

Laporan Praktikum Kontrol Cerdas

Nama : Reza Gusti Eka Prasetya
NIM : 224308044
Kelas : TKA 6B
Akun Github (Tautan) : <https://github.com/RezaGusti00>
Student Lab Assistant : Mas Ali

1. Judul Percobaan

Pendeteksi Warna Menggunakan OpenCV

2. Tujuan Percobaan

Dalam praktikum kontrol cerdas ini terdapat beberapa tujuan antara lain :

- Mempelajari cara menggunakan bahasa python
- Mengaplikasikan pengetahuan tentang OpenCV dalam mendeteksi suatu objek atau warna dengan algoritma kontrol cerdas untuk mengembangkan sistem yang dapat mengambil keputusan secara otomatis berdasarkan data visual.
- 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) adalah bidang ilmu komputer yang berfokus pada pengembangan sistem komputer yang mampu melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia. Dalam konteks sistem intelligent control, pengolahan citra memegang peranan penting dalam mendeteksi dan mengklasifikasi objek atau informasi visual dari lingkungan sekitar. Salah satu library yang umum digunakan dalam pengolahan citra adalah OpenCV (Open Source Computer Vision Library). OpenCV menyediakan berbagai fungsi dan algoritma untuk pengolahan citra, termasuk deteksi warna (Ahadi dkk., 2024).

Deteksi warna adalah metode untuk tracking objek atau klasifikasi benda dalam berbagai aplikasi, termasuk robotika. Proses ini melibatkan identifikasi warna primer dengan memanfaatkan *webcam* dan inisialisasi variabel HSV (Hue, Saturation, Value) untuk menentukan warna yang diinginkan (jurnal nusantara). Metode OpenCV menggunakan model warna HSV (Hue, Saturation, Value) menjadi pilihan utama karena kemampuannya dalam memisahkan informasi warna, saturasi, dan kecerahan. Hue Merepresentasikan jenis warna, seperti merah, kuning, hijau, biru, dan sebagainya. Nilai **Hue** biasanya dinyatakan dalam derajat (0-360) pada lingkaran warna. **Saturation**: Merepresentasikan tingkat kemurnian atau intensitas warna. Nilai Saturation berkisar antara 0 (abu-abu) hingga 1

(warna murni). Value Merepresentasikan kecerahan atau intensitas cahaya warna. Nilai Value berkisar antara 0 (hitam) hingga 1 (putih).

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).

Visual Studio Code (VSCode) adalah sebuah software yang digunakan untuk membuat kode sumber serbaguna yang sangat membantu dalam pengembangan Python. VSCode menawarkan fitur-fitur seperti Basic Editing, Debugging, Extension Marketplace, IntelliSense, Github Integration, Aksesibilitas, Live Share, Auto Save. Integrasinya dengan Github mempermudah pengelolaan version control langsung dari editor, memungkinkan pengembang untuk mengatur perubahan tanpa harus beralih aplikasi. Selain itu, dukungan untuk berbagai bahasa pemrograman dan ekstensi yang luas memungkinkan VSCode untuk disesuaikan dengan kebutuhan pengembangan Python secara spesifik. Terhubung dengan Github melalui ekstensi seperti "Tools Issues" dan "Github Pull Request", VSCode memungkinkan kolaborasi tim yang lebih lancar. Dengan langkah-langkah seperti membuat proyek, menulis dan menjalankan kode, mengelola paket, dan memanfaatkan Git, VSCode menjadi alat yang kuat untuk mengembangkan aplikasi Python dengan efisien dan kolaboratif.

4. Analisis dan Diskusi

A. Analisis

Percobaan pada praktikum yang dilakukan pada minggu pertama ini adalah membuat code menggunakan bahasa python pada software Visual Studio Code. Code yang dibuat bertujuan untuk mendeteksi warna atau objek menggunakan libery dari OpenCv secara real time. OpenCv dapat diakses dengan cara menambahkan kalimat perintah **import cv2 import numpy as np, cv2** digunakan untuk mengakses fungsi-fungsi OpenCV dalam pemrosesan citra dan video. **Numpy** digunakan untuk manipulasi array, yang sangat berguna saat bekerja dengan citra. Pada program terdapat keyword atau perintah **cap = cv2.VideoCapture(0)** yang berfungsi untuk menginialisasi kamera PC/Laptop. Kemudian pengelolaan frame kamera dengan **cap.read()[1]** Mengambil frame dari kamera. Indeks [1] digunakan karena cap.read() mengembalikan tuple (retval, frame). **cv2.flip(frame, 1)** digunakan untuk membalik tampilan kamera secara horizontal sehingga tampilan menjadi mirror (simetris terhadap sumbu vertical). Dalam code menggunakan warna menjadi objek yang diamati, dengan cara menggunakan perintah **hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)** hsv digunakan untuk mengubah ruang warna BGR (format default OpenCV) ke HSV. Ruang warna HSV memudahkan pendeteksian warna karena lebih sesuai dengan persepsi manusia dalam memisahkan warna.

lower_red = np.array([0, 120, 70]) upper_red = np.array([10, 255, 255])
lower_yellow = np.array([20, 100, 100]) upper_yellow = np.array([30, 255, 255])

```
lower_blue = np.array([100, 150, 0]) upper_blue = np.array([140, 255, 255])
lower_green = np.array([40, 40, 40]) upper_green = np.array([70, 255, 255])
```

Code diatas menunjukkan setiap pasang array mendefinisikan batas bawah dan atas dalam ruang HSV untuk mendeteksi warna merah, kuning, biru, dan hijau. Rentang ini menentukan nilai hue, saturation, dan value yang dianggap sebagai warna target.

```
mask_red = cv2.inRange(hsv, lower_red, upper_red) mask_yellow =
cv2.inRange(hsv, lower_yellow, upper_yellow) mask_blue = cv2.inRange(hsv,
lower_blue, upper_blue) mask_green = cv2.inRange(hsv, lower_green,
upper_green)
```

Fungsi cv2.inRange menghasilkan citra biner (mask) dimana pixel dengan nilai dalam rentang yang ditentukan akan bernilai putih (255) dan yang lain hitam (0).

B. Diskusi

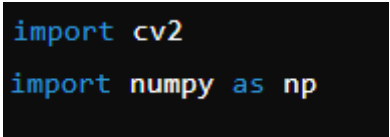
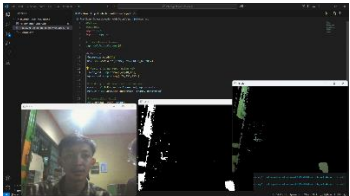
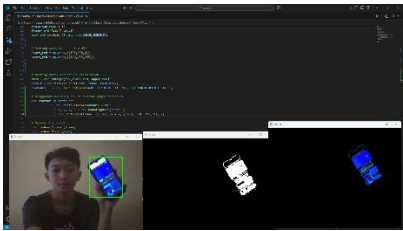
Metode deteksi objek berwarna biru dengan OpenCV dan fitur ***bounding box*** memiliki kelebihan dan kekurangan. Salah satu kelebihan adalah kemudahan dalam mendeteksi objek berdasarkan warna tertentu. Selain itu, penggunaan **HSV** membuat sistem lebih stabil terhadap perubahan pencahayaan dibandingkan dengan **BGR**. Selain itu, ***bounding box*** memungkinkan objek yang terdeteksi ditandai dengan jelas, sehingga memudahkan proses identifikasi dan verifikasi dalam suatu *frame*. Namun, metode ini memiliki keterbatasan, seperti **kesalahan deteksi akibat perubahan pencahayaan atau keberadaan objek lain dengan warna serupa**. Selain itu, metode berbasis warna saja tidak mampu membedakan objek berdasarkan bentuk, sehingga dapat mengurangi tingkat keakuratan sistem.

5. Assignment

Pada praktikum kali ini dengan percobaan mendeteksi objek sederhana berbasis warna menggunakan OpenCV. Selain itu, menggunakan beberapa *software* lain seperti *python* dan *Visual Studi Code*. Sistem ini dirancang untuk mendeteksi warna khususnya warna biru dengan tambahan fitur ***bounding box***. Pertama- tama kita 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” 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. Akan tetapi, terdapat beberapa kekurangan seperti kemungkinan deteksi objek yang tidak diinginkan, seperti latar belakang atau pencahayaan memiliki warna yang serupa.

6. Data dan Output Hasil Pengamatan

Sajikan data dan hasil yang diperoleh selama percobaan. Gunakan tabel untuk menyajikan data jika diperlukan.

No	Variabel	Hasil Pengamatan
1	Import library import cv2 import numpy as np	 <pre>import cv2 import numpy as np</pre> <ul style="list-style-type: none">• cv2: Digunakan untuk mengakses fungsi-fungsi OpenCV dalam pemrosesan citra dan video.• numpy: Digunakan untuk manipulasi array, yang sangat berguna saat bekerja dengan citra.
2	Inisialisasi Kamera cap = cv2.VideoCapture(0)	 <p>Membuka akses ke kamera (biasanya kamera default komputer). Angka 0 menunjukkan kamera pertama yang terdeteksi.</p>
3	Loop Utama untuk Pengolahan Frame while True: frame = cap.read()[1] frame = cv2.flip(frame, 1)	 <p>cv2.flip(frame, 1): Membalik frame secara horizontal sehingga tampilan menjadi mirror (simetris terhadap sumbu vertikal).</p>

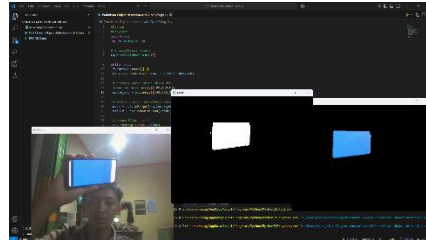
4.

Konversi ke Ruang Warna HSV

```
hsv = cv2.cvtColor(frame,
cv2.COLOR_BGR2HSV)
```

Name	City	State	Year	Score	Rank
Allyssa	Albany	NY	1988/89	129	128
Allyssa	Albany	NY	1989/90	129	128
Allyssa	Albany	NY	1990/91	129	128
Allyssa	Albany	NY	1991/92	129	128
Allyssa	Albany	NY	1992/93	129	128
Allyssa	Albany	NY	1993/94	129	128
Allyssa	Albany	NY	1994/95	129	128
Allyssa	Albany	NY	1995/96	129	128
Allyssa	Albany	NY	1996/97	129	128
Allyssa	Albany	NY	1997/98	129	128
Allyssa	Albany	NY	1998/99	129	128
Allyssa	Albany	NY	1999/00	129	128
Allyssa	Albany	NY	2000/01	129	128
Allyssa	Albany	NY	2001/02	129	128
Allyssa	Albany	NY	2002/03	129	128
Allyssa	Albany	NY	2003/04	129	128
Allyssa	Albany	NY	2004/05	129	128
Allyssa	Albany	NY	2005/06	129	128
Allyssa	Albany	NY	2006/07	129	128
Allyssa	Albany	NY	2007/08	129	128
Allyssa	Albany	NY	2008/09	129	128
Allyssa	Albany	NY	2009/10	129	128
Allyssa	Albany	NY	2010/11	129	128
Allyssa	Albany	NY	2011/12	129	128
Allyssa	Albany	NY	2012/13	129	128
Allyssa	Albany	NY	2013/14	129	128
Allyssa	Albany	NY	2014/15	129	128
Allyssa	Albany	NY	2015/16	129	128
Allyssa	Albany	NY	2016/17	129	128
Allyssa	Albany	NY	2017/18	129	128
Allyssa	Albany	NY	2018/19	129	128
Allyssa	Albany	NY	2019/20	129	128
Allyssa	Albany	NY	2020/21	129	128
Allyssa	Albany	NY	2021/22	129	128
Allyssa	Albany	NY	2022/23	129	128
Allyssa	Albany	NY	2023/24	129	128
Allyssa	Albany	NY	2024/25	129	128
Allyssa	Albany	NY	2025/26	129	128
Allyssa	Albany	NY	2026/27	129	128
Allyssa	Albany	NY	2027/28	129	128
Allyssa	Albany	NY	2028/29	129	128
Allyssa	Albany	NY	2029/30	129	128
Allyssa	Albany	NY	2030/31	129	128
Allyssa	Albany	NY	2031/32	129	128
Allyssa	Albany	NY	2032/33	129	128
Allyssa	Albany	NY	2033/34	129	128
Allyssa	Albany	NY	2034/35	129	128
Allyssa	Albany	NY	2035/36	129	128
Allyssa	Albany	NY	2036/37	129	128
Allyssa	Albany	NY	2037/38	129	128
Allyssa	Albany	NY	2038/39	129	128
Allyssa	Albany	NY	2039/40	129	128
Allyssa	Albany	NY	2040/41	129	128
Allyssa	Albany	NY	2041/42	129	128
Allyssa	Albany	NY	2042/43	129	128
Allyssa	Albany	NY	2043/44	129	128
Allyssa	Albany	NY	2044/45	129	128
Allyssa	Albany	NY	2045/46	129	128
Allyssa	Albany	NY	2046/47	129	128
Allyssa	Albany	NY	2047/48	129	128
Allyssa	Albany	NY	2048/49	129	128
Allyssa	Albany	NY	2049/50	129	128
Allyssa	Albany	NY	2050/51	129	128
Allyssa	Albany	NY	2051/52	129	128
Allyssa	Albany	NY	2052/53	129	128
Allyssa	Albany	NY	2053/54	129	128
Allyssa	Albany	NY	2054/55	129	128
Allyssa	Albany	NY	2055/56	129	128
Allyssa	Albany	NY	2056/57	129	128
Allyssa	Albany	NY	2057/58	129	128
Allyssa	Albany	NY	2058/59	129	128
Allyssa	Albany	NY	2059/60	129	128
Ally					

Warna web

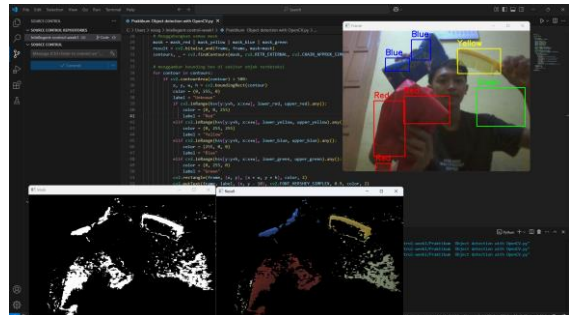


Mengubah ruang warna dari BGR (format default OpenCV) ke HSV. Ruang warna HSV memudahkan pendeteksian warna karena lebih sesuai dengan persepsi manusia dalam memisahkan warna.

5.

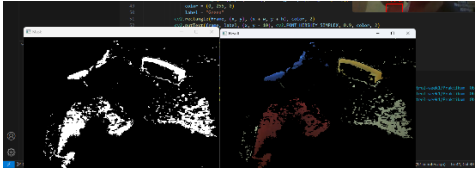
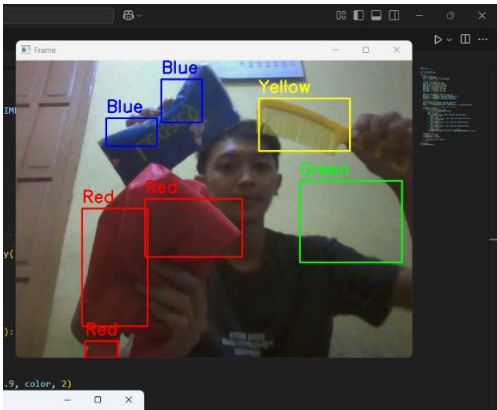
Pendefinisian Rentang Warna

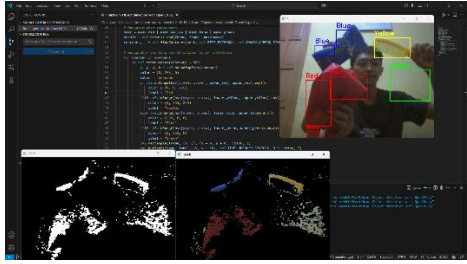
```
lower_red = np.array([0,
120, 70]) upper_red =
np.array([10, 255, 255])
lower_yellow =
np.array([20, 100, 100])
upper_yellow =
np.array([30, 255, 255])
lower_blue =
np.array([100, 150, 0])
upper_blue =
np.array([140, 255, 255])
lower_green =
np.array([40, 40, 40])
upper_green =
np.array([70, 255, 255])
```



Setiap pasang array mendefinisikan batas bawah dan atas dalam ruang HSV untuk mendeteksi warna merah, kuning, biru, dan hijau.

Rentang ini menentukan nilai hue, saturation, dan value yang dianggap sebagai warna target.

<p>6.</p>	<p>Pembuatan Mask untuk Setiap Warna</p> <pre>mask_red = cv2.inRange(hsv, lower_red, upper_red) mask_yellow = cv2.inRange(hsv, lower_yellow, upper_yellow) mask_blue = cv2.inRange(hsv, lower_blue, upper_blue) mask_green = cv2.inRange(hsv, lower_green, upper_green)</pre>	 <p>Fungsi cv2.inRange menghasilkan citra biner (mask) dimana pixel dengan nilai dalam rentang yang ditentukan akan bernilai putih (255) dan yang lain hitam (0).</p>
<p>7.</p>	<p>Penggambaran Bounding Box dan Label pada Objek Terdeteksi</p> <pre>for contour in contours: if cv2.contourArea(contour) > 500: x, y, w, h = cv2.boundingRect(contour) color = (0, 255, 0) label = "Unknown"</pre>	 <p>Filter Kontur: Hanya kontur dengan area lebih dari 500 yang diproses, sehingga menghindari deteksi noise kecil.</p> <p>Bounding Rectangle: Fungsi cv2.boundingRect menghitung koordinat dan ukuran persegi panjang yang membungkus kontur.</p>

8.	Menampilkan Hasil <code>cv2.imshow("Frame", frame) cv2.imshow("Mask", mask) cv2.imshow("Result", result)</code>	 <p>Menampilkan citra hasil bitwise yang hanya menampilkan area dengan warna target.</p>
9.	Penghentian Proram <code>if cv2.waitKey(1) & 0xFF == ord('q'): break</code>	Digunakan untuk menghentikan program

7. Kesimpulan

Kode ini mengintegrasikan OpenCV dan NumPy untuk menangkap video secara real-time, mendeteksi beberapa warna (merah, kuning, biru, hijau) menggunakan rentang warna di ruang HSV, dan menggambar bounding box serta label di sekitar objek yang terdeteksi. Pendekatan ini dapat digunakan dalam berbagai aplikasi seperti pelacakan objek, robotika, atau sistem pengawasan berbasis warna.

8. Saran

Untuk meningkatkan akurasi secara optimal, beberapa langkah dapat diterapkan. Pertama, metode morfologi dapat digunakan untuk mengurangi noise pada hasil deteksi. Kedua, teknik Convolutional Neural Networks (CNN) dapat dimanfaatkan untuk memungkinkan identifikasi objek selain berdasarkan warna.

10. Daftar Pustaka

- Alam, S., Zainal, M., & Fazil, E. (2024). PERANCANGAN SISTEM PENGENALAN WAJAH MENGGUNAKAN PYTHON, OPENCV DAN HAARCASCADE. 9.
- Ahadi, A.H., Gustina, G., Syawal, M.F., Aminuddin, F.H., Anzari, Y., 2024. IMPLEMENTASI SISTEM PENDETEKSI WARNA OBJEK DENGAN OPENCV-PYTHON. SENTRI J. Ris. Ilm. 3, 3573–3578.
<https://doi.org/10.55681/sentri.v3i7.3185>