

Laporan Praktikum Kontrol Cerdas

Nama : Zibran Rizki Arta
NIM : 224308049
Kelas : TKA-6B
Akun Github : <https://github.com/Zibranrizki25>
Student Lab Assistant : Mbak Salma Halizah

1. Judul Percobaan: *Object Detection with OpenCV*

2. Tujuan Percobaan:

Tujuan percobaan deteksi Objek Deteksi Menggunakan OpneCV, 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

Praktikum percobaan pada minggu pertama ini adalah mendeteksi objek berwarna merah, kemudian dilakukan modifikasi program untuk

mendeteksi objek berwarna merah, hijau dan biru secara *real-time* dengan menggunakan OpenCV. Selain itu, juga dilakukan modifikasi menggunakan fitur *bounding box* untuk menandai objek yang terdeteksi.

Tahap pertama yang dilakukan adalah menginisialisasi kamera menggunakan **cv2.VideoCapture** dengan kamera bawaan PC/Laptop atau dengan *webcam*. Kemudian, untuk menjaga konsistensi warna meskipun terdapat perubahan pencahayaan, setiap frame hasil tangkapan diubah format warnanya menjadi HSV. Proses konversi ini memanfaatkan kode **hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)** untuk mengubah dari format BGR asli ke HSV.

Rentang warna merah didefinisikan dengan menggunakan kode **lower_red = np.array([0, 120, 70])** dan **upper_red = np.array([10, 255, 255])** serta dibuat sebuah mask dengan fungsi **mask_red = cv2.inRange(hsv, lower_red, upper_red)** dan **result_red = cv2.bitwise_and(frame, frame, mask=mask_red)**.

Rentang warna biru didefinisikan dengan menggunakan fungsi **lower_blue = np.array([100, 150, 0])** dan **upper_blue = np.array([140, 255, 255])** serta dibuat sebuah mask dengan memanfaatkan fungsi **mask_blue = cv2.inRange(hsv, lower_blue, upper_blue)** dan **result_blue = cv2.bitwise_and(frame, frame, mask=mask_blue)**.

Rentang warna hijau didefinisikan dengan menggunakan kode **lower_green = np.array([40, 40, 40])** dan **upper_green = np.array([80, 255, 255])** serta dibuat sebuah mask dengan memanfaatkan fungsi **mask_green = cv2.inRange(hsv, lower_green, upper_green)** dan **result_green = cv2.bitwise_and(frame, frame, mask=mask_green)**.

Setelah itu, digunakan kode **mask = mask_red + mask_blue + mask_green** dan **result = cv2.bitwise_and(frame, frame, mask=mask)** untuk menggabungkan ketiga mask, kode ini akan menghasilkan satu mask yang mencakup semua area yang terdeteksi berwarna merah, biru, atau hijau.

Kemudian penambahan fitur kontur dan *bounding box* dengan menggunakan kode **cv2.findContours()** untuk mendeteksi kontur area

yang memiliki warna merah, biru atau hijau, sedangkan untuk menggambar *bounding box* menggunakan **cv2.boundingRect()** yang berfungsi untuk memberikan koordinat dan ukuran kotak pada objek yang berwarna merah, biru atau hijau tersebut. *Bounding box* ini digambar pada frame asli menggunakan **cv2.rectangle()**. Penambahan fitur *bounding box* secara signifikan meningkatkan kemampuan sistem dalam memahami dan berinteraksi dengan informasi visual serta kejelasan dan akurasi deteksi, karena tidak hanya menampilkan area yang memiliki warna, tetapi juga menandai objek dengan jelas.

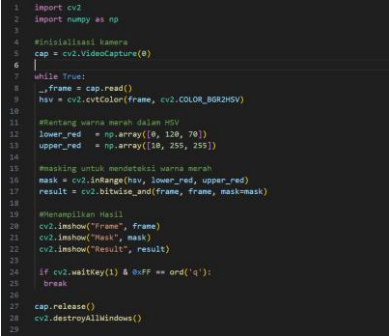
- Diskusi

Metode untuk mendeteksi objek berwarna merah, hijau dan 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 lebih cepat dan efisien, sehingga cocok untuk aplikasi *real-time* seperti pelacakan objek atau pengawasan. Selain itu, penambahan fitur *bounding box* memberikan informasi penting tentang lokasi dan ukuran objek yang terdeteksi dengan warna berbeda beda dalam satu frame. Namun, di sisi lain dengan metode ini memiliki kekurangan atau keterbatasan salah satunya yaitu rentan terhadap *noise* atau gangguan pada citra. *Noise* dapat menyebabkan deteksi palsu atau mengurangi akurasi deteksi. Selain itu, sistem akan mengalami kesalahan deteksi jika lingkungan pencahayaan berubah atau objek lain dengan warna yang serupa.

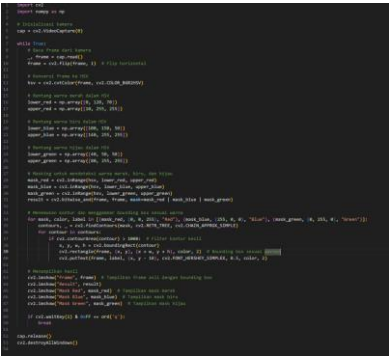

5. Assigment:

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 “nama warna” 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.	<p>- Menginisialisasi frame awal didalam formt BGR, yang kemudian dikonversi ke warna HSV, kode program yaitu:</p> <pre>cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)</pre> <p>- Rentang warna biru dalam HSV yaitu:</p> <pre>power_red = np.array([0, 120, 70]) upper_red = np.array([10, 255, 255])</pre> <p>- Untuk masking mendeteksi warna merah, dimana ketika program</p>	 <pre> 1 import cv2 2 import numpy as np 3 4 #inisialisasi kamera 5 cap = cv2.VideoCapture(0) 6 7 while True: 8 _frame = cap.read() 9 hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV) 10 11 #Rentang warna merah dalam HSV 12 lower_red = np.array([0, 120, 70]) 13 upper_red = np.array([10, 255, 255]) 14 15 #masking untuk mendeteksi warna merah 16 mask = cv2.inRange(hsv, lower_red, upper_red) 17 result = cv2.bitwise_and(frame, frame, mask=mask) 18 19 #Menampilkan Hasil 20 cv2.imshow("Frame", frame) 21 cv2.imshow("Mask", mask) 22 cv2.imshow("Result", result) 23 24 if cv2.waitKey(1) & 0xFF == ord('q'): 25 break 26 27 cap.release() 28 cv2.destroyAllWindows() 29 </pre> <p>“Coding Program”</p>

	<p>mendeteksi warna merah maka akan membuat warna merah terdeteksi menjadi putih, kode programnya yaitu:</p> <pre>Mask = cv2.inRange(hsv, lower_red, upper_red)</pre> <ul style="list-style-type: none"> - Untuk Result mendeteksi warna merah, hanya di bagian warna merah dari frame, sementara untuk warna lainya diubah menjadi hitam kode programnya yaitu: <pre>result = cv2.bitwise_and(frame, frame, mask=mask)</pre> - Kemudian untuk hasil dari program ini menampilkan tiga jendela kamera, yang masing – masing menampilkan jendela frame, jendela mask, dan jendela result, kode programnya yaitu: <pre>cv2.imshow("Frame", frame) cv2.imshow("Mask", mask) cv2.imshow("Result", result)</pre> - Untuk mengakhiri pengulangan berjalannya program, tekan pada tombol ‘q’ yang akan melepaskan akses kamera dan menutup tiga jendela kamera yaitu: <pre>cv2.waitKey(1) & 0xFF == ord('q')</pre> 	 <p>“Jendela Frame”</p>  <p>“Jendela Mask”</p>  <p>“Jendela Result”</p>
--	---	--

2.	<p>- Menginisialisasi frame awal didalam formt BGR, yang kemudian dikonversi ke warna HSV, kode program yaitu:</p> <pre>cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)</pre> <p>- Rentang warna merah, biru, hijau dalam HSV yaitu:</p> <pre>lower_red = np.array([0, 120, 70]) upper_red = np.array([10, 255, 255]) lower_blue = np.array([100, 150, 50]) upper_blue = np.array([140, 255, 255]) lower_green = np.array([40, 50, 50]) upper_green = np.array([80, 255, 255])</pre> <p>- Untuk masking mendeteksi warna merah, biru, hijau dimana ketika program mendeteksi ketiga warna tersebut maka akan membuat warnanya menjadi putih, kode programnya yaitu:</p> <pre>mask_red = cv2.inRange(hsv, lower_red, upper_red) mask_blue = cv2.inRange(hsv, lower_blue, upper_blue)</pre>	 <p>“Coding Program”</p>  <p>“Jendela Frame”</p>
----	---	--

	<pre>mask_green = cv2.inRange(hsv, lower_green, upper_green)</pre> <p>- Untuk Result mendeteksi warna merah, biru, hijau hanya di bagian ketiga ketiga warna tersebut yang muncul dari frame, sementara untuk warna lainya kecuali warna merah, biru, dan hijau diubah menjadi hitam, kode programnya yaitu:</p> <pre>result = cv2.bitwise_and(frame, frame, mask=mask_red mask_blue mask_green)</pre> <p>-Untuk menemukan kontur dari warna merah, biru, hijau. Nantinya setiap warna akan diinisialisasikan kotak warnanya akan berbeda yang disesuaikan dengan warna apa yang terdeteksi, yaitu:</p> <pre>for mask, color, label in [(mask_red, (0, 0, 255), "Red"), (mask_blue, (255, 0, 0), "Blue"), (mask_green, (0, 255, 0), "Green")]: contours,_=cv2.findContours(mask ,cv2.RETR_TREE,cv2.CHAIN_A PPROX_SIMPLE)</pre> <p>-Untuk memunculkan <i>bounding box</i>, kode programnya yaitu:</p> <pre>vc2.boundingRect(contour)</pre>	 <p>“Jendela Mask Red”</p>  <p>“Jendela Mask Green”</p>
--	---	--

- Untuk memunculkan label warna pada kotak bounding box, kode programnya yaitu:

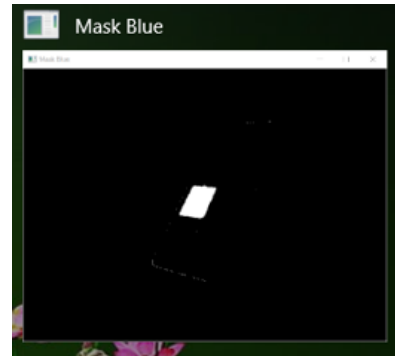
```
cv2.putText(frame, label, (x, y - 10),  
cv2.FONT_HERSHEY_SIMPLEX  
, 0.5, color, 2)
```

Kemudian untuk hasil dari program ini menampilkan lima jendela kamera, yang masing – masing menampilkan jendela frame, tiga jendela mask, dan jendela result, kode programnya yaitu:

```
cv2.imshow("Frame", frame)  
cv2.imshow("Result", result)  
cv2.imshow("Mask Red",  
mask_red)  
cv2.imshow("Mask Blue",  
mask_blue)  
cv2.imshow("Mask Green",  
mask_green)
```

Untuk mengakhiri pengulangan berjalannya program, tekan pada tombol ‘q’ yang akan melepaskan akses kamera dan menutup tiga jendela kamera yaitu:

```
cv2.waitKey(1) & 0xFF == ord('q')
```



“Jendela Mask Blue”



“Jendela Result”

7. Kesimpulan:

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

- Pada program pertama, difungsikan untuk mendeteksi warna merah dengan cara mengkonversi format warna BGR ke format HSV dan menampilkan hasil dalam tiga jendela yaitu, jendela frame, jendela mask, dan jendela result.
- Masking program pertama, dilakukan hanya dengan satu rentang warna merah.
- Pada program kedua, difungsikan untuk mendeteksi warna merah, biru, dan hijau dengan cara mengkonversi format warna BGR ke format HSV dan menampilkan hasil dalam lima jendela yaitu, jendela frame, tiga jendela mask, dan jendela result.
- Pada program kedua, dilengkapi fitur bounding box untuk memberikan batas visual atau kotak pada warna merah, biru, dan hijau yang terdeteksi.

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:

- 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).