Laporan Praktikum Kontrol Cerdas

Nama	**: Nefi Afif Sujatyana**
NIM	**: 224308093**
Kelas	**: TKA-7D**
Akun Github	**: https://github.com/nefiafif612-cmd**

1. **Judul Percobaan:** Deteksi Objek Sederhana dengan *OpenCV*
2. **Tujuan Percobaan:**

   Tujuan dari percobaan Deteksi Objek Sederhana dengan *OpenCV* adalah :

   - 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 beruapa 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 merupakan proses untuk mengidentifikasi dan menentukan keberadaan objek tertentu dalam citra atau video. Dalam bidang pengolahan citra digital dan visi komputer, proses ini dilakukan dengan menerapkan algoritma dan teknik khusus guna mengenali berbagai jenis objek secara otomatis.

Model warna RGB merupakan kependekan dari Red (merah), Green (hijau), dan Blue (biru). Model ini dikenal pula sebagai *additive color* atau warna aditif, karena perpaduan ketiga komponen tersebut dalam intensitas penuh 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, dan kemudian dilakukan modifikasi program agar bisa mendeteksi objek berwarna seperti warna 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.

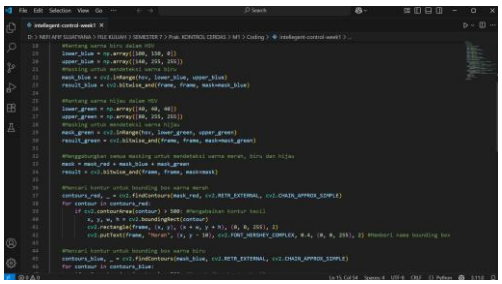
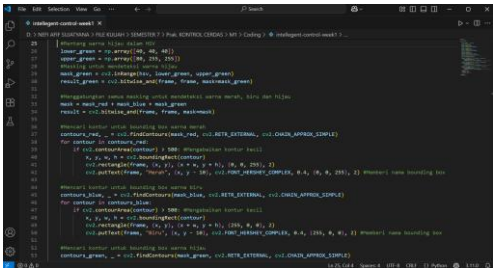
- Diskusi

  Metode deteksi objek berwarna merah, hijau, dan biru menggunakan OpenCV dengan penambahan *bounding box* memiliki keunggulan dalam kemudahan identifikasi objek berdasarkan warna. Penggunaan ruang warna HSV menjadikan proses deteksi lebih efisien dan cocok untuk aplikasi waktu nyata. Fitur *bounding box* juga memberikan informasi posisi dan ukuran objek secara visual. Dengan demikian, metode ini memiliki keterbatasan, seperti sensitivitas terhadap *noise* dan perubahan pencahayaan, yang dapat menurunkan akurasi serta menyebabkan kesalahan dalam pendeteksian objek dengan warna serupa.

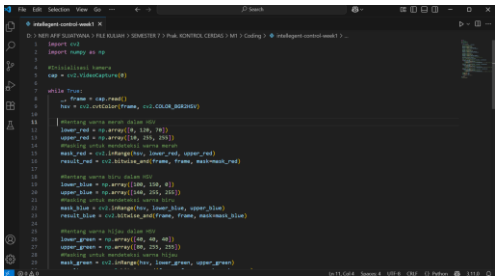
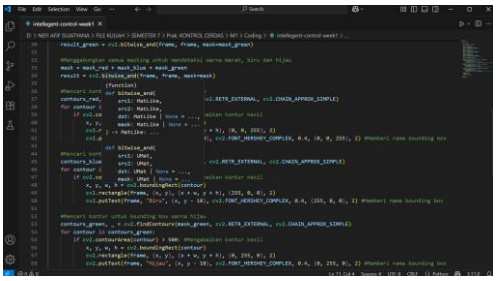
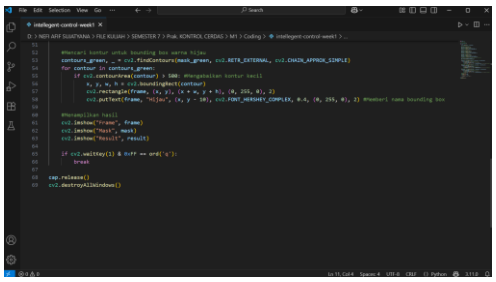
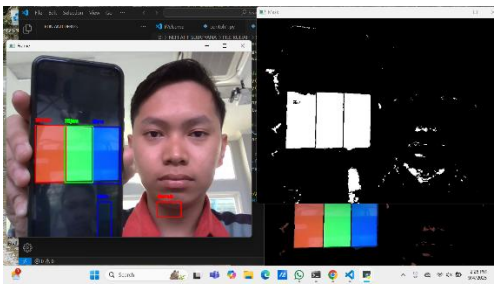
5. *Assignment*:

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.	- Menginisialisasi kamera, yang kemudian dikonversi ke warna HSV, kode program yaitu cv2.cvtColor(frame, cv2.COLOR_BGR2HSV) - Rentang warna biru dan hijau dalam HSV sebagai berikut: - Warna biru yaitu : lower_blue = np.array([100, 150, 0]) upper_blue = np.array([140, 255, 255]) - Warna hijau yaitu : lower_green = np.array([40, 40, 40]) upper_green = np.array([80, 255, 255]) - Masking untuk mendeteksi warna biru dan hijau yaitu sebagai berikut: - Warna biru : mask_blue = cv2.inRange(hsv, lower_blue, upper_blue) result_blue = cv2.bitwise_and(frame, frame, mask=mask_blue) - Warna hijau	 

	mask_green = cv2.inRange(hsv, lower_green, upper_green) result_green = cv2.bitwise_and(frame, frame, mask=mask_green) - Selanjutnya untuk menampilkan hasil sebagai berikut : cv2.imshow("Frame", frame) cv2.imshow("Mask Blue", mask_blue) cv2.imshow("Result Blue", result_blue) cv2.imshow("Mask Green", mask_green) cv2.imshow("Result Green", result_green) - Menekan tombol 'q' yang melepaskan akses kamera dan menutup tiga tab yaitu cv2.waitKey(1) & 0xFF == ord('q')	
2.	- Menginisialisasi kamera, dan kemudian dikonversi ke warna HSV, dengan kode program yaitu : cv2.cvtColor(frame, cv2.COLOR_BGR2HSV) - Rentang untuk warna merah, biru dan hijau dalam HSV sebagai berikut : - Warna merah yaitu : lower_red = np.array([0, 120, 70])	

	upper_red = np.array([10, 255, 255]) - Warna biru yaitu : lower_blue = np.array([100, 150, 0]) upper_blue = np.array([140, 255, 255]) - Warna hijau yaitu : lower_green = np.array([40, 40, 40]) upper_green = np.array([80, 255, 255]) - Masking untuk mendeteksi warna merah, biru dan hijau sebagai berikut : - Warna merah yaitu : mask_red = cv2.inRange(hsv, lower_red, upper_red) result_red = cv2.bitwise_and(frame, frame, mask=mask_red) - Warna biru yaitu : mask_blue = cv2.inRange(hsv, lower_blue, upper_blue) result_blue = cv2.bitwise_and(frame, frame, mask=mask_blue) - Warna hijau yaitu : mask_green = cv2.inRange(hsv, lower_green, upper_green)	   

	result_green = cv2.bitwise_and(frame, frame, mask=mask_green) - Untuk menemukan kontur sebagai berikut : cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE) - Untuk memunculkan *bounding box* sebagai berikut : vc2.boundingRect(contour) - Kemudian untuk menampilkan hasil sebagai berikut : cv2.imshow("Frame", frame) cv2.imshow("Mask", mask) cv2.imshow("Result", result) - Menyelesaikan program, dengan menekan tombol 'q' sebagai berikut : cv2.waitKey(1) & 0xFF == ord('q')	

7. Kesimpulan:

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

- Penambahan fitur *bounding box* dapat meningkatkan kejelasan hasil dari deteksi dan memberikan batas visual yang berupa kotak pada objek yang teridentifikasi.
- Sistem atau metode pendeteksian objek menggunakan OpenCV dan Python dapat mengenali objek berdasarkan warna tertentu, seperti merah, hijau, dan biru.

- Metode atau sistem ini memiliki keterbatasan, khususnya terkait sensitivitas terhadap perubahan kondisi pencahayaan serta potensi kesalahan dalam mendeteksi objek yang memiliki warna serupa.

8. Saran:

Untuk meningkatkan ketepatan deteksi objek dapat digunakan metode lain seperti deteksi tepi *(Canny Edge Detection)* atau model berbasis *machine learning*. Selain itu, apabila warna latar belakang mirip dengan objek yang ingin dideteksi, pertimbangkan penggunaan metode lain seperti deteksi bentuk atau fitur objek selain warna. Selain metode HSV, gunakanmodel *deep learning* seperti YOLO atau Faster R-CNN yang lebih handal dalam deteksi objek.

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).