

# Laporan Praktikum Kontrol Cerdas

Nama : Angger Tri Sudrajad  
NIM : 224308028  
Kelas : TKA-6B  
Akun Github : <https://github.com/Angger1>  
Student Lab Assistant : Mba Salma Halizah

1. Judul Percobaan: **Deteksi objek Sederhana Berwarna Dengan Open CV**
2. Tujuan Percobaan:
  - Memahami konsep dasar kontrol cerdas (intelligent control systems).
  - Mengenali peran AI, *Machine Learning* (ML), dan *Deep Learning* (DL) dalam sistem kendali.
  - Mempelajari penerapan *Computer Vision* dalam sistem kontrol berbasis AI.
  - Menggunakan Python dan OpenCV untuk mendeteksi objek secara sederhana.
  - Memanfaatkan GitHub untuk version control dan Kaggle sebagai sumber dataset.

3. Landasan Teori:

Kecerdasan buatan atau *Artificial Intelligence* (AI) adalah sistem yang mempelajari bagaimana membuat komputer dapat berpikir, belajar, dan bertindak seperti manusia (Dosari & Abouellail, 2023). *Intelligent Control* adalah metode pengendalian sistem yang menggunakan AI, *Machine Learning*, dan *Deep Learning* untuk meningkatkan performa dan efisiensi. Contoh implementasi *Intelligent Control* yaitu pada robotika (kontrol gerak robot berbasis AI), otomotif (*autonomous driving system*), industri (prediksi dan optimasi proses manufaktur), dan medis (AI untuk kontrol peralatan kesehatan) (Marsella dkk., 2023).

*Software* yang digunakan dalam percobaan ini berupa Python dan Open CV. OpenCV (*Open Source Computer Vision Library*) adalah perpustakaan perangkat lunak yang dirancang khusus untuk pengolahan citra dan visi komputer. Dikembangkan oleh Intel dan sekarang bersifat open-source,

OpenCV menyediakan berbagai alat dan fungsi yang memudahkan pengembang untuk membuat aplikasi berbasis pengolahan citra dan visi komputer (Zebua & Rosyani, 2024). Python, sebagai salah satu bahasa pemrograman yang didukung oleh OpenCV, menjadi pilihan populer karena sintaksnya yang mudah dipahami dan kemampuannya untuk integrasi dengan berbagai pustaka lain, seperti NumPy, SciPy, dan scikit-learn. Dengan OpenCV Python, pengembang dapat mengakses berbagai algoritma canggih seperti deteksi objek, pelacakan, pengenalan wajah, dan segmentasi citra. Salah satu aplikasi penting dari OpenCV Python adalah dalam deteksi objek (Alam dkk., 2024). Deteksi objek adalah proses mengidentifikasi dan menemukan instansi objek tertentu dalam gambar atau video. Dalam konteks pengolahan citra dan visi komputer, deteksi objek melibatkan penggunaan algoritma dan teknik khusus untuk mengenali berbagai jenis objek.

Model warna RGB merupakan singkatan dari Red (Merah) Green (Hijau) Blue(Biru). Model warna RGB juga disebut additive color atau warna pencahayaan karena apabila RGB dikombinasikan maka akan menghasilkan warna putih. RGB merupakan model warna yang paling dasar dalam melakukan penyimpanan gambar (Goenawan dkk., 2022). Pada setiap pixel warna memiliki rentang nilai intensitas mulai dari 0 sampai dengan 255. Setiap titik yang berada pada ruang warna RGB merupakan warna dengan memiliki komponen R, G dan B. Untuk titik (0,0,0) merupakan titik warna yang berwarna hitam, sedangkan titik (1,1,1) merupakan titik warna yang berwarna putih

#### 4. Analisis dan Diskusi:

- Analisis

Percobaan praktikum pada minggu pertama ini adalah mendeteksi objek berwarna merah secara *real-time* dengan menggunakan OpenCV. Untuk prosesnya diawali dengan menginisialisasi kamera menggunakan **cv2.VideoCapture(0)** dengan kamera bawaan PC/Laptop atau dengan *webcam*. Kemudian, setiap frame yang diambil dikonversi ke dalam format warna HSV karena warnanya lebih stabil terhadap perubahan daripada BGR dengan menggunakan **cv2.cvtColor(frame,**

**cv2.COLOR\_BGR2HSV**) Rentang warna merah didefinisikan dengan nilai HSV yaitu upper Red **upper\_red = np.array([10,255,255])** dan dibuat dengan sebuah *mask* yang hanya menampilkan piksel warna merah. Selanjutnya, hasil dari proses ditampilkan dalam tiga tab yang berbeda yaitu *frame* untuk menampilkan gambar/foto/kamera asli, *mask* untuk menampilkan hasil dari warna merah, dan *result* untuk menampilkan hasil dari penyaringan warna. Program ini akan terus berjalan hingga pengguna menekan tombol 'q' yang akan melepaskan akses kamera dan menutup tiga tab. Kemudian, terdapat fitur tambahan berupa *bounding box* untuk menandai objek warna merah yang terdeteksi. Setelah melakukan konversi dari BGR ke HSV, sistem menggunakan fungsi **cv2.findContours()** untuk mendeteksi kontur area yang memiliki warna merah sedangkan untuk menggambar *bounding box* menggunakan **cv2.boundingRect()** yang berfungsi untuk memberikan koordinat dan ukuran kotak pada objek yang berwarna merah tersebut. *Bounding box* ini digambar pada frame asli menggunakan **cv2.rectangle()** dengan nilai merah (255, 0, 0). Adanya penambahan fitur *bounding box* ini meningkatkan kejelasan dan akurasi deteksi, karena tidak hanya menampilkan area yang memiliki warna merah, tetapi juga menandai objek dengan jelas.

- Diskusi

Metode untuk mendeteksi objek berwarna merah dengan penambahan fitur *bounding box* dengan OpenCV ini memiliki kelebihan dan kekurangan. Kelebihan dari metode ini adalah memudahkan dalam mendeteksi objek berdasarkan warna tertentu. Dengan penggunaan warna HSV memungkinkan sistem lebih stabil dalam warna terhadap perubahan pencahayaan dibandingkan dengan BGR. Selain itu, dengan menandai objek yang terdeteksi menggunakan *bounding box* atau kotak pembatas, dapat dengan mudah mengenali dan memverifikasi sistem mendeteksi objek berwarna merah dalam suatu frame. Namun, di sisi lain dengan metode ini memiliki kekurangan atau keterbatasan salah satunya yaitu sistem akan mengalami kesalahan deteksi jika lingkungan pencahayaan berubah atau objek lain dengan warna merah yang serupa. Selain itu,

metode berbasis warna saja tidak dapat membedakan objek berdasarkan bentuk dan hal ini dapat mengurangi keakuratan.

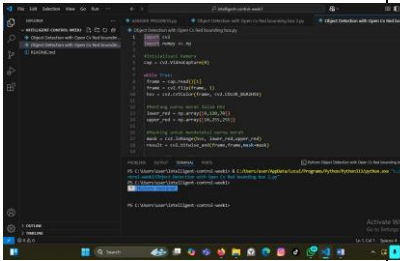
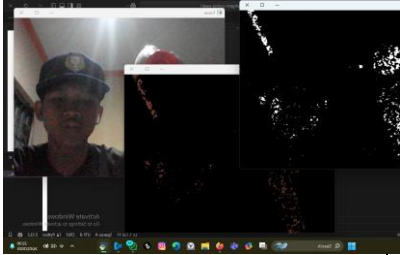
Model warna RGB merupakan singkatan dari Red (Merah) Green (Hijau) Blue(Biru). Model warna RGB juga disebut additive color atau warna pencahayaan karena apabila RGB dikombinasikan maka akan menghasilkan warna putih. RGB merupakan model warna yang paling dasar dalam melakukan penyimpanan gambar (Goenawan dkk., 2022). Pada setiap pixel warna memiliki rentang nilai intensitas mulai dari 0 sampai dengan 255. Setiap titik yang berada pada ruang warna RGB merupakan warna dengan memiliki komponen R, G dan B. Untuk titik (0,0,0) merupakan titik warna yang berwarna hitam, sedangkan titik (1,1,1) merupakan titik warna yang berwarna putih.

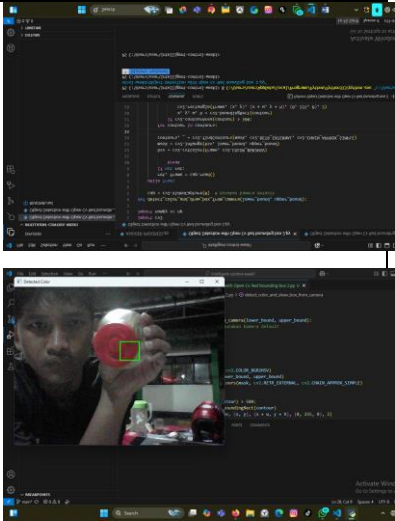
5. *Assigment:*

Dalam praktikum ini bertujuan untuk mendeteksi objek berdasarkan warna menggunakan OpenCV dan bahasa pemrograman Python. Sistem ini dirancang untuk mengenali objek berwarna merah dengan menampilkan bounding box. Proses dimulai dengan menginisialisasi kamera, kemudian citra dikonversi ke format HSV. Selanjutnya, dilakukan masking berdasarkan rentang warna merah, serta ditambahkan bounding box menggunakan fungsi `'cv2.boundingRect()'`. Selain itu, label teks “merah” disisipkan untuk memberikan informasi visual mengenai hasil deteksi. Kehadiran bounding box membuat sistem lebih informatif, memungkinkan pengguna melihat objek yang terdeteksi dengan lebih jelas melalui kamera. Namun, sistem ini memiliki keterbatasan, seperti kemungkinan mendeteksi objek yang tidak relevan akibat latar belakang atau pencahayaan yang memiliki warna serupa.

6. Data dan Output Hasil Pengamatan:

N o.	Variabel	Hasil Pengamatan
1.	- Menginisialisasi kamera, yang kemudian dikonversi ke warna HSV, kode program yaitu	

	<p><code>cv2.cvtColor(frame,</code>  <code>cv2.COLOR_BGR2HSV)</code></p> <ul style="list-style-type: none"> <li>- Rentang warna merah dalam HSV yaitu  <code>lower_red = np.array([0,120,70])</code></li> <li>- Untuk masking mendeteksi warna merah yaitu  <code>mask = cv2.inRange(hs v,</code>  <code>lower_red,upper_red)</code>  <code>result =</code>  <code>cv2.bitwise_and(frame,frame,mask=m</code>  <code>ask)</code></li> <li>- Kemudian untuk menampilkan hasil yaitu <code>cv2.imshow("Frame", frame)</code>  <code>cv2.imshow("Mask", mask)</code>  <code>cv2.imshow("Result", result)</code></li> <li>- Menekan tombol 'q' yang akan melepaskan akses kamera dan menutup tiga tab yaitu <code>cv2.waitKey(1) &amp; 0xFF == ord('q')</code></li> </ul>	 
2.	<ul style="list-style-type: none"> <li>- Menginisialisasi kamera, yang kemudian dikonversi ke warna HSV, yaitu  <code>cv2.cvtColor(frame,cv2.COLOR_BGR</code>  <code>2HSV)</code></li> <li>- Rentang warna merah dalam HSV yaitu <code>lower_red = np.array([0,120,70])</code>  <code>upper_red = np.array([10,255,255])</code></li> <li>- Untuk masking mendeteksi warna merah yaitu <code>mask = cv2.inRange(hs v,</code>  <code>lower_red,upper_red)</code></li> </ul>	

<pre>result=cv2.bitwise_and(frame,frame,mask=mask)</pre> <ul style="list-style-type: none"> <li>- Menemukan kontur yaitu <code>cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)</code></li> <li>- Untuk memunculkan <i>bounding box</i> yaitu <code>cv2.boundingRect(contour)</code></li> <li>- Kemudian untuk menampilkan hasil yaitu <code>cv2.imshow("Frame", frame)</code> <code>cv2.imshow("Mask", mask)</code> <code>cv2.imshow("Result", result)</code></li> <li>- Menyelesaikan program, dengan menekan tombol 'q' yaitu <code>cv2.waitKey(1) &amp; 0xFF == ord('q')</code></li> </ul>	
--	--

#### 7. Kesimpulan:

Berdasarkan praktikum dan analisis yang telah dilakukan, maka dapat diambil kesimpulan yaitu:

- Sistem Metode atau sistem yang diterapkan untuk mengenali objek berwarna merah menggunakan OpenCV dan Python.
- Penambahan fitur bounding box meningkatkan kejelasan hasil deteksi dan mempermudah dalam mengidentifikasi objek.
- Sistem Metode atau sistem ini memiliki keterbatasan dalam sensitivitas terhadap pencahayaan serta berisiko mendeteksi objek yang tidak sesuai.

#### 8. Saran:

Untuk meningkatkan akurasi secara optimal, beberapa langkah dapat diterapkan. Pertama, metode morfologi dapat digunakan untuk mengurangi noise pada hasil deteksi. Kedua, teknik Convolutional Neural Networks (CNN) dapat dimanfaatkan untuk memungkinkan identifikasi objek selain berdasarkan warna.

#### 9. Daftar Pustaka:

- Alam, S., Zainal, M., & Fazil, E. (2024). *PERANCANGAN SISTEM PENGENALAN WAJAH MENGGUNAKAN PYTHON, OPENCV DAN HAARCASCADE*. 9.
- Dosari, F. H. M. A., & Abouellail, S. I. A. D. (2023). Artificial Intelligence (AI) Techniques for Intelligent Control Systems in Mechanical Engineering. *American Journal of Smart Technology and Solutions*, 2(2), 55–64. <https://doi.org/10.54536/ajsts.v2i2.2188>
- Goenawan, A. D., Rachman, M. B. A., & Pulungan, M. P. (2022). Identifikasi Warna Pada Objek Citra Digital Secara Real Time Menggunakan Pengolahan Model Warna HSV. *Jurnal Teknik Informatika dan Elektro*, 4(1), 68–74. <https://doi.org/10.55542/jurtie.v4i1.430>
- Marsella, M., Wijaya, C. S., Wijaya, I., Shidqi, M. T., & Novita, D. (2023). ANALISIS IMPLEMENTASI ARTIFICIAL INTELLIGENCE UNTUK BISNIS: SYSTEMATIC LITERATURE REVIEW. *DEVICE : JOURNAL OF INFORMATION SYSTEM, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY*, 4(2), 133–145. <https://doi.org/10.46576/device.v4i2.4037>
- Zebua, E. T. P., & Rosyani, P. (2024). *Perancangan Deteksi Objek Kendaraan Bermotor Berbasis OpenCV Python menggunakan Metode HOG-SVM untuk Analisis Lalu Lintas Cerdas*. 2(1).