

Laporan Praktikum Kontrol Cerdas

Nama : Cecillia Yowien Arthamevia
NIM : 224308031
Kelas : TKA-6B
Akun Github : <https://github.com/CecilliaYowienArthamevia>
Student Lab Assistant : Salma Halizah Zuhroh

1. Judul Percobaan: *Object Detection with OpenCV*

2. Tujuan Percobaan:

Tujuan dari percobaan deteksi objek dengan *OpenCV* sebagai berikut:

- 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 computer (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

Berdasarkan percobaan praktikum yang dilakukan pada minggu pertama yaitu dengan mendeteksi objek yang memiliki warna merah, biru dan hijau secara *real-time* dengan menggunakan OpenCV. Sesuai dengan

kode program yang telah dibuat diawali dengan menginisialisasi kamera dengan menggunakan kode program **cv2.VideoCapture**, apabila program berjalan maka kamera *webcam* akan menyala. Selanjutnya setiap frame yang akan terdeteksi dikonversi ke dalam format warna HSV, dikarenakan HSV memiliki warna yang lebih stabil terhadap perubahan dengan menggunakan kode program **cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)**. Kemudian mendefinisikan rentang warna merah, biru dan hijau dalam nilai HSV dengan menggunakan kode program warna merah **lower_red1 = np.array([0, 120, 70])**, **upper_red1 = np.array([10, 255, 255])**, **lower_red2 = np.array([170, 120, 70])**, **upper_red2 = np.array([180, 255, 255])**, warna biru **lower_blue = np.array([100, 150, 50])**, **upper_blue = np.array([140, 255, 255])**, dan warna hijau **lower_green = np.array([40, 40, 40])**, **upper_green = np.array([80, 255, 255])**. Dimana masing-masing dibuat dengan sebuah *mask* yang hanya menampilkan piksel warna merah, biru dan hijau dengan nilai 255 dan warna lain menjadi hitam 0. Kemudian menggabungkan semua mask yang dibuat dengan menggunakan kode program **mask = mask_red + mask_blue + mask_green**, **result = cv2.bitwise_and(frame, frame, mask=mask)**. Kemudian menentukan kontur untuk objek yang terdeteksi dengan menggunakan kode program **contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)**, dimana dalam menentukan kontur tersebut terdapat beberapa kemungkinan yang terjadi yang di uraikan dengan menggunakan kode **if** dan **elif**. Selanjutnya membuat bounding box sesuai dengan warna yang terdeteksi dengan kode program **cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)**, **cv2.putText(frame, label, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)**. Selanjutnya, hasil dari proses ditampilkan dalam tiga tab yang berbeda yaitu *frame* untuk menampilkan gambar/foto/kamera asli yang apabila terdapat objek yang berwarna merah, biru dan hijau akan ditandai dengan bounding box sesuai dengan warna objek yang terdeteksi, *mask* untuk menampilkan hasil dari warna merah, biru dan hijau dari objek yang

terdeteksi dan **result** untuk menampilkan hasil dari penyaringan warna. Program akan terus berjalan, apabila ingin mengakhiri berjalannya program maka diperlukan menekan tombol 'q' yang berfungsi untuk melepaskan akses kamera dan menutup tiga tab.

- Diskusi

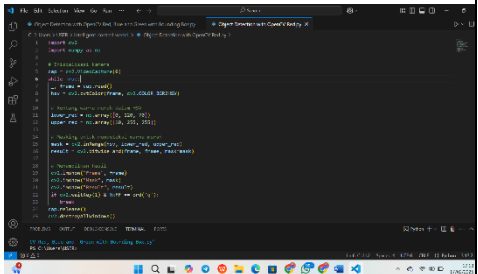
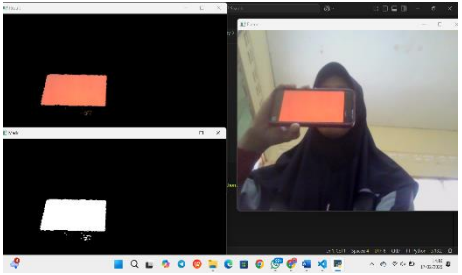
Metode untuk mendeteksi objek berwarna merah, biru dan hijau dan dilengkapi dengan fitur *bounding box* dengan OpenCV ini memiliki kelebihan dan kekurangan. Kelebihan dari metode tersebut adalah memudahkan dalam mendeteksi objek berdasarkan warna tertentu. Dengan penggunaan warna HSV memungkinkan sistem 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 untuk mendeteksi objek berwarna merah, biru dan hijau dalam suatu frame. Namun, selain itu dengan menggunakan metode ini memiliki kekurangan atau keterbatasan salah satunya yaitu sistem akan mengalami kesalahan deteksi jika lingkungan pencahayaan berubah atau objek lain dengan warna yang serupa. Selain itu, metode berbasis warna saja tidak dapat membedakan objek berdasarkan bentuk dan hal ini dapat mengurangi keakuratan.

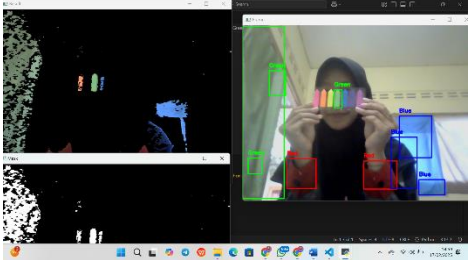
5. Assignment:

Dalam praktikum ini yaitu untuk mendeteksi warna objek menggunakan OpenCV dan dengan menggunakan bahasa pemrograman Python. Pada sistem ini dirancang untuk mendeteksi warna merah, biru dan hijau dengan *bounding box*. Dimulai dari menginisialisasi kamera, yang kemudian dikonversi ke warna HSV, setelah itu dilakukan menyatukan masking yaitu warna merah, biru dan hijau dan ditambahkan fitur *bounding box* menggunakan kode program **cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)**, **cv2.putText(frame, label, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)** dan label teks sesuai warna yang teridentifikasi yang digunakan untuk memberikan informasi visual dari objek yang terdeteksi. Dengan adanya *bounding box*, sistem menjadi lebih informatif, karena pengguna dapat dengan jelas melihat

objek yang dideteksi dalam kamera, akan tetapi terdapat kekurangan seperti kemungkinan dapat mendeteksi objek yang tidak diinginkan, seperti latar belakang atau pencahayaan memiliki warna yang serupa.

6. Data dan Output Hasil Pengamatan:

No.	Variabel	Hasil Pengamatan
1.	<ul style="list-style-type: none"> - Menginisialisasi kamera, yang kemudian dikonversi ke warna HSV, kode program yaitu <code>cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)</code> - Rentang warna merah dalam HSV yaitu <code>lower_red = np.array([0, 120, 70])</code> <code>upper_red = np.array([10, 255, 255])</code> - Selanjutnya menentukan masking untuk mendeteksi warna merah <code>mask = cv2.inRange(hsv, lower_red, upper_red)</code> <code>result = cv2.bitwise_and(frame, frame, mask=mask)</code> - Kemudian untuk menampilkan hasil yaitu <code>cv2.imshow("Frame", frame)</code> <code>cv2.imshow("Mask", mask)</code> <code>cv2.imshow("Result", result)</code> - Menekan tombol 'q' yang akan melepaskan akses kamera 	 

<ul style="list-style-type: none"> - Rentang warna hijau dalam HSV yaitu <code>lower_green = np.array([40, 40, 40])</code> <code>upper_green = np.array([80, 255, 255])</code> <code>mask_green = cv2.inRange(hsv, lower_green, upper_green)</code> - Selanjutnya menggabungkan semua mask <code>mask = mask_red + mask_blue + mask_green</code> <code>result = cv2.bitwise_and(frame, frame, mask=mask)</code> - Menemukan kontur yaitu <code>cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)</code> - Untuk memunculkan <i>bounding box</i> yang sesuai dengan warna objek terdeteksi yaitu <code>cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)</code> <code>cv2.putText(frame, label, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)</code> - Kemudian untuk menampilkan hasil yaitu <code>cv2.imshow("Frame", frame)</code> <code>cv2.imshow("Mask", mask)</code> 	
--	--

	<pre>cv2.imshow("Result", result)</pre> <ul style="list-style-type: none"> - Menyelesaikan program, dengan menekan tombol 'q' yaitu <code>cv2.waitKey(1) & 0xFF == ord('q')</code> 	
--	---	--

7. Kesimpulan:

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

- Metode yang digunakan dalam praktikum ini adalah untuk mendeteksi warna objek menggunakan OpenCV dengan menggunakan software Visual Studio Code.
- Fitur bounding box dapat mengidentifikasi objek, meningkatkan analisis ukuran dan posisi.
- Sistem ini memiliki kelebihan yaitu akurasi yang baik dalam deteksi dengan pengaturan HSV yang tepat, fleksibel, dan dapat diimplementasikan secara *real-time*.
- Sistem ini sensitif terhadap perubahan pencahayaan, kesulitan memisahkan objek yang tumpang tindih, dan dapat terjadi kesalahan mendeteksi warna yang serupa.

8. Saran:

Untuk meningkatkan akurasi deteksi warna suatu objek, dapat menggunakan beberapa teknik, diantaranya metode morfologi dapat digunakan untuk membersihkan noise pada hasil deteksi, sehingga objek biru terdeteksi dengan lebih presisi. Selain itu, penggunaan Convolutional Neural Networks (CNN) yang memungkinkan sistem tidak hanya mendeteksi objek berdasarkan warna, akan tetapi juga berdasarkan fitur lainnya. Kombinasi kedua teknik ini dapat meningkatkan akurasi sistem deteksi objek biru secara signifikan.

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 Lahu Lintas Cerdas*. 2(1).