

# Laporan Praktikum Kontrol Cerdas

Nama : Widhi Widya Hastuti

NIM : 224308025

Kelas : TKA-6A

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

Student Lab Assistant : Yulia Brilianty

## 1. Judul Percobaan

Sistem Deteksi Objek Real-Time dengan OpenCV

## 2. Tujuan Percobaan

- Memahami prinsip dasar dari Sistem Kontrol Cerdas.
- Memahami penerapan Kecerdasan Buatan (AI) dalam sistem kendali.
- Menggunakan Python dan OpenCV untuk melakukan deteksi objek secara dasar.
- Memahami cara penggunaan OpenCV untuk analisis citra.
- Menampilkan hasil deteksi objek secara langsung melalui kamera dalam waktu nyata.

## 3. Landasan Teori

Kecerdasan Buatan (AI) merupakan salah satu cabang ilmu komputer yang berfokus pada pengembangan sistem yang dapat meniru kecerdasan manusia dalam melakukan tugas-tugas tertentu. Dalam konteks sistem kendali, AI digunakan untuk pengambilan keputusan dan mengontrol perangkat atau sistem tanpa memerlukan manusia secara langsung. Salah satu penerapan utama AI dalam sistem kendali adalah melalui algoritma pembelajaran mesin (machine learning) dan pembelajaran mendalam (deep learning), yang memungkinkan sistem untuk belajar dari data dan beradaptasi dengan kondisi yang berubah. AI membantu meningkatkan efisiensi dan akurasi dalam pengambilan keputusan di mana keputusan pengendalian harus cepat, tepat, dan efisien.

Python dan OpenCV (Open Source Computer Vision Library), merupakan kombinasi yang sangat populer dalam dunia pengolahan citra dan visi komputer. OpenCV menyediakan berbagai alat dan algoritma untuk memproses citra, serta mendeteksi objek secara otomatis dalam gambar atau video. Dalam deteksi objek dasar, OpenCV menggunakan teknik-teknik seperti konversi gambar ke grayscale, thresholding, dan deteksi tepi untuk mengenali objek

tertentu. Salah satu metode yang sering digunakan dalam deteksi objek adalah algoritma Haar Cascade, yang efektif dalam mengenali pola objek dalam citra. Dengan menggunakan Python, sistem deteksi objek dapat dibangun secara efisien dan fleksibel, memungkinkan aplikasi deteksi objek baik untuk citra statis maupun video real-time.

OpenCV tidak hanya digunakan untuk deteksi objek, tetapi juga untuk berbagai macam analisis citra yang lebih mendalam. Dalam analisis citra, OpenCV menyediakan alat untuk memproses gambar, mengekstraksi fitur, dan menganalisis informasi visual yang ada. Selain itu, OpenCV juga memungkinkan pengolahan citra secara real-time, yang penting untuk aplikasi-aplikasi seperti pengawasan video, pelacakan objek, dan kendaraan otonom. Dengan memanfaatkan OpenCV, analisis citra menjadi lebih efektif, memungkinkan ekstraksi informasi yang lebih baik dan meningkatkan kemampuan sistem dalam mendeteksi dan memahami dunia di sekitar mereka.

## **4. Analisis dan Diskusi**

### **Analisis**

Pada saat program dijalankan, akan menampilkan 3 hasil yaitu frame, mask, dan result. Jika pada frame mendeteksi warna merah maka pada jendela result akan menampilkan warna merah, sementara untuk jendela mask menampilkan warna hitam-putih. Jika tidak ada objek yang berwarna merah, maka jendela Result akan tampak gelap atau tidak menunjukkan objek tertentu.

Sistem mendeteksi dan memfilter warna merah dengan mengubah citra ke format HSV, kemudian mengatur rentang nilai Hue untuk warna merah. Setelah itu, thresholding diterapkan menggunakan fungsi `cv2.inRange()` untuk menghasilkan mask yang memisahkan objek merah dari latar belakang. Mask ini kemudian digunakan untuk menggabungkan citra asli, sehingga hanya objek merah yang terdeteksi yang muncul dalam gambar hasil. Dengan cara ini, sistem dapat secara efektif mendeteksi objek berwarna merah meskipun dalam kondisi pencahayaan yang bervariasi.

Metode deteksi objek berbasis warna menggunakan OpenCV, seperti yang diterapkan dalam kode ini, dapat berperan penting dalam berbagai aplikasi sistem control cerdas. Dengan memanfaatkan deteksi warna merah, sistem dapat mengenali objek, mengarahkan gerakan, memantau lingkungan secara real-time, dan merespons secara otomatis. Penerapan teknologi ini dalam robotika, sistem pemantauan, dan pengawasan, akan memungkinkan sistem untuk bekerja lebih cerdas, efisien terhadap kondisi yang ada.

## Diskusi

AI dapat meningkatkan sistem kontrol berbasis computer vision dengan memanfaatkan deep learning untuk mengenali objek yang lebih bervariasi, belajar dari data, dan menyesuaikan deteksi berdasarkan perubahan lingkungan secara real-time. Dengan kemampuan tersebut, AI dapat meningkatkan akurasi dan efisiensi dalam aplikasi seperti robotika, dan pemantauan otomatis, sehingga sistem kontrol dapat lebih responsif dan cerdas dalam menghadapi situasi yang dinamis.

Kelebihan utama dari metode ini adalah kemudahannya dalam implementasi dan kecepatan pemrosesan, karena hanya memerlukan konversi warna dan thresholding untuk memisahkan objek dari latar belakang.

Kekurangan metode ini adalah ketergantungan pada kondisi pencahayaan yang stabil dan keterbatasan dalam mendeteksi objek dengan warna yang serupa dengan latar belakang. Selain itu, metode ini kurang efektif untuk objek yang memiliki variasi warna atau untuk mendeteksi objek dalam kondisi yang kompleks, seperti perubahan dalam cahaya atau bayangan.

Meningkatkan akurasi deteksi objek dapat dilakukan dengan mengoptimalkan kualitas data, memilih algoritma tepat, dan meningkatkan efisiensi pemrosesan. Untuk deteksi yang lebih akurat, metode berbasis fitur seperti HOG dan SIFT atau model *deep learning* seperti YOLO, Faster R-CNN, dan SSD dapat digunakan, terutama dalam lingkungan yang kompleks.

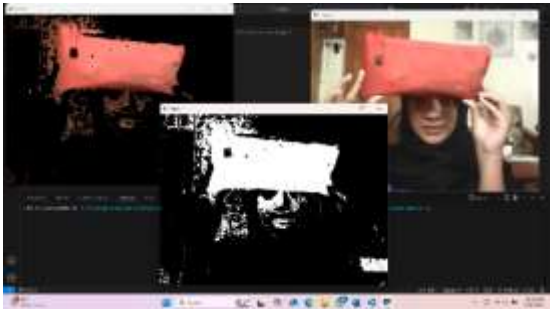
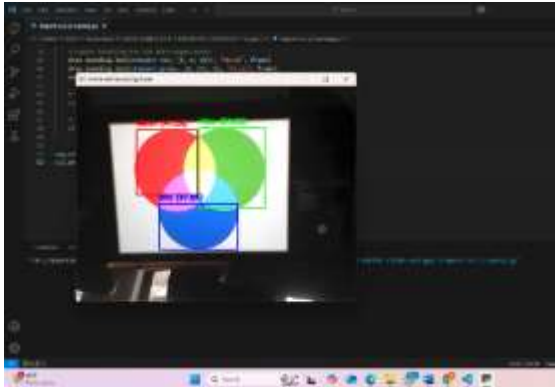
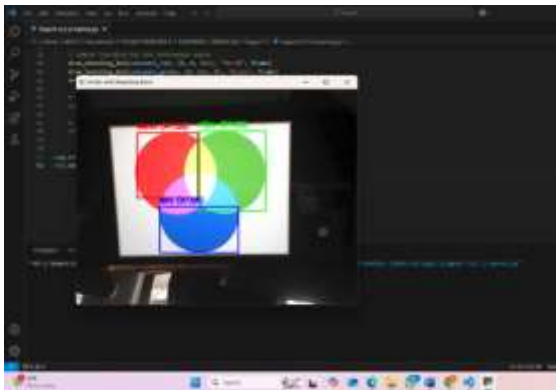
## 5. Assignment

Dalam percobaan ini, ketika program dijalankan, sistem akan menampilkan tiga frame utama, yaitu frame utama, frame mask, dan frame hasil (result). Jika objek berwarna merah terdeteksi oleh kamera, maka hanya warna merah yang akan ditampilkan pada frame hasil, sementara warna lainnya akan diubah menjadi hitam. Di sisi lain, pada frame mask, area yang terdeteksi sebagai warna merah akan ditampilkan dalam warna putih, sedangkan area lainnya tetap berwarna hitam. Sistem ini memanfaatkan model ruang warna HSV (Hue, Saturation, Value) untuk mendeteksi dan memfilter warna merah. Ketika gambar ditangkap oleh kamera dan dikonversi ke format HSV, program akan mencari warna merah berdasarkan rentang nilai Hue, Saturation, dan Value yang telah ditentukan. Kemudian, sistem akan membuat masker biner yang menyoroti area berwarna merah, lalu menggabungkannya dengan citra asli, sehingga hanya bagian yang memiliki warna merah yang tetap terlihat. Penggunaan metode deteksi warna merah dengan HSV Color Filtering memiliki banyak manfaat dalam sistem kendali cerdas (Intelligent Control Systems). Salah satu aplikasinya adalah dalam bidang navigasi dan

kontrol robotik, di mana deteksi warna merah dapat digunakan sebagai penanda (marker) untuk membantu robot dalam navigasi otomatis.

## 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	Mendeteksi warna merah	
2	Mendeteksi warna biru	
3	Mendeteksi warna hijau	

## 7. Kesimpulan

Berdasarkan hasil praktikum dan analisis yang dilakukan, dapat disimpulkan bahwa:

- Sistem ini mampu mendeteksi objek berwarna biru secara real-time dengan memanfaatkan OpenCV.
- Penggunaan fitur bounding box sangat membantu dalam meningkatkan akurasi deteksi dengan memberi penanda pada objek yang terdeteksi, sehingga mempermudah identifikasi dan verifikasi objek.
- Metode ini efektif dalam mendeteksi objek berdasarkan warna tertentu, meskipun terdapat keterbatasan seperti potensi kesalahan deteksi akibat perubahan pencahayaan atau adanya objek lain dengan warna yang serupa.

## 8. Saran

Saran untuk pengembangan sistem deteksi objek ini adalah dengan mengintegrasikan teknik deteksi yang lebih maju, seperti machine learning, untuk membedakan objek berdasarkan bentuk dan warna, serta melakukan pengujian pada bagian kondisi pencahayaan. Selain itu, penggunaan kamera dengan kualitas lebih tinggi dapat meningkatkan kualitas gambar, sementara penerapan sistem umpan balik dapat membantu memperbaiki algoritma dengan memperhatikan kesalahan deteksi yang terjadi

## 9. Daftar Pustaka

Rosebrock, A. (2016). *PyImageSearch Gurus: Computer Vision and Deep Learning with Python*. PyImageSearch.

Raghu, V. & Tiwari, P. (2017). "Real-Time Object Detection Using OpenCV." *International Journal of Computer Applications*, 159(5), 15-19.

Gómez, J., & Pérez, J. (2019). *Real-Time Object Detection and Tracking using OpenCV*. *International Journal of Computer Vision and Image Processing*, 9(2), 35-50.