakan ML untuk mendeteksi apakah ada orang atau kendaraan di dalam rekaman. Model yang sering digunakan untuk tugas ini termasuk Convolutional Neural Network (CNN) dan metode klasik seperti Histogram of Oriented Gradients (HOG). Dengan teknologi ini, sistem bisa secara otomatis mengenali objek dengan lebih cepat dan akurat dalam berbagai aplikasi, seperti keamanan, kendaraan otonom, dan pengenalan wajah.

#### **4. Analisis dan Diskusi**

##### **Analisis**

- Bagaimana performa model dalam mendeteksi warna?

Performa model dalam mendeteksi warna bergantung pada kualitas data pelatihan dan pencahayaan saat pengambilan gambar. Dalam kode ini, model K-Nearest Neighbors (KNN) digunakan untuk mengenali warna berdasarkan nilai RGB dari piksel tengah gambar. Nilai ini kemudian dinormalisasi sebelum diproses oleh model. Jika dataset yang digunakan cukup luas dan beragam, maka model dapat memberikan hasil prediksi warna yang cukup akurat, terutama untuk warna yang jelas perbedaannya.

- Bagaimana perbedaan akurasi jika jumlah dataset ditambah?

Jika jumlah dataset ditambah, akurasi model dalam mendeteksi warna biasanya akan meningkat karena model memiliki lebih banyak. Dengan dataset yang lebih besar dan beragam, model K-Nearest Neighbors (KNN) dapat mengenali lebih banyak variasi warna serta perubahan pencahayaan, sehingga hasil prediksi menjadi lebih akurat. Selain itu, dataset yang lebih luas dapat mengurangi kesalahan dalam klasifikasi warna. Namun, penambahan dataset harus tetap berkualitas dan mencakup berbagai kondisi pencahayaan agar benar-benar membantu model belajar dengan baik. Jika data yang ditambahkan tidak relevan atau kurang bervariasi, model bisa mengalami kesulitan membedakan warna dengan benar. Selain itu, karena KNN membandingkan data baru dengan seluruh dataset, jika jumlah data terlalu besar, proses prediksi bisa menjadi lebih lambat. Oleh karena itu, jika dataset terus bertambah, mungkin perlu mempertimbangkan metode lain seperti Support Vector Machine (SVM) atau Neural Networks agar hasil prediksi tetap cepat dan akurat.

- Bagaimana cara meningkatkan kinerja model klasifikasi?

Pertama, menambah jumlah dan variasi dataset agar model dapat mengenali lebih banyak warna dalam berbagai kondisi pencahayaan. Kedua, peningkatan kontras untuk membantu model membedakan warna dengan lebih jelas. Ketiga, mencoba model lain yang lebih canggih, seperti Support Vector Machine (SVM) atau Neural Networks, yang dapat memberikan hasil lebih akurat dibanding K-Nearest Neighbors (KNN) dalam dataset yang besar.

## Diskusi

- Apa keuntungan Machine Learning dibandingkan metode berbasis aturan (rule-based)?

Machine Learning lebih fleksibel dibanding metode rule-based karena tidak memerlukan aturan yang dibuat secara manual. Model dapat belajar dari data dan menyesuaikan diri dengan perubahan, sedangkan metode rule-based harus diperbarui secara manual jika ada kondisi baru. Selain itu, Machine Learning lebih akurat dalam mengenali pola yang rumit, seperti gambar atau suara, dan lebih tahan terhadap data yang tidak sempurna atau bervariasi.

- Bagaimana ML dapat diintegrasikan lebih lanjut dalam sistem kendali?

Machine Learning bisa membuat sistem kendali lebih cerdas dengan belajar dari data dan menyesuaikan respons secara otomatis. ML dapat digunakan untuk memprediksi kondisi, mengoptimalkan keputusan, dan beradaptasi dengan perubahan lingkungan. Misalnya, dalam robotik atau kendaraan otonom, ML membantu sistem mengenali pola dan mengambil keputusan lebih cepat. Jika dikombinasikan dengan sensor, computer vision, atau IoT, sistem kendali bisa bekerja lebih akurat dan efisien tanpa perlu aturan yang dibuat secara manual.

- Apa saja tantangan dalam penerapan ML dalam sistem real-time?

Tantangan ML dalam sistem real-time termasuk keterbatasan komputasi, karena butuh pemrosesan cepat di perangkat terbatas. Latensi rendah juga penting agar sistem bisa merespons dengan cepat, seperti di kendaraan otonom. Selain itu, kualitas data harus baik agar model tetap akurat di berbagai situasi. Keandalan dan keamanan juga krusial, terutama di sistem kritis seperti medis atau otomotif. Terakhir, sistem harus bisa beradaptasi dengan perubahan lingkungan tanpa kehilangan akurasi.

## 5. Assignment

Program ini digunakan untuk mendeteksi warna secara real-time menggunakan kamera. Dengan bantuan OpenCV, program ini menangkap gambar dari kamera dan mengubahnya ke format HSV agar lebih mudah mengenali warna. Program ini juga menggunakan model SVM

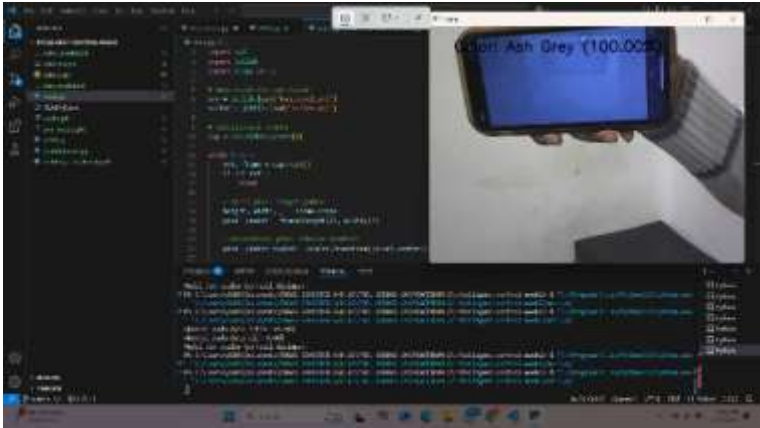
(Support Vector Machine) yang telah dilatih sebelumnya untuk mengklasifikasikan warna berdasarkan nilai RGB. Dataset warna yang tersimpan dalam file colors.csv digunakan sebagai referensi untuk mencocokkan warna yang terdeteksi dengan nama warna yang paling mendekati.

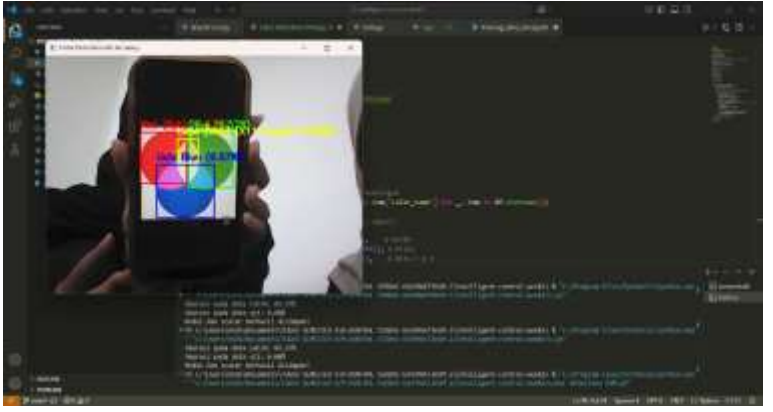
Dalam prosesnya, program mencari objek berwarna dalam gambar dengan menggunakan rentang warna tertentu, seperti merah, hijau, biru, dan kuning. Ketika program menemukan objek yang sesuai, ia akan menggambar kotak di sekitar objek tersebut dan menampilkan nama warnanya. Untuk memastikan keakuratan deteksi, program membandingkan warna yang diprediksi oleh model SVM dengan warna yang ada dalam dataset menggunakan metode perhitungan jarak Euclidean. Program juga menghitung persentase akurasi dari setiap prediksi yang dilakukan.

Hasil deteksi ditampilkan langsung di layar dengan objek yang telah diberi kotak berwarna dan teks yang menunjukkan warna serta tingkat akurasinya. Program akan terus berjalan hingga pengguna menekan tombol 'q' untuk menghentikan proses dan menutup kamera. Dengan cara ini, program bisa digunakan dalam berbagai keperluan, seperti pengenalan warna dalam sistem otomatis atau pemantauan visual berbasis warna.

6. Data dan Output Hasil Pengamatan

Sajikan data dan hasil yang diperoleh selama percobaan. Gunakan tabel untuk menyajikan data jika diperlukan.

No	Variabel	Hasil Pengamatan
1	K-Nearest Neighbors (KNN)	

2	SVM Menggunakan csv datasets	
---	------------------------------	-----------------------------------------------------------------------------------

## 7. Kesimpulan

program ini adalah bahwa keduanya mampu mendeteksi warna secara real-time menggunakan kamera dengan bantuan model pembelajaran mesin. Model K-Nearest Neighbors (KNN) dan Support Vector Machine (SVM) digunakan untuk mengklasifikasikan warna berdasarkan data yang telah dilatih sebelumnya. Program dengan SVM juga menggunakan metode segmentasi warna dalam format HSV untuk mendeteksi objek berwarna dalam gambar. Hasil deteksi ditampilkan langsung di layar, lengkap dengan informasi warna yang terdeteksi serta tingkat akurasi atau probabilitasnya.

Meskipun kedua program bekerja dengan baik, akurasi deteksi masih bergantung pada faktor seperti pencahayaan dan kualitas data latih. Program dengan SVM menggunakan metode Euclidean Distance untuk membandingkan warna yang diprediksi dengan dataset referensi, sementara model KNN menampilkan tingkat kepercayaan terhadap hasil prediksi. Untuk meningkatkan akurasi dan stabilitas, sistem ini dapat dikembangkan lebih lanjut dengan menyesuaikan rentang warna HSV, meningkatkan kualitas dataset, atau menggunakan metode pembelajaran mesin yang lebih canggih.

## 8. Saran

- Menambah jumlah dan variasi data latih, termasuk sampel dengan pencahayaan yang berbeda.
- Menyesuaikan rentang warna dalam format HSV agar lebih optimal dan tidak terlalu sensitive terhadap perubahan cahaya

## 10. Daftar Pustaka

- Andrekhya, M. Z., & Putra, R. A. (2021). *Deteksi Warna Manggis Menggunakan Pengolahan Citra dengan OpenCV Python*. Voteteknika (Vocational Teknik Elektronika dan Informatika), 9(4), 1-10.
- Ahadi, A. H., Gustina, G., Syawal, M. F., Aminuddin, F. H., & Anzari, Y. (2024). *Implementasi Sistem Pendeteksi Warna Objek dengan OpenCV-Python*. SENTRI: Jurnal Riset Ilmiah, 3(7), 3573-3578.