

LAPORAN PROJECT UTS KECERDASAN BUATAN

Disusun Guna Memenuhi Ujian Tengah Semester Mata Kuliah Kecerdasan Buatan

Dosen Pengampu : Prof. Dr. Ir. Siti Nurmaini, M.T.



Disusun Oleh :

Nama : Ciyin Asti Safitri

NIM : 09011182025017

Kelas : SK5 A Regular

**PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
INDRALAYA
2022**

I. JUDUL PROJECT AI

Multiple Color Detection in Real-Time using Image Processing

II. PENDAHULUAN

Warna merupakan hasil dari persepsi mata manusia dalam melihat suatu objek (psycho-physiological).

Berdasarkan hasil penelitian Sir Isaac Newton (1666), yaitu saat ia menemukan beberapa warna dari hasil pembiasan cahaya putih yang berasal dari cahaya matahari pada prisma kaca, ia menyimpulkan bahwa warna dapat diartikan sebagai refleksi cahaya yang alami dari objek.

Deteksi warna adalah proses mendeteksi nama warna apa pun. Sederhana bukan? Nah, bagi manusia ini adalah tugas yang sangat mudah tetapi untuk komputer, itu tidak mudah. Mata dan otak manusia bekerja sama untuk menerjemahkan cahaya menjadi warna. Reseptor cahaya yang ada di mata kita mengirimkan sinyal ke otak. Otak kita kemudian mengenali warna tersebut. Sejak kecil, kita telah memetakan lampu tertentu dengan nama warnanya. Project ini akan menggunakan strategi yang agak sama untuk mendeteksi nama warna.

Warna sendiri terdiri dari 3 warna primer; merah, hijau, dan biru. Di komputer, kita mendefinisikan setiap nilai warna dalam rentang 0 hingga 255. Namun, pada project ini saya hanya fokus mendeteksi beberapa warna diantaranya Merah, Biru, Kuning, dan Hijau.

Dalam project ini, saya membuat aplikasi di mana dengan program Python ini, kita dapat mengenali warna tertentu pada objek yang tertangkap pada kamera/video secara realtime yang nantinya akan menampilkan label berupa bingkai/contour beserta nama warna pada objek tersebut.

III. DASAR TEORI

• Computer Vision

Computer Vision merupakan salah satu bagian dari Artificial Intelligence. Computer Vision adalah salah satu teknologi atau bidang ilmu komputer yang berfokus pada pembuatan sistem digital yang dapat memproses, menganalisis, dan memahami data visual (gambar atau video) dengan cara yang sama seperti yang dilakukan manusia.

Computer vision akan memecah gambar yang ada menjadi banyak bagian-bagian yang berbeda. Jaringan neuron akan mengidentifikasi setiap bagian-bagian yang ada, barulah kemudian menggabungkannya menjadi satu bagian yang utuh. Secara otomatis komputer akan melabeli gambar tersebut. Itulah mengapa semakin banyak data yang dijadikan bahan pembelajaran oleh komputer, kemampuan untuk memprediksi gambar tersebut secara benar juga akan meningkat. Komputer akan memanfaatkan algoritma yang mampu untuk menganalisis warna, bentuk, jarak antar objek, dan seterusnya hingga komputer mampu melabeli seluruh objek sebagai benda tertentu.

Meskipun objek yang dimasukkan merupakan objek tiga dimensi, namun komputer tetap akan mendefinisikannya sebagai objek dua dimensi karena keterbatasan kemampuannya dalam mengingat garis-garis linear dalam satu bidang. Gambar dua dimensi ini kemudian akan diterjemahkan sebagai angka, begitu dijumlahkan akan membentuk nilai tertentu yang merepresentasikan kode warna. Setiap piksel akan mengandung tiga warna utama yaitu merah, hijau, dan biru. Identifikasi warna menjadi hal yang cukup penting untuk memberikan penjelasan mendetail.

• Pengolahan Citra (Image Processing)

Pengolahan citra adalah metode yang digunakan untuk memproses/mengubah citra menjadi bentuk digital, dimana dapat juga dikatakan untuk memperbaiki kualitas, mengambil dan mengubah informasi pada suatu citra. Pengolahan citra telah banyak dikembangkan peneliti dalam pendeteksian objek. Penelitian tersebut menunjukkan bahwa warna dapat dijadikan sebagai nilai acuan untuk melakukan pendeteksian objek melalui kamera secara langsung.

Konsep dasar pemrosesan suatu objek pada gambar menggunakan pengolahan citra diambil dari kemampuan indera penglihatan manusia yang selanjutnya dihubungkan dengan kemampuan otak manusia.

Pengolahan citra adalah jenis dispensasi sinyal di mana input-nya adalah gambar, seperti frame (bingkai) video atau foto dan output-nya dapat berupa gambar atau karakteristik yang terkait dengan gambar tersebut.

Biasanya system atau sistem pemrosesan gambarnya termasuk seperti memperlakukan gambar sebagai sinyal 2 (dua) dimensi sambil menerapkan metode signal processing (pemrosesan sinyal) yang sudah ditetapkan.

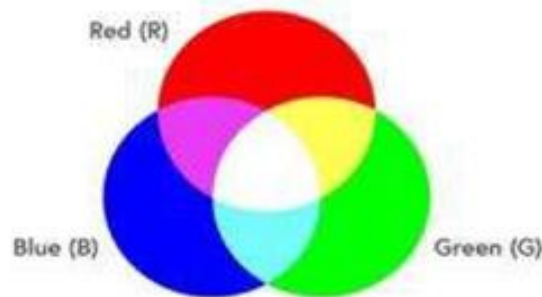
- **Object Detection (Recognition)**

Object detection atau recognition adalah salah satu sub bagian dan teknik visi komputer yang memungkinkan kita untuk mengidentifikasi dan menemukan objek dalam gambar atau video. Dengan jenis identifikasi dan lokalisasi ini, deteksi objek dapat digunakan untuk menghitung objek dalam pemandangan dan menentukan serta melacak lokasi akuratnya, ataupun mendeteksi warna pada objek, sambil memberi label secara akurat.

Bayangkan saja, sebagai contoh misalnya, gambar yang berisi 2 (dua) kucing dan 1 (satu) orang, dengan deteksi objek, maka itu memungkinkan juga kita untuk sekaligus mengklasifikasikan jenis hal yang ditemukan sambil juga menemukan contoh mereka di dalam gambar.

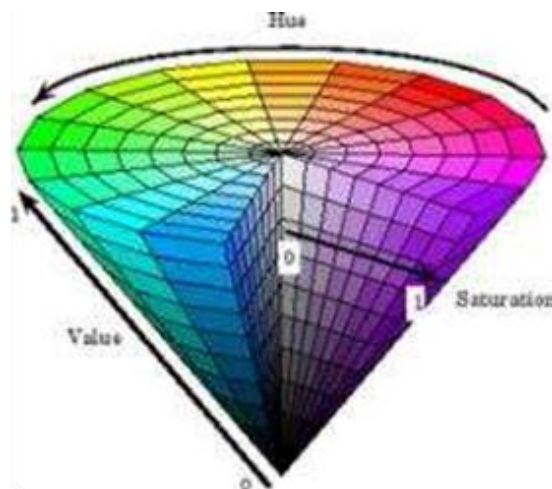
- **RGB**

Setiap warna yang ada pada citra/gambar adalah kombinasi dari warna RGB (Red, Green, Blue). Nilai pembacaan warna dari 0%-100% yaitu 0-255. Jika RGB diberikan nilai 100% pada setiap warna primernya misalnya RGB (255,255,255) maka warna yang akan dihasilkan adalah putih. Dan ketika dua warna diberikan nilai 100% satu warna dengan nilai 0%, misal RGB (255,0,255), (255,255,0), (0,255,255), maka terciptalah warna sekunder yang biasa disebut dengan CMY (Cian, Magenta, Yellow).



- **HSV (Hue, Saturation, Value)**

Model HSV ini membutuhkan warna dasar RGB sebagai acuan dalam pendeteksian warna. Hasil penelitian Irfan (2020) menjelaskan bahwa ruang warna pada HSV merupakan transformasi dari kubus warna RGB menjadi kerucut.

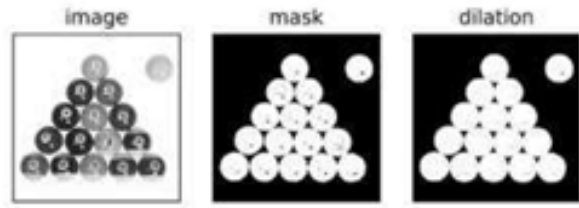


- H (Hue) merupakan sebuah sudut warna sumbu lingkaran pada kerucut, warna merah sebagai sumbu 0° .
- V (Value) merupakan bagian warna dari sumbu vertikal pada kerucut. Nilai $V = 0$ terletak pada bagian akhir sumbu yang berwarna hitam dan nilai $V=1$ pada bagian ujung sumbu berwarna putih. Sumbu V ini merupakan presentasi semua jenis warna keabuan.
- Dan S (Saturation) adalah tingkat saturasi yang mengandung banyak cahaya putih atau kemurnian warna dengan nilai yang berada pada radian dari bentuk sebuah bangun kerucut.

- **Morfologi**

Morfologi adalah teknik pengolahan citra yang bertujuan untuk memperbaiki hasil segmentasi citra tersebut. Transformasi morfologi ini biasanya di bagi dua, yaitu dilasi dan erosi. Namun, pada pendeteksi warna hanya menggunakan transformasi morfologi dilasi.

- ✚ **Dilasi**



Dilasi adalah teknik untuk memfilter noise yang berwarna hitam (0) menjadi warna putih (1) dan memperbesar segmen pada objek citra.

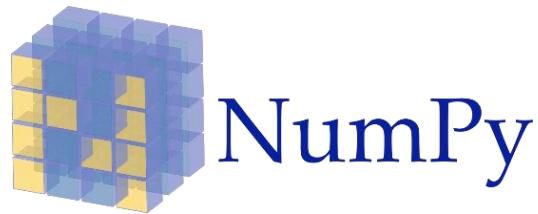
- **OpenCV Python**

OpenCV (Open Source Computer Vision Library) adalah sebuah library yang digunakan untuk mengolah gambar dan video sehingga pengguna dapat meng-ekstrak informasi dari sebuah citra. OpenCV dikembangkan oleh perusahaan Intel yang berfokus untuk menyederhanakan programing yang terkait dengan citra. OpenCV dapat berjalan dengan berbagai bahasa pemrograman, seperti C, C++, Java, Python, dan juga support diberbagai platform seperti Windows, Linux, Mac OS, iOS dan Android. Di dalam opencv terdapat banyak fitur sebagai berikut.



Python merupakan program yang dibuat oleh programmer Belanda bernama Guido Van Rossum. Python tergolong bahasa pemrograman yang berlevel tinggi, namun python dirancang sedemikian rupa sehingga mudah dipelajari dan dipahami. Kelebihan program python ini yaitu mudah untuk di pelajari, dapat menjalankan program dengan banyak fungsi kompleks didalamnya dengan mudah, lebih sedikit menggunakan kode, dan bahkan mampu mengubah program dengan tingkat kerumitang tinggi menjadi mudah. Namun demikian python juga memiliki kekurangan yaitu cukup lambat dijalankan, kurang support terhadap android dan IOS, memiliki keterbatasan dalam akses basis data, dan juga tidak cocok untuk melakukan pekerjaan multi-core/multi- processor.

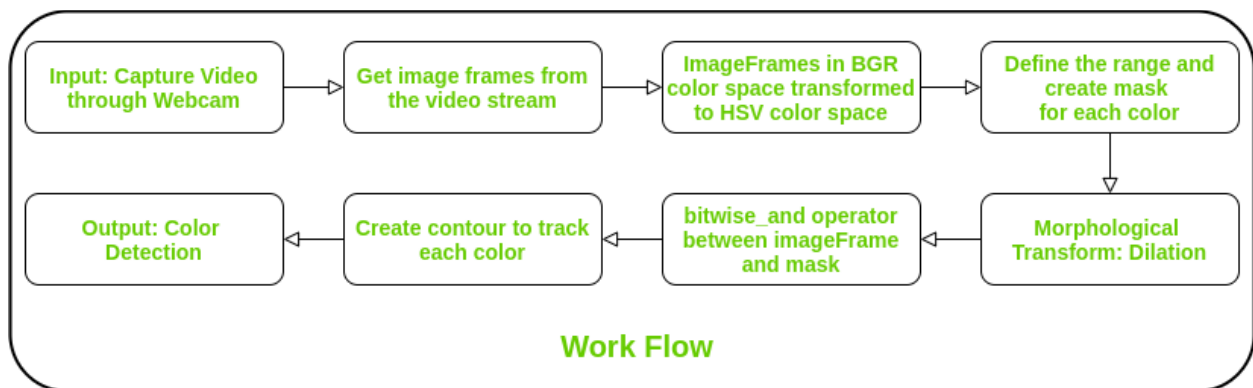
- NumPy Python



NumPy merupakan singkatan dari Numerical Python. NumPy merupakan salah satu library Python yang berfungsi untuk proses komputasi numerik. NumPy memiliki kemampuan untuk membuat objek N-dimensi array. Array merupakan sekumpulan variabel yang memiliki tipe data yang sama. Kelebihan dari NumPy Array adalah dapat memudahkan operasi komputasi pada data, cocok untuk melakukan akses secara acak, dan elemen array merupakan sebuah nilai yang independen sehingga penyimpanannya dianggap sangat efisien.

IV. METODE

Pada project ini, saya menggunakan metode pengolahan citra deteksi objek berdasarkan warna dalam melakukan pendeteksian warna pada objek. Pengolahan citra ini di jalankan menggunakan program Python, library OpenCV dan library NumPy sehingga saya dapat meng-ekstrak informasi dari sebuah citra.



Work flow diatas detailnya dapat dijelaskan sebagai berikut.

1. Input: Rekam video melalui webcam secara realtime

```
4 #Webcam video capture
5 cap = cv2.VideoCapture(0)
```

Pertama-tama ialah dengan membuka dan membaca kamera atau webcam menggunakan program Python. Kamera yang terbuka nantinya akan menangkap kondisi sekitar secara realtime layaknya merekam video.

2. Mendapatkan frame gambar dari video

```
8 #Reading the video
9 #Webcam in image frames
10 _, img = cap.read()
```

Setelah video dari kamera dibaca, selanjutnya ialah membaca video tersebut dan diubah kedalam bentuk frame-per-frame yang mana, frame ini akan digunakan untuk membaca gambarnya dan warna yang akan dideteksi.

3. Konversi RGB ke HSV

```
12 #Converting frame(img == BGR) to HSV(hue-saturation-value)
13 hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
```

Setelah itu, program melakukan konversi frame dari RGB ke HSV. Hal ini dilakukan karena ruang warna pada HSV lebih luas dan mendekati penglihatan mata manusia dibandingkan dengan model RGB, karena

warna yang dibentuk model lain seperti RGB merupakan hasil campuran dari warna-warna primer. Sehingga program dapat dengan optimal membaca gambar dengan warna yang mendekati warna yang dapat dilihat oleh manusia. Selain itu, konversi ini juga dapat digunakan untuk mengurangi image noise pada saat capture.

4. Menentukan range atau jangkauan dan membuat mask untuk setiap warna yang dideteksi

```
15     #Define range of red color in HSV
16     red_lower = np.array([136, 87, 111], np.uint8)
17     red_upper = np.array([180, 255, 255], np.uint8)
18
19     #Define range of blue color in HSV
20     blue_lower = np.array([99, 115, 150], np.uint8)
21     blue_upper = np.array([110, 255, 255], np.uint8)
22
23     #Define range of yellow color in HSV
24     yellow_lower = np.array([22, 60, 200], np.uint8)
25     yellow_upper = np.array([60, 255, 255], np.uint8)
26
27     #Define range of green color in HSV
28     green_lower = np.array([25, 52, 72], np.uint8)
29     green_upper = np.array([102, 255, 255], np.uint8)
30
31     #Finding the range of red,blue,yellow,green colors in the image
32     #Define mask
33     red = cv2.inRange(hsv, red_lower, red_upper)
34     blue = cv2.inRange(hsv, blue_lower, blue_upper)
35     yellow = cv2.inRange(hsv, yellow_lower, yellow_upper)
36     green = cv2.inRange(hsv, green_lower, green_upper)
```

Pada proses ini, terdapat pendeklarasian range warna yang akan dideteksi dalam bentuk array. Hal ini berguna untuk mencocokkan tingkat deteksi yang akan diperoleh, semakin besar range warnanya, semakin besar kemungkinan warna yang akan terdeteksi. Karena, program akan mencocokkan warna 3 layer tersebut dengan frame yang terbaca secara real-time.

5. Proses morfologi: dilasi

```
38     #Morphological transformation, Dilation
39     kernal = np.ones((5,5), "uint8")
40
41     red = cv2.dilate(red, kernal)
42     res = cv2.bitwise_and(img, img, mask=red)
43
44     blue = cv2.dilate(blue, kernal)
45     res1 = cv2.bitwise_and(img, img, mask=blue)
46
47     yellow = cv2.dilate(yellow, kernal)
48     res2 = cv2.bitwise_and(img, img, mask=yellow)
49
50     green = cv2.dilate(green, kernal)
51     res3 = cv2.bitwise_and(img, img, mask=green)
```

Proses ini dilakukan untuk memperbaiki hasil segmentasi citra yang di tangkapnya. Jika hasil filterisasi HSV dapat memfilter seluruh noise pada citra, maka dilasi tidak perlu lagi dilakukan. Namun sebaliknya, jika

setelah melakukan filter HSV pada citra dan masih menyisakan noise, maka akan di bersihkan oleh metode morfologi dilasi.

6. Proses operator bitwise_and antara gambar dan mask

Proses mendeteksi warna tertentu menggunakan operator “bitwise_and” yang digunakan untuk membalik gambar dan mask sehingga objek tertentu saja yang akan terlihat/muncul. Ini berguna untuk mempermudah komputer mencocokkan warna yang muncul dengan hanya mendeteksi warna tertentu saja.

7. Membuat kontur untuk melacak setiap warna yang terdeteksi

```
53     #Creating contour to track red color
54     (contours, hierarchy) = cv2.findContours(
55         red, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
56
57     for pic, contour in enumerate(contours):
58         area = cv2.contourArea(contour)
59         if(area > 300):
60             x, y, w, h = cv2.boundingRect(contour)
61             img = cv2.rectangle(img, (x, y), (x+w, y+h), (0, 0, 255), 2)
62             cv2.putText(img, "Merah", (x, y),
63                 cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255))
64
65     #Creating contour to track blue color
66     (contours, hierarchy) = cv2.findContours(
67         blue, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
68
69     for pic, contour in enumerate(contours):
70         area = cv2.contourArea(contour)
71         if(area > 300):
72             x, y, w, h = cv2.boundingRect(contour)
73             img = cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
74             cv2.putText(img, "Biru", (x, y),
75                 cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0))
76
77     #Creating contour to track yellow color
78     (contours, hierarchy) = cv2.findContours(
79         yellow, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
80
81     for pic, contour in enumerate(contours):
82         area = cv2.contourArea(contour)
83         if(area > 300):
84             x, y, w, h = cv2.boundingRect(contour)
85             img = cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 255), 2)
86             cv2.putText(img, "Kuning", (x, y),
87                 cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 255, 255))
88
89     #Creating contour to track green color
90     (contours, hierarchy) = cv2.findContours(
91         green, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
92
93     for pic, contour in enumerate(contours):
94         area = cv2.contourArea(contour)
95         if(area > 300):
96             x, y, w, h = cv2.boundingRect(contour)
97             img = cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
98             cv2.putText(img, "Hijau", (x, y),
99                 cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 255, 0))
```

Proses ini dilakukan jika terdapat warna yang terdeteksi atau sama dengan suatu warna pada range tertentu. Program akan mendeteksi dan membuat bingkai/kontur yang digunakan sebagai penanda bahwa terdapat warna yang sama dengan warna yang dideklarasikan pada range HSV.

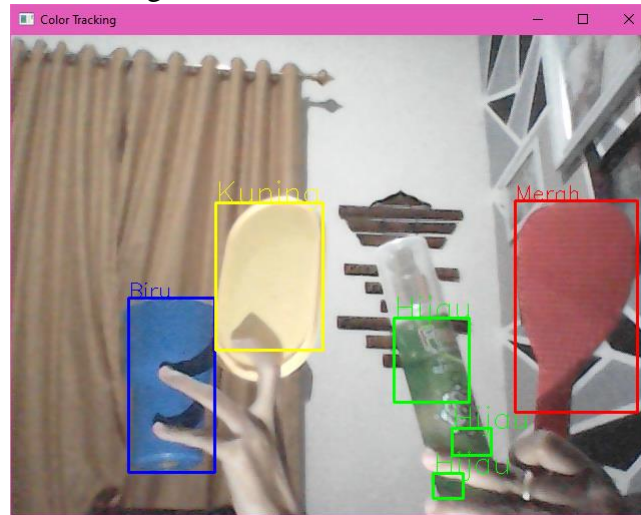
8. Hasil output: Penampilan data

```
101     #Program Termination
102     cv2.imshow("Color Tracking", img)
103     if cv2.waitKey(10) & 0xFF == ord('q'):
104         cap.release()
105         cv2.destroyAllWindows()
106         break
```

Setelah semua proses selesai maka monitor program akan menampilkan data hasil pengolahan citra yang berupa pembacaan dan pengolahan deteksi warna secara langsung menggunakan webcam laptop.

V. HASIL DAN PEMBAHASAN

Hasil output dari project ini ialah sebagai berikut.



Pada project ini, metode yang di gunakan untuk melakukan deteksi warna yaitu metode pengolahan citra deteksi objek berdasarkan warna. Proses mendeteksi warna ini dimulai dari penangkapan warna pada objek secara real-time, mengkonversi warna RGB ke HSV, dilanjutkan dengan melakukan threshold dengan mendeklarasikan range dari warna yang akan dideteksi, setelah itu proses morfologi untuk memfilter noise yang tidak dibutuhkan pada citra, kemudian melakukan pencocokan gambar dengan range warna yang dideklarasikan untuk menentukan kesamaan warnanya dengan kontur sebagai penanda jika terdapat warna yang sama. Hasil project ini untuk dapat mengenali warna secara real-time dimana objek yang ditangkap akan menghasilkan bingkai/kontur dengan label nama warna sesuai dengan warna pada objek tersebut. Untuk warna yang bisa dideteksi pada project saya ini diantaranya merah, biru, kuning, dan hijau.

VI. KESIMPULAN

Berdasarkan hasil dari project yang telah saya lakukan, dapat diperoleh beberapa kesimpulan sebagai berikut.

1. Deteksi warna adalah proses mendeteksi nama warna apa pun. Bagi manusia ini adalah tugas yang sangat mudah tetapi untuk komputer, itu tidak mudah. Mata dan otak manusia bekerja sama untuk menerjemahkan cahaya menjadi warna.
2. Pengolahan citra dapat bekerja dengan baik dalam mendeteksi warna.
3. Program Python yang telah di buat dapat berfungsi dengan baik dan dapat bekerja sesuai yang di rancang.

VII.SUMBER REFERENSI

<https://www.geeksforgeeks.org/multiple-color-detection-in-real-time-using-python-opencv/>