

Laporan Praktikum Kontrol Cerdas

Nama : Moh.Farhan Baihaqi

NIM : 224308037

Kelas : TKA-6B

Akun Github (Tautan) : <https://github.com/farhanbaihaqi21>

Student Lab Assistant :Mbak Dwi

1. Judul Percobaan : ***Object Detection with OpenCV & Kaggle Dataset***

2. Tujuan Percobaan :

- Memahami konsep dasar control cerdas (intelligent control system).
- Menggunakan Python dan OpenCV untuk mendeteksi objek secara sederhana.
- Memahami sistem deteksi warna menggunakan OpenCV.

3. Landasan Teori :

Artificial Intelligence merupakan bidang ilmu computer untuk pembuatan *software* dan *hardwere* yang dapat bekerja seperti kecerdasan manusia. AI digunakan dalam proses otomasi seperti *machine learning*, *natural language processing* dan *computer vision* untuk pengolahan data, untuk mendeteksi object, pengendalian, perencanaan (Fadjeri et al., 2025). *Object detection* merupakan pengembangan untuk mendeteksi dan mengklasifikasi *object* dalam bentuk gambar atau video. Proses deteksi dapat dilakukan dengan berbagai macam metode yang pada dasarnya melakukan pembacaan fitur-itur dari seluruh objek pada citra input. Fitur dari objek pada citra akan dibandingkan dengan fitur dari objek referensi atau template lalu dibandingkan dan ditentukan apakah objek yang dideteksi termasuk objek yang ingin dideteksi. Object detection dapat di gunakan berbagai aplikasi seperti pengawasan, pengenalan wajah, dan kendaraan tanpa pengemudi(Fauzan & Wibowo, 2021).

Software yang digunakan dalam *object detection* adalah bahasa pemrograman *python*. Bahasa pemrograman *python* merupakan bahasa pemrograman tingkat tinggi yang dibuat agar mudah dibaca dan diterapkan , serta bersifat *open source*. *Python*

dapat dijalankan pada berbagai sistem operasi, seperti Mac, Windows, dll. Dalam pengembangan sistem atau perangkat lunak berbasis *python* diperlukan *library* yang dapat digunakan untuk mendeteksi objek (Wantania, 2020). OpenCV (*Open-Source Computer Vision*) merupakan sebuah *library* perangkat lunak *computer vision* dan *machine learning* yang bersifat *open-source*. OpenCV dibangun untuk menyediakan infrastruktur umum untuk aplikasi *computer vision* dan mempercepat persepsi mesin (*machine perception*) (Rizkatama & Nugroho, 2021).

4. Analisis dan Diskusi :

-Analisis

Percobaan dalam praktikum minggu ini adalah deteksi objek warna biru menggunakan OpenCV. Untuk prosesnya dengan mengimport library OpenCV untuk pemrosesan gambar dan video untuk mengolah data gambar. Selanjutnya menyiapkan kamera dari kamera bawaan laptop/webcam dengan menginisiasi kamera ke dalam codingan menggunakan **cv2.VideoCapture**. Selanjutnya membaca frame dari kamera, frame dari kamera di inisiasi dalam codingan yaitu **_,frame=cap.read()** konversi dari **BGR(Blue, Green, Red)** menjadi **HSV (Hue Saturation Value)** dengan inisiasi dalam codingan **hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)**. Kemudian menentukan batas warna bawah dan atas untuk menentukan jenis warna **lower_blue = np.array([100,50,0])** **upper_blue = np.array([140,255,255])**. Kemudian membuat masking untuk mendeteksi warna biru **mask=cv2.inRange(hsv,lower_blue,upper_blue)** fungsinya menampilkan piksel warna biru dengan nilai 255 yang akan berwarna putih dan warna piksel yang bukan warna biru dengan nilai 0 yang akan berwarna putih. **result=cv2.bitwise_and(frame,frame,mask=mask)**. bounding box dengan menentukan objek batas dengan fungsi **contours, _ = cv2.findContours(mask, cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)** Selanjutnya menampilkan objek yang berwarna selain biru akan menjadi warna hitam dan menampilkan warna biru jika objek berwarna biru. Kemudian membuat bounding box untuk mendeteksi warna biru dengan fungsi **x, y, w, h**

= **cv2.boundingRect(contour cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)**. Selanjutnya menampilkan hasil program **cv2.imshow("blue colour",frame) cv2.imshow("Mask",mask) cv2.imshow("Result",result)**. Untuk mengakhiri program dengan menekan tombol 'q' maka jendela windows akan tertutup.

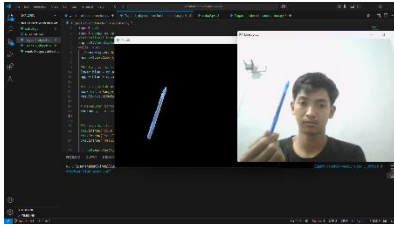
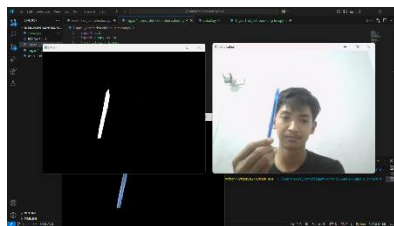
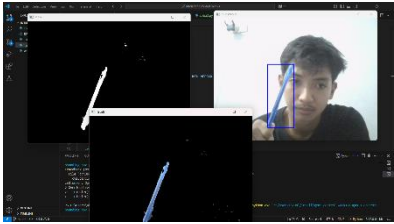
-Diskusi


Dalam AI terdapat proses pengolahan data yang dikembangkan dalam Computer vision untuk memahami dan menganalisis gambar atau video, yang dapat digunakan untuk mendeteksi warna biru. Diperlukan library OpenCV untuk menjalankan program dari *python*. Dalam mendeteksi warna biru diperlukan konversi BGR menjadi HSV karena memungkinkan sistem lebih stabil dalam warna terhadap perubahan pencahayaan. Selanjutnya menggunakan bounding box untuk menentukan batas warna biru agar dapat mendeteksi objek warna biru dalam suatu frame.

5. Assignment :

Dalam program ini menggunakan bahasa pemrograman Python yang membutuhkan library OpenCV untuk mendeteksi warna biru dengan menginisiasi kamera dari BGR ke HSV berdasarkan rentang warna biru. bounding box dengan menentukan objek batas dengan fungsi **contours**, **_ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)**. Kemudian menggunakan fungsi **for contour in contours: if cv2.contourArea(contour) > 500: x, y, w, h = cv2.boundingRect(contour) cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)** ditambahkan untuk membuat rectangle untuk mendeteksi atau membatasi warna biru.

6. Data dan Output Hasil :

No	Variabel	Hasil Pengamatan
1.	<pre> #mengimport library openCV import cv2 import numpy as np #inisialisasi kamera cap=cv2.VideoCapture (0) while True: _,frame=cap.read() hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV) #Rentang warna biru dalam HSV lower_blue = np.array([100,50,0]) upper_blue = np.array([140,255,255]) #Masking untuk mendeteksi warna biru mask=cv2.inRange(hsv,lower_blue,upper_blue) result=cv2.bitwise_and(frame,frame,mask=mask) # Menemukan kontur dari objek yang terdeteksi contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE) #Menampilkan hasil cv2.imshow("blue colour",frame) cv2.imshow("Mask",mask) cv2.imshow("Result",result) if cv2.waitKey(1) & 0xFF== ord('q'): break cap.release() cv2.destroyAllWindows() </pre>	 
2.	<pre> import cv2 import numpy as np #inisialisasi kamera cap=cv2.VideoCapture (0) while True: _,frame=cap.read() hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV) #Rentang warna biru dalam HSV lower_blue = np.array([100,50,0]) upper_blue = np.array([140,255,255]) </pre>	

<pre> #Masking untuk mendeteksi warna biru mask=cv2.inRange(hsv,lower_blue,upper_blue) result=cv2.bitwise_and(frame,frame,mask=mask) # Menemukan kontur dari objek yang terdeteksi contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE) #Bounding box for contour in contours: if cv2.contourArea(contour) > 500: # bounding box di sekitar objek biru x, y, w, h = cv2.boundingRect(contour) cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2) #Menampilkan hasil cv2.imshow("blue colour",frame) cv2.imshow("Mask",mask) cv2.imshow("Result",result) if cv2.waitKey(1) & 0xFF== ord('q'): break cap.release() cv2.destroyAllWindows() </pre>	
--	--

7. Kesimpulan :

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

- Ketika kita membuat kodingan untuk Bounding Box hasil dari output sistem ini memiliki keterbatasan dalam sensitivitas dan Cahaya jika pencahayaan kurang maka hasil nya juga tidak terbaca.
- Sistem yang digunakan untuk mendeksi warna biru menggunakan python dan Visual Studio code.

- Penambahan kodaingan untuk bounding box memudahkan kita untuk mendeteksi objek yang akan dideteksi

8. Saran :

Untuk meningkatkan akurasi deteksi, dapat melakukan penyesuaian pada rentang warna biru yang digunakan dalam masking, serta mempertimbangkan untuk menerapkan teknik pemrosesan citra tambahan, pengurangan noise, sebelum melakukan deteksi kontur.

9. Daftar Pustaka :

Fadjeri, A., Burhaein, E., & Widiyono, I. P. (2025). *Pengembangan Teknologi AI berbasis Computer Vision dan Robotika untuk Latihan Akurasi Olahraga Tennis Lapangan Duduk bagi Atlet Disabilitas Fisik. 1.*

Fauzan, M. R., & Wibowo, A. P. W. (2021). PENDETEKSIAN PLAT NOMOR KENDARAAN MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE V3 DAN TESSERACT. *Jurnal Ilmiah Teknologi Infomasi Terapan*, 8(1), 57–62. <https://doi.org/10.33197/jitter.vol8.iss1.2021.718>

Rizkatama, G. N., & Nugroho, A. (2021). *Sistem Cerdas Penghitung Jumlah Mobil untuk Mengetahui Ketersediaan Lahan Parkir berbasis Python dan YOLO v4.*

Wantania, B. B. M. (2020). *Penerapan Pendeteksian Manusia Dan Objek Dalam Keranjang Belanja Pada Antrian Di Kasir. 1.*