

Laporan Praktikum Kontrol Cerdas

Nama : Permata Luthfiya Akram
NIM : 224308018
Kelas : TKA 6A
Akun Github (Tautan) : <https://github.com/permatalaa>
Student Lab Assistant : Yulia Brilianty

1. Judul Percobaan

Penerapan Machine Learning menggunakan model SVM dan KNN untuk mendeteksi warna dan akurasi secara Real Time pada software OpenCV.

2. Tujuan Percobaan

Tujuan percobaan dari praktikum ini adalah sebagai berikut:

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.
5. Mengelola dataset dan melakukan pelatihan model sederhana.

3. Landasan Teori

Machine Learning adalah salah satu cabang dari *Artificial Intelligent* (kecerdasan buatan) yang membuat system bisa mengadaptasi kemampuan manusia untuk belajar. *Machine learning* memungkinkan perangkat komputer mempelajari data dan pola tanpa adanya peng-*coding*-an yang khusus. Sistem komputer belajar dari data, pola, dan perilaku pengguna dan membuat keputusan sesuai kebutuhan dengan intervensi manusia yang minimal (Ramadhani & Sembiring, 2022). Pada percobaan ini juga menggunakan Scikit-learn. Scikit Learn difokuskan pada *Machine Learning*, misalnya pemodelan data. Ini tidak melihat bagaimana proses pemuatan, penanganan, manipulasi, dan visualisasi data. *Library* yang ada di balik Scikit-learn adalah NumPy, SciPy, Matplotlib, Ipython, SymPy, dan Pandas (Gifari & Widya, 2020).

Computer vision adalah bidang ilmu komputer yang berfokus pada pembuatan sistem digital yang dapat memproses, menganalisis, dan memahami data visual (gambar atau video) dengan cara yang sama seperti yang dilakukan manusia. *Computer vision* berfungsi untuk memahami konten gambar digital yang mungkin melibatkan penggalan deskripsi dari gambar, yang mungkin berupa objek, deskripsi teks, model tiga dimensi (3D), dan sebagainya (Mulyawan, 2025). *Computer vision* adalah teknologi yang memungkinkan komputer untuk memahami dan menganalisis gambar atau video secara otomatis. *Computer vision* memiliki beberapa langkah utama dalam cara kerjanya, yaitu pengolahan data visual dengan mengumpulkan data visual dari berbagai sumber, seperti kamera, sensor, atau basis data gambar. Setelah gambar diproses, sistem akan melakukan ekstraksi fitur untuk mendeteksi elemen penting dalam gambar, seperti bentuk, warna, dan tekstur. Dan terakhir setelah fitur diekstraksi, algoritma *machine learning* atau *deep learning* digunakan untuk mengenali dan menginterpretasikan objek yang ada dalam gambar. Sistem akan mencocokkan pola yang telah dipelajari sebelumnya dengan data baru untuk menentukan kategori atau identitas objek (Nugroho, 2025).

4. Analisis dan Diskusi

Analisis

- a. Bagaimana performa model dalam mendeteksi warna?

Performa model dalam deteksi warna sangat tergantung pada algoritma yang digunakan dan kualitas data pelatihan. Dalam studi yang menggunakan metode K-NN (K-Nearest Neight) untuk klasifikasi warna mode, model yang sangat akurat dapat diklasifikasikan. Ini menunjukkan bahwa pemilihan algoritma yang sesuai dan data pelatihan representatif sangat penting untuk tugas deteksi warna.

- b. Bagaimana perbedaan akurasi jika jumlah dataset ditambah?

Meningkatkan jumlah dataset biasanya meningkatkan akurasi model *machine learning*. Dengan lebih banyak data, model dapat mempelajari pola yang berbeda dan mengurangi adaptasi yang berlebihan. Misalnya, penelitian telah menunjukkan bahwa menggunakan lebih banyak data pelatihan dapat secara signifikan meningkatkan output model.

- c. Bagaimana cara meningkatkan kinerja model klasifikasi?

Berikut cara yang dapat diterapkan untuk meningkatkan kinerja model klasifikasi:

- a) Pengumpulan data tambahan

Meningkatkan jumlah dan keragaman data pelatihan akan membantu model mengidentifikasi pola yang lebih luas dan meningkatkan generalisasi.

- b) Pengolahan fitur

Membuat dan memilih fitur yang sesuai dapat meningkatkan kemampuan model saat memahami data.

c) Pengaturan *hyperparameter*

Pengaturan parameter model yang tidak dipelajari selama pelatihan dapat mengoptimalkan kinerja.

d) Penggunaan teknik *ensemble*

Beberapa model dapat digabungkan untuk meningkatkan akurasi dengan mengurangi variabilitas dan bias.

e) Validasi dan *cross-validation*

Membagi data menjadi subset untuk pelatihan dan pengujian membantu dalam mengevaluasi kinerja model secara lebih akurat dan mencegah *overfitting*.

Diskusi

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

Keuntungan dari *machine learning* dibandingkan dengan metode yang biasa adalah kemampuannya untuk belajar dari data dan beradaptasi tanpa aturan manual. Berbeda dengan pendekatan *rule-based* berdasarkan aturan eksplisit yang harus ditulis secara individual, *machine learning* dapat secara otomatis mengidentifikasi pola yang kompleks, bahkan ketika sulit untuk dijelaskan dengan aturan tetap. Misalnya, model *machine learning* dapat mengenali hubungan non-linier dalam data dan membuat prediksi yang lebih akurat ketika data pelatihan meningkat. Kemampuan generalisasi ini membuat *machine learning* lebih fleksibel dan efisien dalam lingkungan yang dinamis, memungkinkannya untuk mengelola skenario yang belum pernah terlihat sebelumnya.

2) Bagaimana ML dapat diintegrasikan lebih lanjut dalam sistem kendali?

Integrasi untuk *machine learning* ke dalam sistem kontrol dapat dilakukan dalam berbagai cara untuk meningkatkan kinerja dan kemampuan beradaptasi sistem. Model prediktif berbasis *machine learning* dapat, misalnya, memeriksa perilaku suatu sistem dan secara otomatis mengoptimalkan parameter kontrol yang sangat berguna untuk aplikasi seperti kontrol robot dan sistem otomasi industri. Selain itu, sistem pembelajaran memungkinkan sistem pembelajaran melalui pengalaman langsung, memungkinkan mereka untuk menemukan pedoman kontrol yang optimal untuk situasi kompleks seperti navigasi kendaraan otonom. *Machine learning* juga dapat dikombinasikan dengan teknologi IoT untuk memproses data sensor secara real time, memungkinkan keputusan yang lebih cerdas dan responsif.

3) Apa saja tantangan dalam penerapan ML dalam sistem *real-time*?

Penerapan *machine learning* dalam sistem nyata tidak dapat dipisahkan dari tantangan. Salah satu hambatan terbesar adalah waktu komputer yang terbatas, terutama untuk model kompleks seperti pembelajaran yang mendalam. Ini dapat membutuhkan sumber daya yang hebat dan menyulitkan untuk memenuhi batas waktu yang ketat dalam aplikasi nyata. Selain itu, data yang dikumpulkan secara real time dapat dibatasi atau membatasi latensi, yang dapat mengurangi akurasi dan kecepatan respons sistem. Ada juga tantangan keamanan di mana model *machine learning* rentan terhadap serangan kontroversial yang dapat mempengaruhi keandalan sistem, terutama dalam aplikasi penting seperti kendaraan otonom dan proses industri. Untuk mengatasi tantangan ini, penelitian lebih lanjut tentang optimisasi model menunjukkan bahwa penggunaan komputasi tepi dan teknik inferensi ringan sangat penting untuk mendukung penggunaan ML di lingkungan waktu nyata yang aman dan efisien.

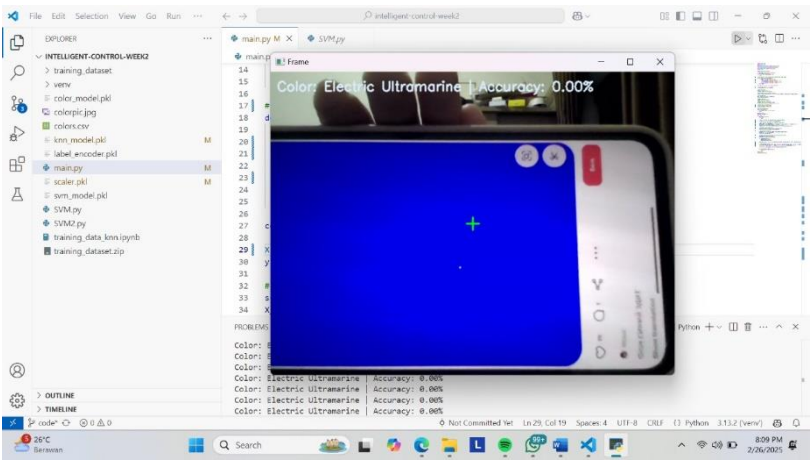
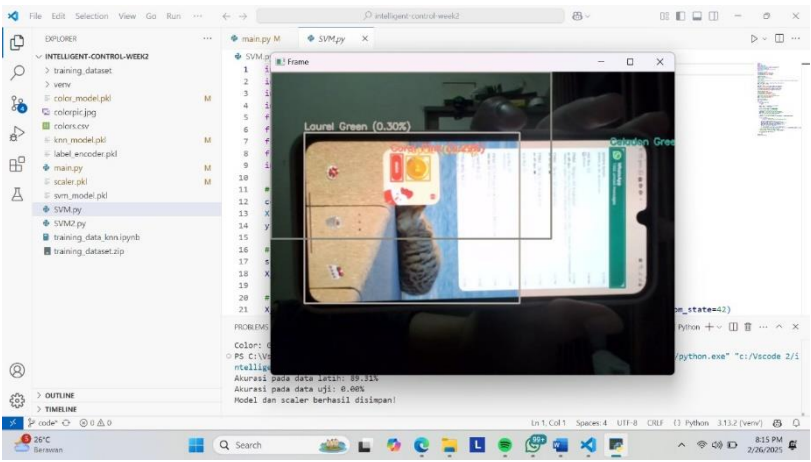
5. Assignment

Program ini mendeteksi warna secara real-time menggunakan kamera, menggunakan *software* OpenCV untuk menangkap gambar dan mengubahnya ke format HSV agar lebih mudah mengenali warna. Model SVM (*Support Vector Machine*) yang telah dilatih sebelumnya mengklasifikasikan warna berdasarkan fitur RGB, sementara referensi warna disimpan dalam file `colors.csv`. Deteksi objek berwarna dilakukan dengan mencari rentang warna tertentu (misalnya merah, hijau, biru, dan kuning), lalu menggambar *bounding box* dan menampilkan nama warna beserta akurasinya.

Untuk meningkatkan akurasi, fitur RGB dinormalisasi dengan `StandardScaler`. Namun, program masih terbatas pada warna-warna utama dan rentan terhadap fluktuasi pencahayaan. Ini dapat diatasi dengan teknik seperti *adaptive histogram equalization* atau augmentasi data. Selain itu, jika dataset membesar, performa SVM bisa menurun, sehingga model lain seperti *Random Forest* atau CNN mungkin lebih optimal. Meski begitu, dengan penyempurnaan pada rentang warna dan penyesuaian filter, program ini memiliki potensi besar untuk berbagai aplikasi pengenalan warna dalam kondisi pencahayaan yang beragam.

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	Model KNN	
2	Model SVM	

7. Kesimpulan

Dari percobaan yang telah dilakukan, dapat ditarik kesimpulan bahwa:

1. Program mendeteksi warna utama secara *real-time* dengan OpenCV dan SVM untuk mengukur akurasi.
2. Performa program dapat dipengaruhi oleh pencahayaan dan ukuran dataset, tetapi bisa diatasi dengan teknik seperti augmentasi data atau *histogram equalization*.
3. Program dapat dikembangkan lebih lanjut dengan optimasi rentang warna, filter tambahan, atau model yang lebih kompleks seperti Random Forest atau CNN.

8. Saran

Pada percobaan selanjutnya, penggunaan model yang lebih kompleks, seperti Convolutional Neural Networks (CNN), serta penambahan dataset yang lebih beragam dapat diterapkan agar data yang dihasilkan menjadi lebih variatif dan informatif

10. Daftar Pustaka

- Gifari, J., & Widya, A. (2020, Desember). Belajar Machine Learning Dengan Library Python: Scikit-Learn. *DqLab AI-Powered Learning*. <https://dqlab.id/belajar-machine-learning-dengan-library-python-scikit-learn>
- Mulyawan, R. (2025, February 18). *Computer Vision: Pengertian, Apa itu Visi Komputer? Tujuan, Fungsi, Jenis, Contoh dan Pentingnya + Bedanya dengan Image Processing!* <https://rifqimulyawan.com/blog/computer-vision/>
- Nugroho, I. (2025, January 23). *Computer Vision Adalah: Definisi, Fungsi, dan Contoh*. <https://dibimbing.id/blog/detail/computer-vision-adalah>
- Ramadhani, A., & Sembiring, M. (2022). SISTEM KENDALI BERBASIS MACHINE LEARNING MENGGUNAKAN MODEL NEIVE BAYES PADA PENGERINGAN PADI OTOMATIS. *Journal of Science and Social Research*.