

Implementasi Deteksi Warna Menggunakan Phyton dan OpenCV dengan ESP32CAM OV2640

1st Wisnu Al Rasyid
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

wisnuar@student.telkomuniversity.ac.id

2nd Sony Sumaryo
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

sonysumaryo@telkomunivesity.ac.id

3rd Erwin Susanto
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

erwinsusanto@telkomuniversity.ac.id

Abstract – Tujuan utama dari aplikasi ini adalah untuk mengidentifikasi warna dengan prediksi yang tepat dengan namanya. Warna Primer RGB memiliki nilai 0- 255 yang bila dikombinasikan dapat menghasilkan 16.581.375 warna. Sebuah penelitian mengatakan, manusia normal dapat dengan jelas mengidentifikasi hampir 1 juta corak warna. Namun dalam kasus tertentu ada yang dapat melihat 1% (yaitu 10.000 warna) dari manusia normal.

Kata kunci –, RGB value, OpenCv, pandas

I. PENDAHULUAN

Sebelum membahas spekulasi proyek ini, penting untuk mengetahui definisi pendeteksian warna. Secara sederhana, ini adalah proses mengidentifikasi nama warna apa pun. Sudah jelas bahwa manusia melakukan tindakan ini secara alami dan tidak berusaha keras untuk melakukannya. Sementara itu, tidak demikian halnya dengan komputer.

Mata dan otak manusia bekerja secara terkoordinasi untuk menerjemahkan cahaya menjadi warna. Reseptor cahaya yang ada di mata mengirimkan sinyal ke otak yang pada gilirannya mengenali warna. Tidak berlebihan jika dikatakan bahwa manusia telah memetakan cahaya tertentu dengan nama warnanya sejak kecil. Strategi yang sama berguna dalam mendeteksi nama warna dalam proyek ini.

Tiga warna yang berbeda, Merah, Hijau dan Biru dilacak dengan memanfaatkan dasar-dasar visi komputer. Setelah kompilasi berhasil, ketika kita menjalankan kode, sebuah jendela akan mengalihkan ke gambar yang ditampilkan yang jalurnya diberikan sebagai argumen.

Selain itu, kami memperoleh nama warna piksel bersama dengan komposisi tiga warna yang berbeda yaitu merah, biru dan hijau. Ini sangat membantu dalam mengenali warna dan dalam robotika. Salah satu aplikasi deteksi warna dengan visi komputer adalah pada mobil tanpa pengemudi. Sistem ini berguna untuk mendeteksi lalu lintas dan lampu latar kendaraan dan mengambil keputusan untuk berhenti, memulai, dan melanjutkan perjalanan. Hal ini juga memiliki banyak aplikasi di industri untuk memilih dan menempatkan objek berwarna berbeda oleh lengan robot. Deteksi warna juga digunakan sebagai alat dalam berbagai aplikasi pengeditan gambar dan menggambar.

II. METODA DAN MATERIAL AKSES KAMERA

Pertama kita perlu mengakses ESP32CAM dengan

python, cara ini dapat dilakukan dengan memprogram ESP32CAM dengan kode default pada “Camera Web Server Example” pada Library ESP32, kemudian akses kamera menggunakan cv2.VideoCapture(“(link kamera)”) Ekstraksi warna primer RGB : Pada tahap ini sebuah pixel dari tangkapan kamera di uraikan, isinya adalah nilai dari warna Red, Green, dan Blue yang masing-masing memiliki nilai dari 0-255. Pemisahan data dari dataset

```
index = ["color_name", "HEX", "R", "G", "B", "HUE", "HSL", "HSV"]
```

Kalkulasi jarak terdekat dari koordinat warna pada dataset : Jarak terdekat dikalkulasi dengan membandingkan nilai dari warna yang dideteksi dengan yang ada di dataset.

Library panda pada python berperan penting dalam hal ini untuk melakukan berbagai operasi pada nilai yang dipisahkan koma seperti pd.read.csv()

```
D = abs(R-int(csv.loc[i, "R"])) + abs(G-int(csv.loc[i, "G"])) + abs(B-int(csv.loc[i, "B"]))
```

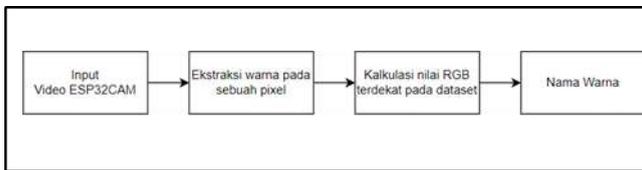
(syntax untuk mencari koordinat warna terdekat)

Menampilkan Hasil kalkulasi dan warna yang dideteksi : Ditampilkan warna yang dideteksi, nilai deteksi, dan nama warna yang memiliki nilai terdekat dari dataset. Menggunakan sebuah kotak pada bagian bawah tampilan layar.

Untuk menampilkan nama dari warna dan nilai yang dideteksi dapat menggunakan.

```
text = get_color_name(r, g, b) + ' R=' + str(r) + ' G=' + str(g) + ' B=' + str(b)nload/153/116 (accessed Sep. 3, 2024).
```

III. CARA KERJA SISTEM



Pada gambar diatas adalah cara kerja dari deteksi warna, dengan input video dari ESP32CAM, kemudian di ekstrak sebuah pixel dan di uraikan nilai RGB nya kemudian dikalkulasikan nilai terdekat pada dataset dan diperoleh namanya.

IV. EKSPERIMEN



Pada eksperimen diatas diperoleh warna grizzly dengan nilai deteksi $R = 144$, $G = 92$, dan $B = 30$ sedangkan pada datasetnya yakni warna Grizzly memiliki nilai $R = 136$, $G = 88$, dan $B = 24$, warna yang Nampak pada mata penulis

adalah warna oranye dengan sedikit kecoklatan dan hasil deteksi terlihat sedikit gelap dari kondisi aslinya.

V. KESIMPULAN

Pada tulisan ini kita dapat mengetahui warna dari yang dideteksi oleh ESP32CAM dengan membandingkannya pada dataset dengan menggunakan modul pandas dan opencv pada pyth

REFERENSI

- [1] P. Raguraman, A. Meghana, Y. Navya, Sk. Karishma, and S. Iswarya, "Color detection of RGB images using python and opencv," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 109–112, Jan. 2021. doi:10.32628/cseit217119
- [2] M. Z. Andrekha and Y. Huda, "Deteksi Warna Manggis menggunakan pengolahan citra dengan opencv python," *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, <https://ejournal.unp.ac.id/index.php/voteteknika/article/view/14251/105691> (accessed Sep. 3, 2024).
- [3] Sutojo, T., Mulyanto, E., Suhartono, V., & NURHAYATI, O. D. (2009). *Teori pengolahan citra digital*.
- [4] *Pengolahan Citra Digital Untuk Mendeteksi ...*, <https://publikasi.dinus.ac.id/index.php/semantik/article/dow>
- [5] "OpenCV-python tutorials," *OpenCV*, https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html (accessed Sep. 2024)