

Laporan Praktikum Kontrol Cerdas

Nama : Rizky Surya Pratama
NIM : 224308045
Kelas : TKA-6B
Akun Github : <https://github.com/Suurya4>
Student Lab Assistant : Mas Dimas

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.
- Dapat Menggunakan Python dan OpenCV untuk mendeteksi objek secara sederhana.
- Dapat memahami mengenai Bounding Box untuk mendeteksi warna.
- Memanfaatkan GitHub untuk version control dan Kaggle sebagai sumber dataset.

3. Landasan Teori:

Kecerdasan buatan sendiri adalah kemampuan mesin untuk meniru dan melakukan tugas yang biasanya membutuhkan kecerdasan manusia. Artificial Intelligence atau kecerdasan buatan dapat disebut begitu dapat mengaplikasikan pengetahuan, dapat mempelajari kehidupan manusia serta merekam berbagai respon yang diberikan oleh manusia untuk mereka (AI) kembangkan dengan kemampuannya sendiri (Afandi & Kurnia, 2023).

Pengolahan citra digital (digital image processing) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/ gambar dilakukan secara digital menggunakan komputer (Sutoyo, et al., 2009). Deteksi objek merupakan salah satu bidang dari computer vision

yang dapat mendeteksi objek semantik dalam gambar atau video. Deteksi objek merujuk sebagai suatu metode yang berfungsi untuk menemukan dan mengidentifikasi keberadaan suatu objek dengan warna tertentu sesuai dengan yang diprogram (Sarmah & Ankur, 2023). Pengolahan Citra Digital ini menggunakan bahasa pemrograman Python, yang fungsinya melihat hasilnya OpenCV. Pada aplikasi pengolahan citra digital pada umumnya, citra digital dapat dibagi menjadi 3, color image, black and white image, dan binary image. Color image atau RGB ini masing-masing piksel memiliki warna tertentu, warna tersebut adalah merah (red), hijau (green) dan biru (blue). Jika masing-masing warna memiliki range 0- 255. Pada kali ini juga menggunakan bounding box untuk mempermudah mendeteksi warna. Bounding Box adalah sebuah kotak imajiner yang digunakan untuk mengelilingi objek yang telah terdeteksi. Bounding Box memiliki bentuk kotak dan ukurannya sama dengan ukuran objek yang terdeteksi (Yusup, et al., 2024). Untuk membuat Bounding Box, koordinat piksel yang diperlukan adalah upper-left(UL), upper-right(UR), lower-left (LL), dan lower-right(LR).

4. Analisis dan Diskusi:

- Analisis

Praktikum control cerdas pada pertemuan ini dilakukan untuk mendeteksi objek berwarna biru secara *real-time* dengan bounding box menggunakan OpenCV dan dengan Bahasa pemrograman python. Untuk prosesnya diawali dengan menginisialisasi kamera menggunakan **cv2.VideoCapture** dengan kamera bawaan PC/Laptop atau dengan *webcam*. Kemudian **cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)** digunakan untuk mengubah gambar dari RGB ke HSV karena deteksi warna lebih stabil terhadap pencahayaan. Lalu masking untuk rentang warna biru didefinisikan dengan nilai HSV yaitu **mask = cv2.inRange(hsv, np.array([100, 150, 70]), np.array([140, 255, 255]))** dan membuat **mask** yang hanya mendeteksi piksel warna biru dengan nilai putih 255 dan warna lain menjadi hitam 0. Selanjutnya, menerapkan masking pada frame asli dengan **cv2.bitwise_and()** untuk menampilkan hanya bagian frame yang sesuai dengan warna biru, menghilangkan bagian

lain.. 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 objek warna biru, lalu **cv2.contourArea(cnt) > 500** untuk mendeteksi objek yang lebih besar dari 500 piksel, agar tidak menangkap noise kecil. **cv2.boundingRect(cnt)** berfungsi membuat kotak pembatas (**bounding box**) di sekitar objek biru yang terdeteksi. *Bounding box* ini digambar pada frame menggunakan **cv2.rectangle()** dengan nilai biru (255, 0, 0) dan **cv2.putText()** menambahkan teks "**Biru**" di atas kotak objek yang terdeteksi. Untuk nilai biru (255,0,0) dan tata letak tulisan dapat diubah ubah sesuai dengan keinginan. Hasil dari proses ditampilkan dalam 3 tab, dengan **cv2.imshow()**, yang pertama menampilkan *frame* untuk menampilkan gambar/foto/kamera asli dengan bounding box, *mask* untuk menampilkan hasil dari warna biru (hitam putih), dan *result* untuk menampilkan gambar asli namun hanya objek yang berwarna biru yang terlihat.. Adanya penambahan fitur *bounding box* ini meningkatkan kejelasan dan akurasi deteksi.

- Diskusi

Metode deteksi objek berwarna biru menggunakan OpenCV dengan bounding box memiliki kelebihan dan kekurangan. Kelebihannya adalah kemudahan dalam mendeteksi objek berdasarkan warna tertentu, yang membuat metode ini efisien dan sederhana untuk digunakan untuk deteksi real time. Penggunaan HSV, sistem menjadi lebih stabil karena HSV memungkinkan pemisahan warna lebih baik meskipun intensitas cahaya berubah dibandingkan dengan BGR. Selain itu, bounding box membantu dalam mengidentifikasi dan pelacakan objek, karena memberikan tanda visual yang jelas pada objek yang terdeteksi dalam sebuah kotak. Dimana pada percobaan kali ini menggunakan warna biru.

Namun, bounding box juga memiliki beberapa kelemahan diantaranya adalah kurang akurat dalam mendeteksi objek dengan bentuk kompleks. Selain itu, bounding box tidak dapat membedakan objek yang tumpang

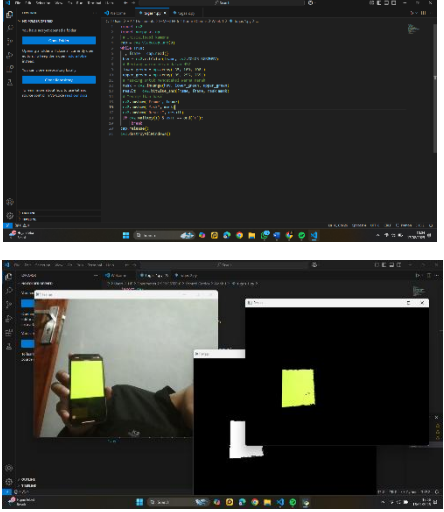
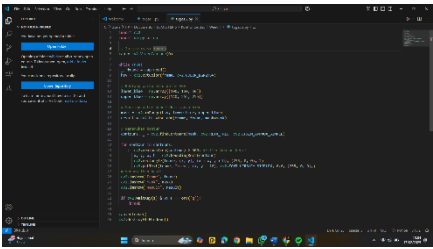
tindih, sehingga metode ini kurang efektif karena sulit digunakan dalam kondisi di mana beberapa objek berdekatan atau saling menutupi. Perubahan pencahayaan juga dapat menyebabkan kesalahan deteksi

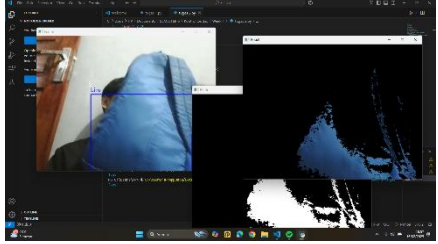
5. Assignment:

Dalam praktikum minggu ini bertujuan untuk mendeteksi objek warna menggunakan OpenCV dan dengan menggunakan bahasa pemrograman Python. Pada sistem ini dirancang untuk mendeteksi warna biru dengan bounding box. Cara kerja pada sistem ini dimulai dengan inisialisasi kamera, lalu mengonversinya dari RGB ke HSV, kemudian dilakukan masking dengan rentang warna biru untuk memisahkan objek warna biru dengan warna yang lain, Lalu sistem mendeteksi kontur objek pada hasil marking. Bounding box ditambahkan menggunakan **cv2.boundingRect()** dan label teks berwarna biru ditambahkan menggunakan **cv2.putText()** untuk memberikan informasi visual kepada pengguna bahwa objek telah terdeteksi. Dengan adanya *bounding box*, sistem menjadi informatif, karena pengguna dapat dengan jelas melihat objek yang dideteksi dalam kamera secara cepat dan mudah serta efektif dalam mendeteksi objek yang berbentuk teratur. Namun, bounding box kurang akurat untuk bentuk kompleks, objek bertumpuk, dan objek bergerak cepat.

Data dan Output Hasil Pengamatan:

No	Variabel	Hasil Pengamatan
1.	<ul style="list-style-type: none"> - Menginisialisasi kamera, yang kemudian dikonversi dari RGB ke warna HSV, dengan program yaitu <code>cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)</code> - Rentang warna hijau dalam HSV yaitu <code>lower_blue = np.array([35, 100, 100])</code> <code>upper_blue = np.array([85, 255, 255])</code> 	

	<ul style="list-style-type: none"> - Masking mendeteksi warna hijau yaitu <code>mask = cv2.inRange(hsv, lower_green, upper_green)</code> <code>result = cv2.bitwise_and(frame, frame, mask=mask)</code> - Untuk menampilkan hasil yaitu <code>cv2.imshow("Frame", frame)</code> <code>cv2.imshow("Mask", mask)</code> <code>cv2.imshow("Result", result)</code> - 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 dari RGB ke warna HSV, yaitu <code>cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)</code> - Rentang warna biru dalam HSV yaitu <code>mask = cv2.inRange(hsv, np.array([100, 150, 70]), np.array([140, 255, 255]))</code> - Untuk masking mendeteksi warna biru yaitu <code>result = cv2.bitwise_and(frame, frame, mask=mask)</code> - Menemukan kontur yaitu <code>cv2.findContours(mask, cv2.RETR_TREE,</code> 	

<pre>cv2.CHAIN_APPROX_SIMPLE)</pre> <ul style="list-style-type: none"> - Untuk memunculkan <i>bounding box</i> yaitu dengan <code>cv2.boundingRect(cnt)</code> - Untuk menambahkan teks "Biru" di atas kotak objek yang terdeteksi yaitu <code>cv2.putText</code> - 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> 	
---	--

6. Kesimpulan:

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

- Penambahan fitur bounding box membuat hasil deteksi lebih jelas dan memudahkan mengidentifikasi objek.
- Sistem bounding box memiliki kelemahan kurang akurat dalam mendeteksi objek dengan bentuk kompleks, tidak dapat membedakan objek yang tumpang tindih dan sensitive terhadap perubahan pencahayaan.

7. Saran:

Untuk meningkatkan akurasi yang tepat, terdapat beberapa hal yang dapat dilakukan yaitu pertama, menggunakan adaptive thresholding atau machine learning untuk deteksi lebih presisi dan menggunakan ukuran bounding box yang tepat, karena ukuran sangat mempengaruhi akurasi deteksi.

8. Daftar Pustaka

- Afandi , A. R. & Kurnia, H., 2023. Revolusi teknologi : Masa depan kecerdasan buatan (AI) dan dampaknya terhadap masyarakat. *Universitas Cokroaminoto Yogyakarta*.
- S. & A., 2023. Object detection and conversion of text to speech for visually impaired. *ADBU Journal of Engineering Technology*.
- Sutoyo, T. et al., 2009. Teori Pengolahan Citra Digital. *Andi Yogyakarta dan UDINUS Semarang*.
- Yusup, R. M., Yuliana, S. & Anugrah, A. F., 2024. PENDETEKSIAN OBJEK MENGGUNAKAN OPENCV DAN METODE YOLOv4-TINY UNTUK MEMBANTU TUNANETRA. *Journal of Computer Science and Information Technology (JCSIT)*.