

Identifikasi Warna Pada Objek Citra Digital Secara Real Time Menggunakan Pengolahan Model Warna HSV

Aaqila Dhiyaanisafa Goenawan¹, M. Bakhara Alief Rachman², Mutiara Persada Pulungan³^{1,2,3}
Ilmu Komputer
STIMIK ESQ
Kampus Menara 165, Lantai 18 & 19, Jl. Tb. Simatupang Kav. 1, Cilandak Timur, Pasar Minggu
a.dhiyaanisafa@students.esqbs.ac.id¹, m.bakhara.a.r@students.esqbs.ac.id¹,
m.persada.p@students.esqbs.ac.id¹

ABSTRAK

Teknologi sekarang sudah menjadi satu kesatuan dengan kehidupan manusia. Dengan teknologi, kehidupan manusia menjadi lebih terbantu. Teknologi sekarang telah berkembang menjadi beberapa bentuk. Salah satu bentuk hasil perkembangan teknologi adalah Machine Learning. Machine learning adalah sebuah bentuk teknologi berupa program komputer yang nantinya program tersebut dapat belajar dan menyelesaikan masalahnya sendiri. Salah satu masalah yang dapat diselesaikan dengan Machine Learning adalah identifikasi warna pada sebuah citra digital. Hasil dari penelitian ini adalah penjabaran proses kerja program dan hasil akhirnya.

Kata Kunci: Machine Learning, Identifikasi Warna, Citra Digital.

ABSTRACT

Technology has now become an integral part of human life. With technology, human life becomes more helpful. Technology has now developed into several forms. One of the results of technological developments is Machine Learning. Machine learning is a form of technology in the form of a computer program that later the program can learn and solve its own problems. One of the problems that can be solved by machine learning is the identification of colors in a digital image. The result of this research is the description of the program work process and the final result.
Keywords: Machine Learning, Color Identification, Digital Image.

1. Pendahuluan

1.1. Latar Belakang

Sebagai manusia, kita pasti sering menemui berbagai masalah dalam kehidupan. Salah satu kemampuan yang diberikan oleh Tuhan kepada manusia adalah mampu untuk mempelajari masalah tersebut. Setelah mempelajari masalah tersebut, secara bertahap manusia akan mulai menemukan solusi dan akhirnya masalah tersebut dapat diselesaikan. Dengan berkembangnya zaman, teknologi juga ikut berkembang. Teknologi berkembang menjadi beberapa bentuk. Salah satu bentuk berkembangnya teknologi adalah Machine learning.

Machine learning adalah sebuah program komputer yang nantinya mampu untuk kita latih dan dapat menyelesaikan suatu masalah dengan sendirinya. Salah satu contoh manfaat adanya machine learning adalah mampu untuk melakukan identifikasi warna pada sebuah citra digital. Dengan adanya manfaat tersebut, nantinya teknologi ini mampu untuk membantu manusia, terutama manusia yang lahir dalam keadaan buta warna.

Buta warna merupakan kondisi di mana penderitanya mengalami kesulitan membedakan warna tertentu (buta warna sebagian) atau bahkan keseluruhan warna (buta warna total). Berdasarkan data dari Colour Blind Awareness, setidaknya 300 juta orang di seluruh dunia mengidap buta warna. Oleh karena itu, dibuatnya laporan ini adalah untuk menjelaskan proses apa yang nantinya dilalui untuk penelitian ini dan bagaimana hasil akhirnya.

1.2. Perumusan Masalah

Berdasarkan latar belakang diatas dapat dirumuskan menjadi beberapa masalah, yaitu:

1. Bagaimana cara membuat program yang dapat menganalisis warna pada sebuah citra digital?
2. Apa saja objek citra digital yang dimanfaatkan?
3. Bagaimana hasil akhir dari analisis citra digital?

1.3. Batasan Masalah

Berdasarkan latar belakang diatas dapat dirumuskan menjadi beberapa batasan masalah, yaitu:

1. Program untuk mengidentifikasi warna secara *real time*.
2. Kualitas alat penangkap citra masih rata-rata.
3. Konversi model warna dari BGR ke HSV yang masih belum akurat.

1.4. Tujuan dan Manfaat Penelitian

Tujuan penelitian ini adalah :

1. Menjabarkan proses pembuatan program citra digital
2. Apa saja objek citra digital yang dijadikan input
3. Memaparkan hasil akhirnya

2. Landasan Teori

2.1 Citra Digital

Citra adalah suatu foto atau gambar statis yang berasal dari sensor vision (webcam). Sedangkan digital merupakan pengolahan citra/gambar dilakukan secara digital menggunakan komputer. Secara matematis, citra merupakan suatu fungsi kontinu dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit.

Citra digital merupakan gambar dua dimensi $f(x,y)$ dimana dilakukan diskritisasi koordinat spasial/sampling dan diskritisasi tingkat kuantisasi (kecemerlangan/ keabuannya) yang bisa ditampilkan pada layar komputer sebagai diskrit nilai digital yang disebut pixel. Sebuah citra digital dapat diwakili oleh sebuah matriks dimana indeks baris maupun kolomnya menyatakan sebuah titik pada citra tersebut dan elemen matriksnya (yang dikenal sebagai elemen gambar/ picture element/ piksel/ pixel/ pels) menyatakan tingkat keabuan di titik tersebut.

Citra digital tidaklah selalu merupakan hasil langsung dari data rekaman sebuah sistem. Tetapi terkadang merupakan hasil rekaman data yang sifatnya kontinyu seperti gambar pada monitor tv, foto pada sinar-x dan lain-lain. Dengan begitu untuk memperoleh suatu citra digital dibutuhkan sebuah proses konversi, sehingga selanjutnya citra tersebut bisa diproses menggunakan komputer.

2.2 Pengolahan Citra Digital

Pengolahan Citra Digital merupakan proses pengolahan dan analisis citra atau teknik-teknik mengolah, memanipulasi dan memodifikasi citra yang banyak melibatkan persepsi visual. Dalam proses ini, memiliki ciri data masukan dan informasi keluaran dalam bentuk citra. Secara umum, istilah pengolahan citra digital diartikan sebagai pemrosesan citra dua dimensi dengan menggunakan komputer. Fungsi utamanya adalah meningkatkan kualitas gambar agar terlihat lebih jelas. Hal ini ditegaskan oleh Munir (2014) bahwa masukan dari proses pengolahan citra adalah citra dan keluarannya juga citra yang berkualitas lebih baik dari pada citra masukannya.

2.3 Model Warna RGB

Model warna RGB merupakan singkatan dari Red (Merah) Green(Hijau) Blue(Biru). Model warna RGB juga disebut additive color atau warna pencahayaan karena apabila RGB dikombinasikan maka akan menghasilkan warna putih. RGB merupakan model warna yang paling dasar dalam melakukan penyimpanan citra. Model warna ini juga sering digunakan untuk menampilkan citra pada layar komputer. Pada setiap pixel warna memiliki rentang nilai intensitas mulai dari 0 sampai dengan 255. Rentang intensitas ini bernilai 8bit. Setiap titik yang berada pada ruang warna RGB merupakan warna dengan memiliki komponen R, G dan B. Untuk titik (0,0,0) merupakan titik warna yang berwarna hitam, sedangkan titik (1,1,1) merupakan titik warna yang berwarna putih.

2.4 Model Warna HSV

Model warna HSV (Hue, Saturation, Value) adalah salah satu model warna alternatif lain selain model warna RGB. Hue menyatakan warna asli, seperti merah, ungu, dan kuning. Hue digunakan untuk memisahkan antara warna dan menentukan warna kemerahan, kehijauan, dan sebagainya dari cahaya. Hue dikaitkan dengan panjang gelombang cahaya. Saturation menyatakan tingkat kemurnian suatu warna, yang menunjukkan seberapa banyak warna putih diterapkan pada warna tersebut. Nilai adalah atribut yang menyatakan jumlah cahaya yang diterima mata terlepas dari warna.

Mirip seperti saturation, value ini memiliki rentang dari 0 hingga 1. Apabila nilai value ini sebesar 0. Maka warna tersebut akan menjadi hitam. Sedangkan apabila nilai value bernilai 1 maka tingkat kandungan warna hitam pada warna tersebut menjadi hilang. Kelebihan HSV adalah terdapat warna - warna yang sama dengan yang ditangkap oleh indera manusia. Sedangkan warna yang dibentuk model lain seperti RGB merupakan hasil campuran dari warna - warna primer.

2.5 Machine Learning

Machine learning dapat didefinisikan sebagai aplikasi komputer dan algoritma matematika yang bisa bekerja tanpa bantuan program manusia berulang-ulang. Ilmu mesin bisa belajar sendiri dengan cara menganalisa data, misalnya mengenali wajah hewan kucing dengan anjing. Salah satu teknik pengaplikasian machine learning adalah tanpa data maka tidak akan bisa bekerja. Oleh karena itu hal yang pertama kali disiapkan adalah data. Data biasanya akan dibagi menjadi 2 kelompok, yaitu data training dan data testing. Data training nantinya akan digunakan untuk melatih algoritma untuk mencari model yang cocok, sementara data testing akan dipakai untuk mengetes dan mengetahui performa model yang didapatkan pada tahapan testing.

2.6 OpenCV

OpenCV (Open Source Computer Vision Library), adalah sebuah library open source yang berisi pemrograman untuk teknologi computer vision secara realtime. Open source computer vision library menggunakan bahasa pemrograman C/C++, dan telah dikembangkan ke Python, Java, Matlab. Salah satu tujuan OpenCV adalah untuk menyediakan infrastruktur computer vision yang mudah digunakan sehingga membantu pemakainya membangun aplikasi vision yang cukup canggih dengan cepat. OpenCV digunakan secara luas di perusahaan, kelompok peneliti, dan badan pemerintahan.

2.7 NumPy

NumPy (Numerical Python) adalah library Python yang fokus pada scientific computing. NumPy memiliki kemampuan untuk membentuk objek N-dimensional array, yang mirip dengan list pada Python. Struktur data NumPy lebih membutuhkan ukuran yang lebih kecil dibandingkan dengan List tetapi mempunyai performa yang lebih cepat. Keunggulan NumPy array dibandingkan dengan list pada Python adalah konsumsi memori yang lebih kecil serta runtime yang lebih cepat.

3. Metodologi Penelitian

Gambar 1. Flowchart Pengolahan Citra

Berikut penjelasan mengenai flowchart pengolahan citra pada Gambar 1

1. Mulai
Untuk memulai proses pengolahan data citra, peneliti menjalankan program yang telah ditulis dalam bahasa pemrograman Python dengan menggunakan IDE Anaconda Spyder (*Scientific Python Development Environment*).
2. Mengambil video dari *webcam* secara *real time*
Setelah menjalankan program, *webcam* pada laptop akan terbuka secara otomatis sesuai dengan perintah yang dituliskan pada program.
3. Mendapat citra dari video
Webcam yang terbuka akan menangkap citra per *frame* secara *real time* seperti merekam video.
4. Konversi BGR ke HSV
Dikarenakan citra yang ditangkap oleh *webcam* menggunakan model warna BGR, dilakukan konversi dari model warna BGR ke model warna HSV karena model warna HSV menampilkan warna “aktual” dengan memisahkan luma (intensitas gambar) dari kroma (informasi warna) dan membuat ruang warna pada HSV lebih luas daripada BGR. Dengan konversi model warna BGR ke HSV, komputer juga dapat

Identifikasi Warna Pada Objek Citra Digital Secara Real Time Menggunakan Pengolahan Model Warna HSV

menghilangkan bayangan dari citra yang ditangkapnya. Metode ini dilakukan dengan menggunakan alat bantu program HSV-Color-Picker yang dirancang oleh pengguna Github dengan *username* aliieldinayman.

5. Menentukan batas atas dan batas bawah model warna HSV
Skala model warna HSV bergantung pada *library* yang digunakan. Program yang dirancang menggunakan *library* OpenCV yang memiliki skala H: 0-179, S: 0-255, V: 0-255. Peneliti menentukan batasan-batasan secara manual dengan bantuan HSV *colormap* yang dibuat oleh salah satu pengguna stack overflow dengan *username* Kinght 金. Pada penelitian ini, kami menentukan 8 batasan-batasan warna pada model warna HSV yaitu warna merah, kuning, hijau, biru, coklat, hitam, oranye, dan *cyan*.

Gambar 2. HSV *Colormap*

6. Membuat *mask* untuk setiap warna
Selanjutnya, *mask* digunakan untuk memfokuskan pembacaan citra pada bagian yang ingin dibaca saja. Dalam penelitian ini, proses *masking* digunakan untuk memfokuskan pembacaan citra pada warna-warna yang telah ditentukan.
7. Morfologi: Dilasi pada citra
Proses ini dilakukan untuk menghilangkan noise pada citra yang ditangkap. Disini, kami menggunakan proses dilasi untuk memperbaiki gangguan pada citra.
8. Bitwise_and diantara citra dan mask
Operasi ini dilakukan untuk menerapkan *mask* yang telah dibuat kepada citra.
9. Membuat *contour* untuk setiap warna
10. *Contour* digunakan untuk mencari objek tertentu dari sebuah citra. Dalam penelitian ini, *contour* digunakan untuk mencari objek-objek warna yang telah ditentukan.
11. Penampilan data
Setelah semua proses selesai, program akan menampilkan data hasil pengolahan citra.

4. Hasil dan Pembahasan

4.1 Hasil

Penelitian untuk mengidentifikasi warna-warna secara *real time* ini dirancang dengan menggunakan bahasa pemrograman Python, *library* pendukung yaitu OpenCV dan NumPy untuk melakukan *image processing*, serta model warna HSV yang berasal dari hasil konversi model warna BGR. Pengujian ini dilakukan dengan menggunakan *rear camera handphone* Huawei Nova 2i dengan resolusi 16MP + 2MP. Pembahasan detail pada pengujian ini akan membahas mengenai identifikasi warna merah.

Sebelum dilakukan identifikasi warna, program akan membuka kamera secara otomatis seperti pengambilan video dan mengambil citra per *frame*. Langkah pertama yang akan dilakukan oleh program adalah melakukan konversi model warna citra, yang awalnya BGR, menjadi model warna HSV.

```
hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)
```

Gambar 2. Pembacaan citra dengan model warna HSV

Proses selanjutnya adalah mendeteksi warna-warna dari citra yang ditangkap dengan batasan-batasan atas dan bawah yang telah ditentukan untuk program membaca warna-warnanya.

```
# Penentuan batas bawah dan atas pada warna merah
red_lower = np.array([136, 87, 111], np.uint8)
red_upper = np.array([180, 255, 255], np.uint8)
red_mask = cv2.inRange(hsvFrame, red_lower, red_upper)
```

Identifikasi Warna Pada Objek Citra Digital Secara Real Time Menggunakan Pengolahan Model Warna HSV

Warna	HSV Batas Bawah (<i>Lower Range</i>)	HSV Batas Atas (<i>Upper Range</i>)
Merah	[136, 87, 111]	[180, 255, 255]
Hijau	[25, 52, 72]	[102, 255, 255]
Biru	[94, 80, 2]	[120, 255, 255]
Coklat	[5, 164, 61]	[25, 184, 141]
Kuning	[25, 70, 120]	[30, 255, 255]
Oranye	[10, 100, 20]	[25, 255, 255]
Cyan	[85, 100, 20]	[95, 255, 255]
Black	[0, 0, 0]	[50, 50, 50]

Tabel 1. Batas Atas dan Batas Bawah Warna

Berdasarkan nilai yang telah didapatkan dari penentuan batas atas dan bawah, program akan membuat *masking* terlebih dahulu untuk memfokuskan pembacaan dan melakukan morfologi dengan dilasi untuk menghilangkan noise pada citra yang dibaca. Setelah itu akan dilakukan operasi *bitwise_and* untuk menerapkan hasil dari dilasi dan *masking* pada citra yang akan dibaca.

Gambar 3. Hasil Dilasi

Gambar 4. Hasil Bitwise

Proses terakhir adalah proses *contour*, dimana proses ini merupakan proses untuk mencari objek spesifik pada citra. Proses ini merupakan proses untuk menandai warna yang sedang diidentifikasi.

Gambar 5. Hasil identifikasi warna merah dengan *Contour*

Gambar 6. Hasil identifikasi banyak warna dengan *Contour*

4.2 Pembahasan

Identifikasi warna pada objek citra sangat dipengaruhi oleh resolusi kamera dan lensa yang digunakan karena semakin tinggi resolusi, semakin bagus kamera dalam menangkap piksel pada citra serta semakin sedikit *noise* yang ditangkap. Lensa yang bagus juga mempengaruhi ketajaman, kontras, dan warna citra. Dengan itu, keakuratan identifikasi beberapa warna pada pengujian ini masih belum sempurna dikarenakan resolusi kamera dan lensa yang belum sempurna, serta pencahayaan pada ruangan pengambilan citra.

Warna	Jumlah Sampel	Tingkat Keakuratan (%)
Merah	2	100%
Hijau	2	100%
Biru	2	100%
Coklat	2	50%
Kuning	2	50%
Oranye	2	95%
Cyan	2	100%
Black	2	100%

Tabel 2. Keakuratan Identifikasi Warna

Tingkat keakuratan pada pengujian ini diukur dari seberapa akuratnya program dapat mengidentifikasi warna. Pada tabel diatas, warna coklat, kuning, oranye, dan cyan belum memiliki tingkat keakuratan 100% karena warna-warna tersebut masih teridentifikasi secara bersamaan, seperti coklat dan oranye, hitam dan biru, serta kuning dan hijau dikarenakan *range* warnanya yang mirip. Faktor lain yang mempengaruhi keakuratan adalah resolusi kamera, kualitas lensa, serta pencahayaan pada ruangan pengambilan citra. Dengan itu, bisa dikatakan bahwa tingkat akurasi identifikasi warna sekitar 86.8%.

5 Kesimpulan dan Saran

5.1 Kesimpulan

Dari percobaan yang dilakukan dapat disimpulkan bahwa program pengenalan warna menggunakan ruang warna HSV berhasil menjawab tujuan penelitian, yaitu:

1. Penelitian ini dilakukan menggunakan bahasa pemrograman python dan beberapa library seperti opencv dan numpy. Program yang dirancang bertujuan untuk membantu para penderita buta warna agar dapat membedakan warna dengan benar. Program dapat mendeteksi tujuh warna bersamaan secara real time, yaitu merah, biru, hijau, kuning, coklat, orange, hitam dan cyan dengan tingkat akurasi hingga 86,8%
2. Kondisi terbaik untuk menggunakan program pengenalan warna yaitu dengan dukungan kualitas webcam yang digunakan serta pencahayaan ruangan.

5.2 Saran

Berdasarkan hasil penelitian yang dilakukan, pada dasarnya penelitian ini berjalan dengan baik. Namun penulis ingin mengemukakan beberapa saran yang mudah-mudahan bermanfaat bagi kemajuan pendidikan pada umumnya. Adapun saran yang penulis ajukan adalah sebagai berikut :

1. Hendaknya para peneliti selanjutnya dapat mengembangkan program ini kedalam program identifikasi dan klasifikasi lainnya.
2. Hendaknya para peneliti selanjutnya dapat menambahkan pengenalan warna-warna yang lainnya.

Referensi

- [1] Baum, David, 1999. *E-commerce* . New Jersey : oracle corp
- [2] Hadi, S. A., Purnama, J., Amin Soetomo, M. A., & Galinium, M. (2015). C2C *E-commerce* trust level measurement and analysis in Indonesia. Proceedings - 2014 6th International Conference on Information Technology and Electrical Engineering: Leveraging Research and Technology Through University-Industry Collaboration, ICITEE 2014. <https://doi.org/10.1109/ICITEED.2014.7007900>.
- [3] RY, Novie Noordiana. "faktor-faktor yang memiliki kontribusi pada keberlangsungan dan kesuksesan umkm yang dikelola oleh kalangan usia muda." jurnal relasi stie mandala jember 13.1 (2017).
- [4] Tsalgatidou, A., & Pitoura, E. (2001). Business models and transactions in mobile electronic commerce : requirements and properties, 37, 221–236.
- [5] Oliveira, T., Alhinho, M., Rita, P., & Dhillon, G. (2017). Modelling and Testing Consumer Trust Dimensions in *E-commerce* . Computers in Human Behavior. <https://doi.org/10.1016/j.chb.2017.01.050>
- [6] Werneck, V. M. B., Oliveira, A. de P. A., & do Prado Leite, J. C. S. (2009). Comparing GORE Frameworks: i-star and KAOS. Wer, (January), 1–12. Retrieved from http://wer-papers.googlecode.com/svn/history/r71/trunk/dataset/wer09/WER09_4.pdf

Identifikasi Warna Pada Objek Citra Digital Secara Real Time Menggunakan Pengolahan Model Warna HSV

- [7] Halodoc. (2019, October 13). *Kapan Waktu Yang tepat Lakukan tes Buta Warna?* halodoc. Retrieved January 5, 2022, from <https://www.halodoc.com/artikel/kapan-waktu-yang-tepat-lakukan-tes-buta-warna>
- [8] *Buta Warna*. Alodokter. (2021, November 13). Retrieved January 5, 2022, from <https://www.alodokter.com/buta-warna>
- [9] *Choosing the correct upper and lower HSV boundaries for color detection with `cv::inrange` (opencv)*. Stack Overflow. (1960, June 1). Retrieved January 5, 2022, from <https://stackoverflow.com/questions/10948589/choosing-the-correct-upper-and-lower-hsv-boundaries-for-color-detection-withcv/48367205#48367205>
- [10] *Multiple color detection in real-time using python-opencv*. GeeksforGeeks. (2020, May 10). Retrieved January 5, 2022, from <https://www.geeksforgeeks.org/multiple-color-detection-in-real-time-using-python-opencv/>
- [11] I. S. Areni, I. Amirullah, and N. Arifin, "Klasifikasi Kematangan Stroberi Berbasis Segmentasi Warna dengan Metode HSV," *J. Penelit. Enj.*, vol. 23, no. 2, pp. 113–116, 2019, doi: 10.25042/jpe.112019.03.
- [12] M. Z. Andrekha and Y. Huda, "Deteksi Warna Manggis Menggunakan Pengolahan Citra dengan Opencv Python," *Voteteknika (Vocational Tek. Elektron. dan Inform., vol. 9, no. 4, p. 27, 2021, doi: 10.24036/voteteknika.v9i4.114251.*
- [13] R. D. Kusumanto, A. N. Tompunu, and S. Pambudi, "Klasifikasi Warna Menggunakan Pengolahan Model Warna HSV Abstrak," *J. Ilm. Tek. Elektro*, vol. 2, no. 2, pp. 83–87, 2011.
- [14] A. N. M. Nayif, R. Magdalena, and E. Susatio, "Aplikasi Pengenal Warna Untuk Penderita Buta Warna Menggunakan Pengolahan Model Warna HSV," vol. 7, no. 1, pp. 586–593, 2020.
- [15] *Image masking with opencv*. PyImageSearch. (2021, April 17). Retrieved January 5, 2022, from <https://www.pyimagesearch.com/2021/01/19/image-masking-with-opencv/>
- [16] *Contour detection using opencv (python/C++)*. LearnOpenCV. (2021, October 18). Retrieved January 5, 2022, from <https://learnopencv.com/contour-detection-using-opencv-python-c/>