

Laporan Praktikum Kontrol Cerdas

Nama : Maheswara Aryasatya Nugraha
NIM : 224308036
Kelas : TKA-6B
Akun Github : <https://github.com/maheswaraaryasatya>
Student Lab Assistant : Mas Alfian

1. Judul Percobaan: *Object Detection with OpenCV & Kaggle Dataset*
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 (Artificial Intelligence atau AI) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan seperti mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia (Setia, 2019). *Intelligent Control* adalah metode pengendalian sistem yang menggunakan AI, *Machine Learning*, dan *Deep Learning* untuk meningkatkan performa dan efisiensi (Dosari & Abouellail, 2023). 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).

Software yang digunakan untuk praktikum ini berupa Python dan Open CV. Python merupakan bahasa pemrograman yang multifungsi dan dapat digunakan untuk *Machine Learning* dan *Deep Learning*. Python memiliki

penulisan sintaksis yang mudah, memiliki *library* yang lengkap dan memiliki dukungan komunitas yang kuat karena python bersifat *open source* (Sulistiyowati dkk., 2024). Untuk menuliskan *source code* python anda dapat menggunakan IDE seperti VS code, sublime text, PyCharm, dan lainnya. OpenCV (Open Source Computer Vision Library) adalah perpustakaan komputer open source visi dan perangkat lunak pembelajaran dengan metode yang paling cepat dan memiliki *library* paling lengkap untuk *computer vision* (Muchtar & Apriadi, 2019). OpenCV dibangun untuk menyediakan infrastruktur umum untuk aplikasi visi komputer dan untuk mempercepat penggunaan persepsi mesin dalam produk komersial.

4. Analisis dan Diskusi:

- Analisis

Percobaan praktikum pada minggu pertama ini adalah mendeteksi objek berwarna biru secara *real-time* dengan menggunakan OpenCV. Untuk prosesnya diawali dengan menginisialisasi kamera menggunakan **cv2.VideoCapture** 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 biru didefinisikan dengan nilai HSV yaitu **lower_blue = np.array([100, 150, 50])** dan **upper_blue = np.array([140, 255, 255])** dan dibuat dengan sebuah *mask* yang hanya menampilkan piksel warna biru dengan nilai putih 255 dan warna lain menjadi hitam 0. Selanjutnya, hasil dari proses ditampilkan dalam tiga tab yang berbeda yaitu *frame* untuk menampilkan gambar/foto/kamera asli, *mask* untuk menampilkan hasil dari warna biru, 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 biru yang terdeteksi. Setelah melakukan konversi dari BGR ke HSV, sistem menggunakan fungsi **cv2.findContours()** untuk mendeteksi kontur area yang memiliki

warna biru, sedangkan untuk menggambar *bounding box* menggunakan **cv2.boundingRect()** yang berfungsi untuk memberikan koordinat dan ukuran kotak pada objek yang berwarna biru tersebut. *Bounding box* ini digambar pada frame asli menggunakan **cv2.rectangle()** dengan nilai biru (255, 0, 0). Adanya penambahan fitur *bounding box* ini meningkatkan kejelasan dan akurasi deteksi, karena tidak hanya menampilkan area yang memiliki warna biru, tetapi juga menandai objek dengan jelas.

- Diskusi

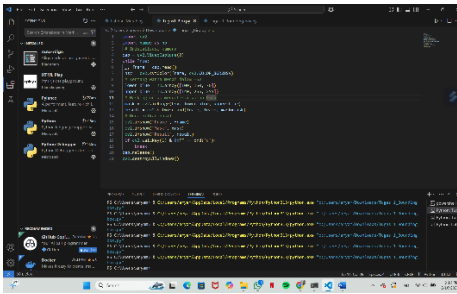
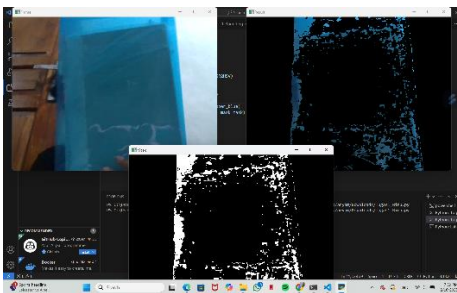
Metode untuk mendeteksi objek berwarna biru dengan penambahan fitur *bounding box* dengan OpenCV ini memiliki kelebihan dan kekurangan. Kelebihan dari metode ini adalah memudahkan dalam mendeteksi objek berdasarkan warna tertentu. Dengan penggunaan warna HSV memungkinkan sistem lebih stabil dalam warna 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 biru 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 biru yang serupa. Selain itu, metode berbasis warna saja tidak dapat membedakan objek berdasarkan bentuk dan hal ini dapat mengurangi keakuratan.

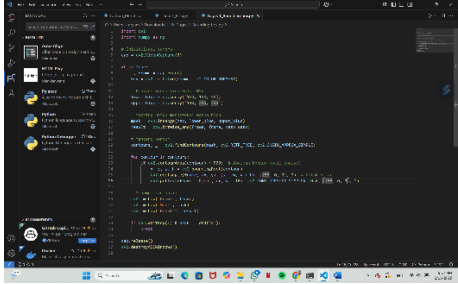
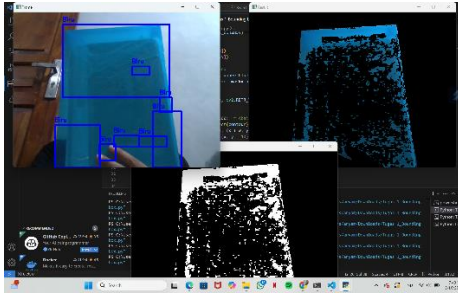
5. *Assigment*:

Dalam praktikum ini adalah untuk mendeteksi objek warna berbasis warna menggunakan OpenCV dan dengan menggunakan bahasa pemograman Python. Pada sistem ini dirancang untuk mendeteksi warna biru dengan *bounding box*. Dimulai dengan menginisialisasi kamera, yang kemudian dikonversi ke warna HSV, setelah itu dilakukan masking berdasarkan rentang warna biru dan ditambahkan fitur *bounding box* menggunakan fungsi **cv2.boundingRect()** dan label teks “biru” ditambahkan untuk memberikan informasi visual mengenai hasil deteksi. Dengan adanya *bounding box*, sistem menjadi lebih informatif, karena pengguna dapat dengan jelas melihat objek

yang dideteksi dalam kamera. Namun, terdapat kekurangan seperti kemungkinan deteksi 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 biru dalam HSV yaitu <code>lower_blue = np.array([100, 150, 50])</code> <code>upper_blue = np.array([140, 255, 255])</code> - Untuk masking mendeteksi warna biru yaitu <code>mask = cv2.inRange(hsv, lower_blue, upper_blue)</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 dan menutup tiga tab yaitu <code>cv2.waitKey(1) & 0xFF == ord('q')</code> 	 

2.	<ul style="list-style-type: none"> - Menginisialisasi kamera, yang kemudian dikonversi ke warna HSV, yaitu <code>cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)</code> - Rentang warna biru dalam HSV yaitu <code>lower_blue = np.array([100, 150, 50])</code> <code>upper_blue = np.array([140, 255, 255])</code> - Untuk masking mendeteksi warna biru yaitu <code>mask = cv2.inRange(hsv, lower_blue, upper_blue)</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> yaitu <code>cv2.boundingRect(contour)</code> - Kemudian untuk menampilkan hasil yaitu <code>cv2.imshow("Frame", frame)</code> <code>cv2.imshow("Mask", mask)</code> <code>cv2.imshow("Result", result)</code> - 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:

- Sistem atau metode yang digunakan untuk mendeteksi objek berwarna biru menggunakan OpenCV dan Python.
- Penambahan fitur bounding box membuat hasil deteksi lebih jelas dan memudahkan mengidentifikasi objek.
- Sistem atau metode ini memiliki keterbatasan dalam sensitivitas pencahayaan dan kemungkinan mendeteksi objek yang tidak relevan.

8. Saran:

Untuk meningkatkan akurasi yang tepat, terdapat beberapa hal yang dapat dilakukan yaitu pertama, menggunakan metode morfologi untuk mengurangi noise pada hasil deteksi. Kemudian yang kedua, menggunakan teknik Convolutional Neural Networks (CNN) untuk memungkinkan identifikasi objek selain warna.

9. Daftar Pustaka:

- 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>
- Muchtar, H., & Apriadi, R. (2019). Implementasi Pengenalan Wajah Pada Sistem Penguncian Rumah Dengan Metode Template Matching Menggunakan Open Source Computer Vision Library (OpenCV). *RESISTOR (elektRONika kEndali telekomunikaSI tenaga liSTrik kOmputeR)*, 2(1), 39. <https://doi.org/10.24853/resistor.2.1.39-42>
- Setia, B. (2019). Penerapan Logika Fuzzy pada Sistem Cerdas. *Jurnal Sistem Cerdas*, 2(1), 61–66. <https://doi.org/10.37396/jsc.v2i1.18>
- Sulistiyowati, I., Maulana Ichsan, H., & Anshory, I. (2024). KONVEYOR PENYORTIR OBJEK DENGAN DETEKSI WARNA MENGGUNAKAN KAMERA ESP-32 BERBASIS OPEN-CV PYTHON. *Prosiding TAU SNARS-TEK Seminar Nasional Rekayasa dan Teknologi*, 4(1), 35–41. <https://doi.org/10.47970/snarstek.v2i1.711>