

Laporan Praktikum Kontrol Cerdas

Nama : Vivi Aulia Husna Wahidah
NIM : 224308023
Kelas : TKA-6A
Akun Github (Tautan) : <https://github.com/vivi-aulia>
Student Lab Assistant : Muhammad Mahirul Faiq

1. Judul Percobaan

Week 1: Colors Detection with OpenCV

2. Tujuan Percobaan

Tujuan dari praktikum “Colors detection with OpenCV”, mahasiswa diharapkan mampu:

- Memahami prinsip dasar sistem kontrol cerdas (intelligent control systems).
- Mengetahui peran Artificial Intelligence (AI), Machine Learning (ML), dan Deep Learning (DL) dalam penerapan sistem kontrol.
- Mempelajari implementasi Computer Vision dalam sistem kontrol berbasis AI.
- Menggunakan Python dan OpenCV untuk mengenali serta mendeteksi beberapa warna.

3. Landasan Teori

1. Sistem Kontrol Cerdas

Sistem kontrol cerdas menggabungkan teknik kecerdasan buatan (Artificial Intelligence, AI) dengan teori kontrol klasik untuk mengendalikan sistem yang kompleks dan dinamis. Pendekatan ini memungkinkan sistem untuk belajar dari pengalaman, beradaptasi terhadap perubahan lingkungan, dan membuat keputusan secara mandiri.

2. Kecerdasan Buatan (*Artificial Intelligence, AI*)

AI adalah cabang ilmu komputer yang berfokus pada pengembangan sistem yang mampu melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia, seperti pengenalan pola, pemahaman bahasa alami, dan pengambilan keputusan. AI mencakup berbagai teknik dan metode, termasuk pembelajaran mesin (Machine Learning, ML) dan pembelajaran mendalam (Deep Learning, DL), yang memungkinkan sistem untuk belajar dari data dan meningkatkan kinerjanya seiring waktu.

3. Pembelajaran Mesin (*Machine Learning, ML*) dan Pembelajaran Mendalam (*Deep Learning, DL*)

Machine Learning adalah subbidang dari AI yang berfokus pada pengembangan algoritma yang memungkinkan komputer untuk belajar dari data dan membuat prediksi atau keputusan tanpa diprogram secara eksplisit. Deep Learning, sebagai subbidang dari Machine Learning, menggunakan jaringan saraf tiruan dengan banyak lapisan (deep neural networks) untuk memodelkan representasi data yang kompleks. DL telah mencapai kemajuan signifikan dalam berbagai aplikasi, seperti pengenalan suara, visi komputer, dan pemahaman bahasa alami.

4. Visi Komputer (*Computer Vision*)

Visi komputer adalah bidang dalam AI yang memungkinkan komputer untuk memahami dan menafsirkan dunia visual, seperti gambar dan video. Teknik ini digunakan dalam berbagai

aplikasi, termasuk pengenalan wajah, pelacakan objek, dan deteksi warna. Dalam sistem kontrol berbasis AI, visi komputer berperan dalam mengolah data visual untuk menghasilkan keputusan yang lebih cerdas.

5. Python dan OpenCV

Python adalah bahasa pemrograman tingkat tinggi yang dikenal karena sintaksnya yang sederhana dan kemudahan penggunaannya. Python telah menjadi pilihan utama dalam pengembangan aplikasi AI dan ML karena ekosistem pustaka dan kerangka kerja yang luas. OpenCV (*Open Source Computer Vision Library*) adalah pustaka sumber terbuka yang dirancang untuk aplikasi visi komputer dan pengolahan citra. OpenCV menyediakan berbagai fungsi dan algoritma untuk tugas-tugas seperti deteksi objek, pengenalan wajah, dan pelacakan gerakan.

6. Prinsip Deteksi Warna

Deteksi warna dalam visi komputer dilakukan dengan memanfaatkan representasi warna dalam berbagai model warna, seperti RGB (*Red, Green, Blue*) dan HSV (*Hue, Saturation, Value*). Model RGB adalah model warna yang paling umum digunakan dalam tampilan digital, sementara model HSV sering digunakan dalam pemrosesan gambar karena lebih sesuai dengan persepsi manusia terhadap warna. Dalam OpenCV, konversi warna dari RGB ke HSV sering digunakan untuk meningkatkan akurasi deteksi warna karena HSV lebih tahan terhadap perubahan pencahayaan dibandingkan dengan RGB.

7. Algoritma Deteksi Warna dengan OpenCV

Dalam eksperimen deteksi warna, OpenCV menyediakan beberapa langkah utama:

1. Membaca gambar atau video menggunakan OpenCV.
2. Mengonversi gambar dari format RGB ke HSV.
3. Menentukan rentang warna yang akan dideteksi.
4. Menggunakan operasi thresholding untuk mengekstrak warna yang diinginkan.
5. Menampilkan hasil deteksi warna dalam bentuk gambar atau video yang telah diproses.

Dengan pendekatan ini, sistem dapat mengenali warna tertentu dalam suatu objek dan menggunakannya untuk berbagai aplikasi, seperti pelacakan objek dan pengendalian robot berbasis warna.

4. Analisis dan Diskusi

Analisis Hasil:

- Apa yang terjadi saat objek berwarna merah muncul di kamera? Ketika terdapat objek berwarna merah muncul di kamera akan terdeteksi oleh sistem sehingga akan terbentuk bounding box yang mendeteksi objek berwarna merah
- Bagaimana sistem mendeteksi dan memfilter warna merah? Dalam sistem ini pendeteksian warna menggunakan teknik pemrosesan citra dengan OpenCV. Berikut adalah langkah-langkah yang dilakukan:
 1. Gambar dari kamera dikonversi dari ruang warna BGR (*Blue, Green, Red*) ke ruang warna HSV (*Hue, Saturation, Value*).
 2. Definisi rentang warna mencakup nilai-nilai *Hue, Saturation, dan Value* yang sesuai dengan warna. Contoh HSV warna merah:
`lower_red = np.array([0, 120, 70])`
`upper_red = np.array([10, 255, 255])`
 3. Deteksi kontur (batas objek) dalam gambar biner ditemukan menggunakan fungsi `cv2.findContours`.
 4. Bounding box digambar di sekitar kontur yang terdeteksi. Hanya kontur dengan area yang lebih besar dari nilai tertentu yang akan ditampilkan untuk menghindari deteksi noise atau objek kecil.
- Bagaimana metode ini dapat diterapkan dalam *intelligent control systems*? Metode deteksi warna dan bounding box ini dapat diterapkan dalam sistem kontrol cerdas seperti pada sistem pengawasan, robotika, dan kendaraan otonom.

Diskusi:

- Bagaimana AI dapat meningkatkan sistem kontrol berbasis Computer Vision? Penggunaan AI dapat meningkatkan akurasi deteksi dengan dilatih dengan akurasi yang lebih baik. AI dapat terus belajar dan beradaptasi sehingga dapat memperbaiki kinerja dan mengurangi kesalahan.
- Apa kelebihan dan kekurangan metode deteksi objek berbasis warna? Kelebihan metode ini relatif sederhana dan cepat untuk diimplementasikan, tidak memerlukan komputasi yang berat. Aplikasi penggunaan yang luas seperti robotika dan pengawasan. Kekurangan metode ini sangat sensitif terhadap perubahan kondisi pencahayaan. Dapat mendeteksi noise atau objek kecil yang tidak relevan jika tidak ada filter tambahan untuk menghilangkan deteksi yang tidak diinginkan.
- Bagaimana cara meningkatkan akurasi sistem deteksi objek? Untuk meningkatkan akurasi sistem deteksi objek, dapat dilakukan dengan beberapa hal seperti kalibrasi kamera, peningkatan resolusi gambar, dan penyesuaian rentang warna.

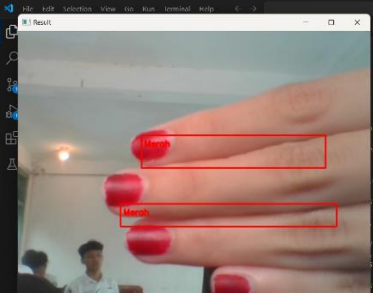
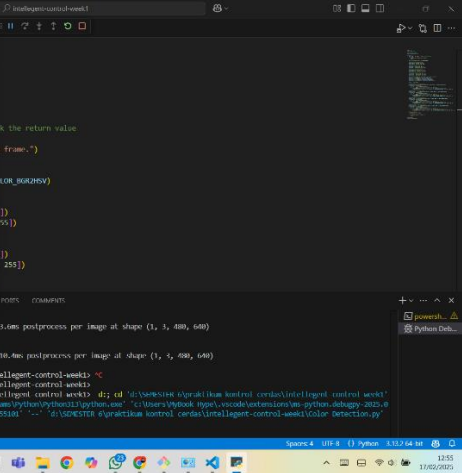
5. Assignment

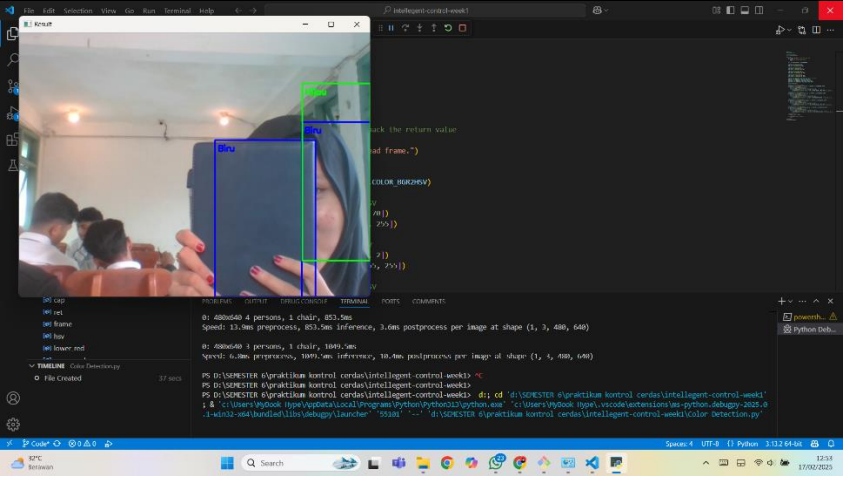
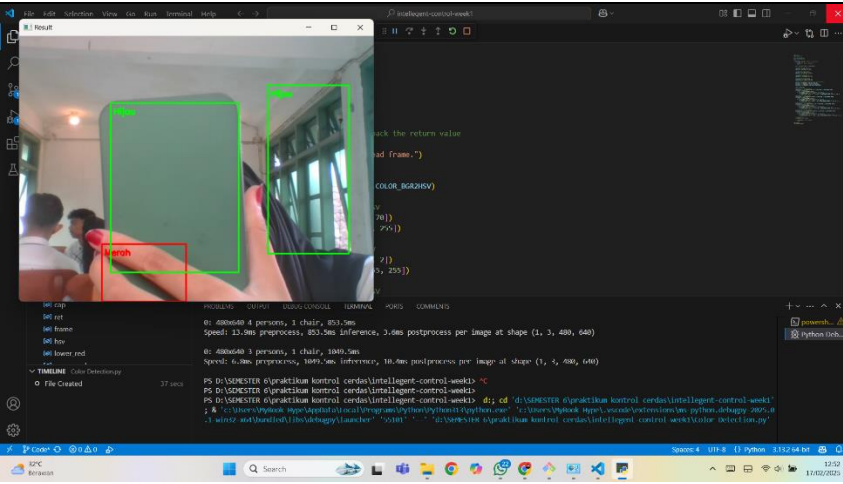
Dalam pengerjaan assignment dilakukan beberapa hal antara lain,

1. Melakukan studi literatur mengenai Intelligent control systems, Artificial intelligence (AI), Machine Learning (ML), Deep Learning (DL), Python, dan OpenCV.
2. Membuat akun github
3. Mengakukan instalasi software, VSCode, Phyton, dan GIT.
4. Membuat repository dan clone kedalam laptop.
5. Membuat kode program dan menjalankan program untuk menampilkan hasil deteksi warna.
6. Melakukan Commit dan Push Kode ke GitHub.

6. Data dan Output Hasil Pengamatan

Data dan hasil yang diperoleh selama percobaan.

No	Variabel	Hasil Pengamatan
1	Pengujian warna merah	 

2	Pengujian warna biru	
3	Pengujian warna hijau	

7. Kesimpulan

Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa sistem kontrol cerdas yang menggabungkan kecerdasan buatan (*Artificial Intelligence*, AI) dan visi komputer (*Computer Vision*) dapat meningkatkan efektivitas dalam deteksi warna menggunakan OpenCV. Dengan menerapkan model warna HSV, sistem deteksi warna menjadi lebih akurat dibandingkan dengan model RGB, terutama dalam kondisi pencahayaan yang bervariasi.

Selain itu, penggunaan algoritma pemrosesan citra dengan OpenCV memungkinkan proses deteksi warna dilakukan secara real-time dan efisien. Langkah-langkah seperti konversi warna, penentuan rentang warna, dan operasi *thresholding* terbukti mampu mengekstrak warna target dengan baik, sehingga sistem ini dapat diterapkan dalam berbagai bidang, seperti pelacakan objek, kontrol robot berbasis warna, serta sistem otomatisasi berbasis pengenalan visual.

Dengan demikian, praktikum ini memberikan pemahaman yang lebih mendalam tentang bagaimana AI, ML, dan computer vision dapat diterapkan dalam sistem kontrol cerdas, khususnya dalam deteksi warna menggunakan OpenCV dan Python. Dengan menggunakan metode ini memiliki potensi besar dalam pengembangan sistem pengawasan, robotika, dan kendaraan otonom, terutama dalam identifikasi dan pelacakan objek berdasarkan warna.

8. Saran

- Untuk meningkatkan akurasi deteksi warna, perlu dilakukan kalibrasi lebih lanjut terhadap rentang HSV agar sesuai dengan kondisi pencahayaan yang berbeda.
- Percobaan selanjutnya dapat melibatkan deteksi warna dalam lingkungan yang lebih kompleks dengan berbagai sumber cahaya untuk menguji ketahanan algoritma terhadap perubahan lingkungan.

9. Daftar Pustaka

- 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. <https://ejournal.nusantaraglobal.or.id/index.php/sentri/article/view/3185>
- Oktivisari, P., & Ardiansyah, E. (2017). Image Processing Warna untuk Proses Interupsi LED pada Lampu Lalu Lintas Berbasis OpenCV. *Jurnal Komunika: Jurnal Komunikasi, Media dan Informatika*, 6(2). <https://jurnal.kominfo.go.id/index.php/komunika/article/view/1172>
- Dini, A. J. (2019). Sistem Deteksi Warna Daun Padi Menggunakan Metode Irisan dan Korelasi di dalam OpenCV. *Universitas Muhammadiyah Yogyakarta*. <http://repository.umy.ac.id/handle/123456789/30142>