

Laporan Praktikum Kontrol Cerdas

Nama : Gunawan Wibisono
NIM : 224308083
Kelas : TKA – 7D
Akun *Github* (Tautan) : <https://github.com/gunawanwibisono83>
Student Lab Assistant : -

1. Judul Percobaan

Penerapan *Computer Vision* dalam mendeteksi warna dengan OpenCV & Python

2. Tujuan Percobaan

Tujuan dilakukannya percobaan ini yaitu :

- a. Mahasiswa dapat memahami konsep dasar dari kontrol cerdas.
- b. Mahasiswa dapat mengaplikasikan *Computer Vision* menggunakan Python dan OpenCV dalam mendeteksi warna suatu objek.
- c. Mahasiswa dapat mengetahui cara kerja deteksi warna dengan OpenCV.

3. Landasan Teori

Kontrol cerdas adalah cabang penelitian & simulasi aktivitas, serta pengembangan sistem kontrol dan pemrosesan informasi dengan kecerdasan serupa manusia (Faith, 2022). Berbeda dengan kontrol tradisional yang bergantung pada model matematis serta kurangnya fleksibilitas, kontrol cerdas dengan teknologi AI mampu menyesuaikan diri dengan perubahan lingkungan dan kondisi sistem melalui proses pembelajaran. Kecerdasan buatan atau *artificial intelligence* (AI) pada dasarnya merupakan sebuah “mesin” yang berkemampuan dalam melakukan berbagai hal yang dianggap memerlukan kecerdasan saat manusia mengoperasikannya (Disemadi, 2021). Salah satu lingkup utama AI yaitu *Computer Vision*.

Computer Vision adalah proses pembelajaran dan analisa gambar atau video untuk memperoleh hasil sebagaimana yang bisa dilakukan manusia. Atau lebih mudahnya *Computer Vision* mencoba meniru cara kerja visualisasi manusia (Dompeipen dkk., 2021). *Computer Vision* mempunyai beberapa macam bidang salah satunya adalah *object detection*. *Object detection* bekerja dengan mengenali objek yang berada pada gambar dan letak objek tersebut. Salah satu cara untuk mengenali objek dalam suatu gambar yaitu dengan mendeteksi warna dari objek tersebut. Selama ini banyak sistem yang masih bergantung pada pendekatan konvensional berbasis histogram yang rentan terhadap *noise*. Oleh karena itu, sistem deteksi warna yang dipraktikkan dalam percobaan kali ini menggunakan bahasa pemrograman Python dan *library* OpenCV guna mengembangkan sistem yang lebih efisien (Ardianto, 2024).

Python telah menjadi bahasa pemrograman yang populer dalam pengembangan perangkat lunak dan keberadaan *library* OpenCV yang berisi fungsi-fungsi pemrograman seperti konversi ruang warna untuk teknologi *Computer Vision* secara *real time*. Konversi ruang warna dilakukan pada ruang warna RGB (*Red, Green, Blue*) bawaan kamera *device* ke ruang warna HSV (*Hue, Saturation, Value*). Ruang warna HSV dipilih karena lebih sesuai dengan cara manusia mengenali warna. *Hue* mewakili jenis warna, *Saturation* menunjukkan tingkat kemurnian suatu warna, dan *Value* menyatakan tingkat kecerahan. Dengan menentukan rentang nilai HSV tertentu, sistem dapat memfilter objek sesuai warna yang diinginkan.

4. Analisis dan Diskusi

a. Analisis

Pada percobaan pertama ini, saya *running* kode program Python untuk mendeteksi warna merah dengan OpenCV. Hasilnya muncul 3 tampilan jendela dengan label *Frame*, *Mask*, dan *Result* dengan hasil yang berbeda. Ketika objek berwarna merah diarahkan ke kamera *device*, pada tampilan *Frame* objek terlihat dalam warna asli (natural) sebagaimana ditangkap oleh kamera. Lalu pada tampilan *Mask*, area yang sesuai dengan rentang HSV (*Hue, Saturation, Value*) merah berubah menjadi putih, sedangkan area lain

menjadi hitam, sehingga objek merah tampak terisolasi dari latar belakang. Sementara itu, tampilan *Result* menunjukkan kombinasi antara gambar asli dengan hasil *mask*, di mana objek merah tampak lebih menonjol dibanding area sekitarnya.

Sistem mendeteksi warna merah dengan mengkonversi warna dari kamera yang semula dalam format RGB (*Red, Green, Blue*) menjadi format HSV (*Hue, Saturation, Value*). *Hue* menunjukkan jenis warna, *Saturation* menggambarkan seberapa pekat warna, sedangkan *Value* menunjukkan tingkat kecerahan. Setelah diubah menjadi HSV, kemudian ditentukan rentang nilai yang mewakili warna merah. Setiap rentang nilai piksel dalam gambar yang diambil kemudian dibandingkan. Piksel yang sesuai akan ditandai dalam sebuah *mask*, sedangkan piksel lain akan ditandai hitam. *Mask* ini selanjutnya digunakan untuk menyoroti bagian berwarna merah, sehingga objek merah dapat terlihat lebih jelas.

Contoh implementasi metode deteksi warna berbasis *Computer Vision* ini dapat diterapkan untuk proses penyortiran barang berdasarkan warna atau mendeteksi kualitas produk. Meskipun sederhana, metode ini akan memiliki manfaat berlebih ketika digabungkan dengan algoritma AI yang lebih kompleks.

b. Diskusi

Penerapan AI untuk meningkatkan sistem kontrol berbasis *Computer Vision* dapat membuat sistem mendeteksi objek berdasarkan warna secara lebih adaptif dan cepat dalam komputasi.

Kelebihan dari metode ini yaitu relatif sederhana, serta tidak membutuhkan komputasi yang rumit sehingga dapat dijalankan hampir di semua perangkat dengan cepat. Namun, kekurangannya yaitu hasil deteksi sangat dipengaruhi oleh kondisi cahaya dimana perubahan intensitas lampu atau bayangan bisa membuat warna tampak berbeda sehingga sistem salah mengenali objek.

Cara untuk meningkatkan akurasi dapat dilakukan dengan menggunakan kamera yang memiliki pencahayaan yang stabil.

5. Assignment

Berdasarkan percobaan awal yang dilakukan untuk mendeteksi objek berwarna merah, program kemudian dikembangkan agar mampu mendeteksi warna biru dan hijau. Dengan menambah fungsi `area = cv2.contourArea(cnt)`; **if area > 2000** modifikasi dilakukan dengan memberi aturan baru yaitu hanya objek besar yang dapat terdeteksi sehingga *noise* kecil tidak terdeteksi. Selain itu, ditambahkan pula fungsi `cv2.boundingRect(cnt)`; `cv2.rectangle`; `cv2.putText` untuk memunculkan *bounding box* berbentuk persegi panjang serta teks untuk memberikan penanda pada objek yang sesuai dengan warna yang telah ditentukan. Untuk menyederhanakan sistem, tampilan yang semula terdiri dari tiga jendela diubah menjadi satu jendela saja.

6. Data dan Output Hasil Pengamatan

No.	Variabel	Hasil Pengamatan
1.	Deteksi Warna Merah	
2.	Deteksi Warna Hijau	
3.	Deteksi Warna Biru	

7. Kesimpulan

Berdasarkan percobaan yang telah dilakukan, maka dapat diambil kesimpulan bahwa sistem berhasil mendeteksi objek dengan warna tertentu (merah, hijau, dan biru) menggunakan metode konversi ruang warna. Program yang dimodifikasi dapat menampilkan *bounding box* serta label pada objek sesuai warnanya, sehingga objek dapat diidentifikasi secara jelas. Penerapan AI dalam meningkatkan sistem kontrol berbasis *Computer Vision* untuk mendeteksi objek berdasarkan warna terbukti lebih adaptif. Namun, akurasi deteksi masih dipengaruhi oleh intensitas cahaya.

8. Saran

Bisa dilakukan modifikasi lebih lanjut terhadap program sistem deteksi warna ini serta penggunaan kamera yang memiliki intensitas cahaya yang lebih baik guna meningkatkan akurasi deteksi.

9. Daftar Pustaka

- Ardianto, D. (2024). Color Detector in an Image using Python and Computer Vision Library. *Journal of Intelligent Systems and Information Technology*, 1(1), 25–30.
- Disemadi, H. (2021). Urgensi Regulasi Khusus dan Pemanfaatan Artificial Intelligence dalam Mewujudkan Perlindungan Data Pribadi di Indonesia. *Jurnal Wawasan Yuridika*, 5(2), 177–199. <https://doi.org/10.25072/jwy.v5i2.460>
- Dompeipen, T., Sompie, S., & Najoan, M. (2021). Computer Vision Implementation for Detection and Counting the Number of Humans. *Jurnal Teknik Informatika*, 16(1), 65–76.
- Faith, F. (2022, September 14). *What is the Use of Intelligent Control System? Application of Intelligent Control System*. <https://www.fourfaith.com/industry-news/intelligent-control-system.html>