

Sistem Pengenalan Warna Menggunakan Metode K-Nearest Neighbors

Alwin Marcellino¹, Fionna Caroline²

^{1,2}Jl. Rajawali 14, 30113 Palembang, (0711) 376 400

³Program Studi Informatika, FIKR, Universitas Multi Data Palembang, Palembang

e-mail: *¹alwinmarcellino@mhs.mdp.ac.id, ²fionnacaroline@mhs.mdp.ac.id

Abstrak

Penelitian ini berisi tentang pemanfaatan metode K-Nearest Neighbors (KNN) untuk meningkatkan efisiensi sistem pengenalan warna. Melibatkan pelatihan model dengan dataset warna yang representatif, penelitian ini menemukan parameter optimal yaitu fitur warna Red Green Blue (RGB) untuk meningkatkan akurasi pendeteksian warna. Dataset tersebut diklasifikasikan menggunakan metode KNN dengan menghitung jarak tetangga terdekat dengan nilai $K=3$. Sistem yang diimplementasikan menunjukkan kinerja handal dalam mengatasi variasi warna.

Kata kunci— K-Nearest Neighbors, Pengenalan Warna, Red Green Blue, Pengolahan Gambar

Abstract

This study revolves around the utilization of the K-Nearest Neighbors (KNN) method to enhance the efficiency of a color recognition system. Involving the training of the model with a representative color dataset, the research identifies optimal parameters, specifically the Red Green Blue (RGB) color features, to improve color detection accuracy. The dataset is classified using the KNN method by calculating the distances to the nearest neighbors with a K -value of 3. The implemented system demonstrates robust performance in handling color variations.

Keywords— K-Nearest Neighbors, Color Recognition, Red Green Blue, Image Processing.



This is an open-access article under the [CC-BY-CA](https://creativecommons.org/licenses/by-sa/4.0/) license

1. PENDAHULUAN

Warna memiliki peran yang penting dalam kehidupan sehari-hari dan memegang peran krusial dalam berbagai aplikasi teknologi, seperti pengolahan citra, pengenalan pola, dan lain-lain. Sistem pengenalan warna menjadi salah-satu bidang penelitian yang menarik karena dapat memberikan kontribusi signifikan terhadap perkembangan teknologi. Salah satu metode yang banyak digunakan dalam pengenalan warna adalah Metode K-Nearest Neighbors (KNN), yang terkenal karena sederhana namun efektif, dengan klasifikasi objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek baru sehingga hasilnya bisa lebih akurat [5].

K-NN merupakan salah satu jenis algoritma machine learning yang bekerja dengan prinsip bahwa "setiap titik data yang berdekatan satu sama lain akan berada di kelas yang sama. Dengan kata lain, KNN mengklasifikasikan titik data baru berdasarkan kemiripan." Dalam konteks pengenalan warna, K-NN dapat digunakan untuk mengklasifikasikan warna berdasarkan kemiripan dengan fitur warna *red*, *green*, *blue* (RGB) yang sudah dikenali sebelumnya.

Tujuan dari penelitian ini adalah untuk menghasilkan sistem pengenalan warna yang akurat dan efektif dalam mengklasifikasikan citra berdasarkan fitur warna RGB. Metode validasi yang digunakan dalam penelitian ini adalah eksperimen dengan menggunakan citra digital sebagai data uji. Data uji akan diolah menggunakan algoritma K-NN untuk mengklasifikasikan citra berdasarkan fitur warna, dan kemudian dilakukan evaluasi terhadap akurasi sistem pengenalan warna yang dihasilkan.

Penelitian dengan tema yang serupa sudah pernah dilakukan sebelumnya oleh R. D. Kusumanto, dkk (2011) dengan menggunakan pengolahan model warna *hue, saturation, value* (HSV) untuk menentukan jenis warna secara realtime. Peneliti tersebut berpendapat bahwa warna coklat memiliki respon yang baik dengan menggunakan metode tersebut dibandingkan dengan 5 jenis warna lainnya, yaitu kuning, hijau, biru, hitam, dan putih [1].

Penelitian selanjutnya oleh Syamsul Mujahidin (2015) dengan judul Klasifikasi Warna Kulit berdasarkan Ruang Warna RGB. Penelitian ini dilakukan dengan membandingkan tingkat presisi dari beberapa metode yang digunakan, yaitu Bayesian sebesar 94.7416%, Multi Perceptrons 99.4552%, dan K-NN sebesar 99.2351% [2].

Selanjutnya penelitian yang dilakukan oleh Nurlindasari Tamsir, dkk (2020) dengan judul Aplikasi Citra Untuk Mendeteksi dan Mengenali Warna Menggunakan Algoritma Midpoint. Penelitian ini bertujuan untuk merancang aplikasi pengolahan citra dalam mendeteksi warna berbasis android mobile dan mengimplementasikan algoritma midpoint [6].

Berdasarkan ketiga penelitian tersebut, peneliti mendapatkan ide untuk membuat penelitian mengenai identifikasi warna. Dan di era saat ini penelitian tentang identifikasi warna memiliki signifikansi dalam berbagai aplikasi. Dalam pengolahan citra dan visi komputer, kontribusinya terletak pada peningkatan akurasi dan responsivitas sistem. Di bidang robotika dan otomasi, kemampuan identifikasi warna memungkinkan robot berinteraksi lebih efektif dengan lingkungannya, termasuk dalam tugas pemilihan objek dan navigasi. Industri manufaktur mendapatkan manfaat dari identifikasi warna untuk kontrol kualitas produk, sementara di bidang keamanan, sistem dapat membedakan objek atau kejadian dengan lebih akurat. Dalam pertanian dan pemrosesan makanan, identifikasi warna membantu dalam pengelompokan produk, sementara di aplikasi pendidikan dan hiburan, memberikan pengalaman interaktif yang lebih baik. Penerapan identifikasi warna juga meluas ke bidang kesehatan, membantu mendiagnosis kondisi kesehatan melalui identifikasi warna pada gambar medis. Dengan demikian, penelitian ini memberikan dasar untuk pengembangan teknologi yang dapat meningkatkan efisiensi dan akurasi dalam berbagai konteks aplikasi yang melibatkan warna.

2. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah menggabungkan teknik ekstraksi fitur warna dengan algoritma klasifikasi KNN untuk mengenali dan mengklasifikasikan warna dalam gambar. Saat melakukan pengujian, data melewati dua proses yaitu ekstraksi dan klasifikasi. Dengan demikian, metode ini memberikan pemahaman mengenai cara ekstraksi fitur warna dan implementasi sederhana dari algoritma KNN dalam konteks pengenalan warna pada gambar.

2.1 Metode K-means Clustering

Metode K-Means Clustering pada penelitian ini digunakan untuk mengelompokkan warna dalam gambar menjadi beberapa *cluster*, di mana setiap *cluster* diwakili oleh warna pusatnya. Secara umum algoritma K-Means Clustering dimulai dari menentukan nilai *K cluster* awal, nilai *K* yang dimaksud adalah jumlah *cluster* yang ingin dibentuk. Kemudian menentukan nilai awal centroid secara acak. Hitung jarak setiap data yang ada terhadap masing-masing centroid salah satu cara untuk menghitung jarak bisa menggunakan rumus *Euclidean Distance* hingga ditemukan jarak yang paling dekat dari setiap data dengan centroid. Klasifikasikan setiap data

berdasarkan kedekatannya dengan centroid. Perbaharui nilai centroid berdasarkan rata-rata. Lakukan langkah tersebut hingga nilai centroid tidak berubah atau konvergen [7].

2.2 Metode K-Nearest Neighbors

K-Nearest Neighbors (KNN) merupakan salah satu algoritma machine learning yang populer dikarenakan implementasinya yang sederhana. K-Nearest Neighbor bekerja dengan cara mengklasifikasikan suatu data yang memiliki jarak paling dekat dengan data latih dengan nilai k sebagai jumlah tetangga terdekat [3]. Pada penelitian ini, peneliti menggunakan *Euclidean Distance* untuk menghitung jarak antara data baru dan data latih. Secara umum algoritma K-Nearest Neighbors dimulai dari menentukan jumlah tetangga (K) yang akan digunakan dalam pertimbangan penentuan kelas. Menghitung jarak dari data baru ke masing-masing data pada dataset. Mengurutkan jarak yang telah dihitung dan menentukan tetangga terdekat berdasarkan nilai K untuk penentuan kelas dari data baru [4]. Berikut merupakan rumus dari *Euclidean Distance* yang digunakan:

$$d(i, j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad (1)$$

2.3 Pengumpulan Data

Penelitian ini dimulai dengan pengumpulan data, di mana data latih yang diperoleh terdiri dari citra-citra yang memuat berbagai perwakilan warna yang akan dijadikan dasar dalam proses pengenalan. Berikut merupakan contoh data citra yang digunakan:



Gambar 1. Citra
Tes Merah



Gambar 2. Citra
Tes Violet



Gambar 3. Citra
Tes Hijau

2.4 Pra-proses

Pada tahap pra-proses data, terdapat beberapa proses berikut yang dilakukan sebelum dilakukannya ekstraksi warna:

2.4.1 Segmentasi Citra

Segmentasi citra adalah proses membagi citra menjadi bagian-bagian yang berbeda berdasarkan karakteristik tertentu. Pada penelitian ini, metode segmentasi yang digunakan adalah metode Otsu Thresholding yang digunakan untuk memisahkan citra menjadi dua kelas berdasarkan nilai intensitasnya. Ambang batas yang ditentukan memisahkan piksel menjadi dua kelompok berdasarkan perbedaan intensitas, membantu fokus pada objek yang ingin dianalisis.

2.4.2 Operasi Morfologi

Operasi morfologi yang paling dasar adalah dilasi, erosi, opening, dan closing. Pada penelitian ini operasi morfologi yang digunakan adalah dilasi dan erosi. Kedua proses tersebut memiliki rumus sebagai berikut:

$$\text{Dilasi: } A \oplus B = \{x | (B_x) \cap A \neq \emptyset\} \quad (2)$$

$$\text{Erosi: } A \ominus B = \{x | (B_x) \subseteq A\}$$

(3)

Dari rumus tersebut tujuan dari operasi dilasi adalah untuk menemukan nilai maksimum local pada dalam suatu wilayah gambar yang ditetapkan dengan nilai “1”, berkebalikan dengan operasi dilasi, operasi erosi ekuivalen dengan mencari nilai minimum local dalam suatu wilayah gambar yang tertutup oleh “1” [8].

2.5 Ekstraksi Ciri RGB

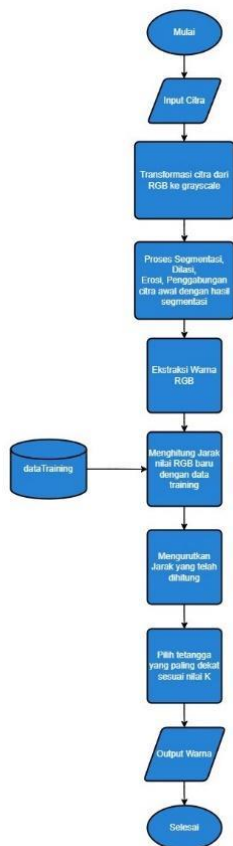
Tahap ini memfokuskan pada ekstraksi yang bertujuan untuk menyoroti karakteristik citra masukan, yang nantinya akan diterapkan dalam proses prediksi menggunakan metode K-Nearest Neighbors (KNN). Dalam penelitian ini, nilai Red, Green, Blue (RGB) diekstraksi melalui penerapan metode K-Means Clustering dengan k=8 sebagai parameter untuk proses ekstraksi.

2.6 Klasifikasi Warna

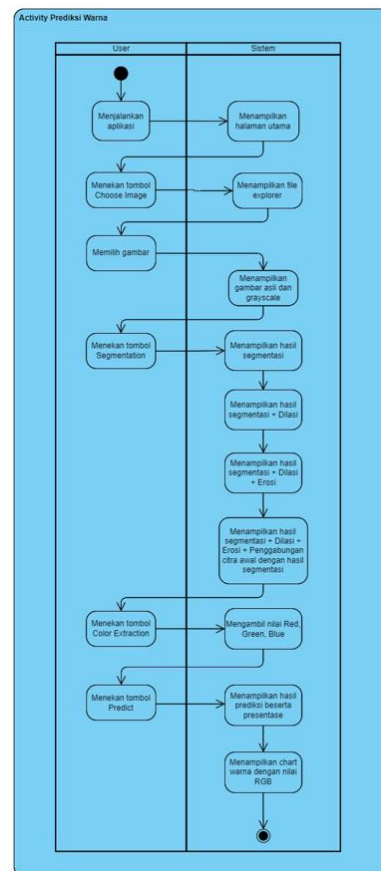
Tahap ini bertujuan untuk mengklasifikasikan warna setiap nilai RGB yang telah diekstrak pada proses sebelumnya. Pada penelitian ini, klasifikasi dilakukan dengan menggunakan metode KNN dengan nilai parameter k=3.

3. HASIL DAN PEMBAHASAN

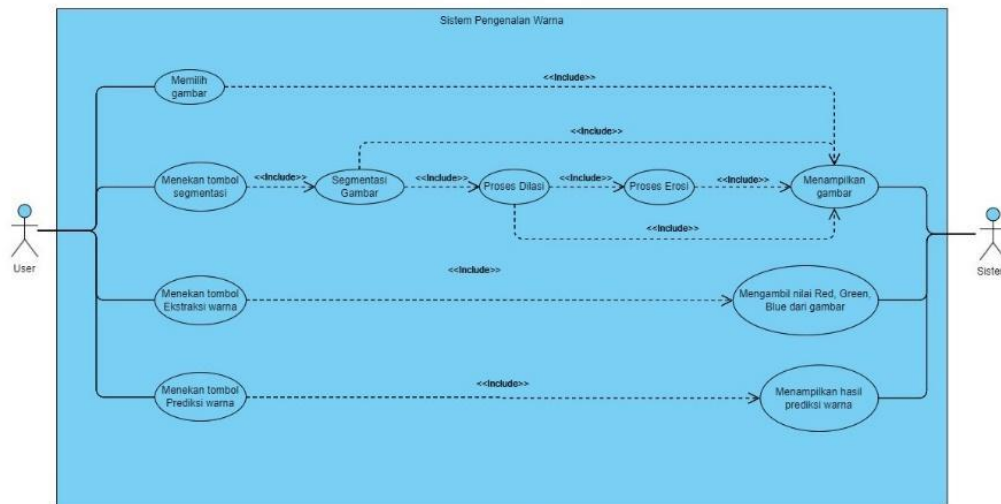
Berikut merupakan alur kerja dari sistem yang telah dibuat:



Gambar 4. Flowchart

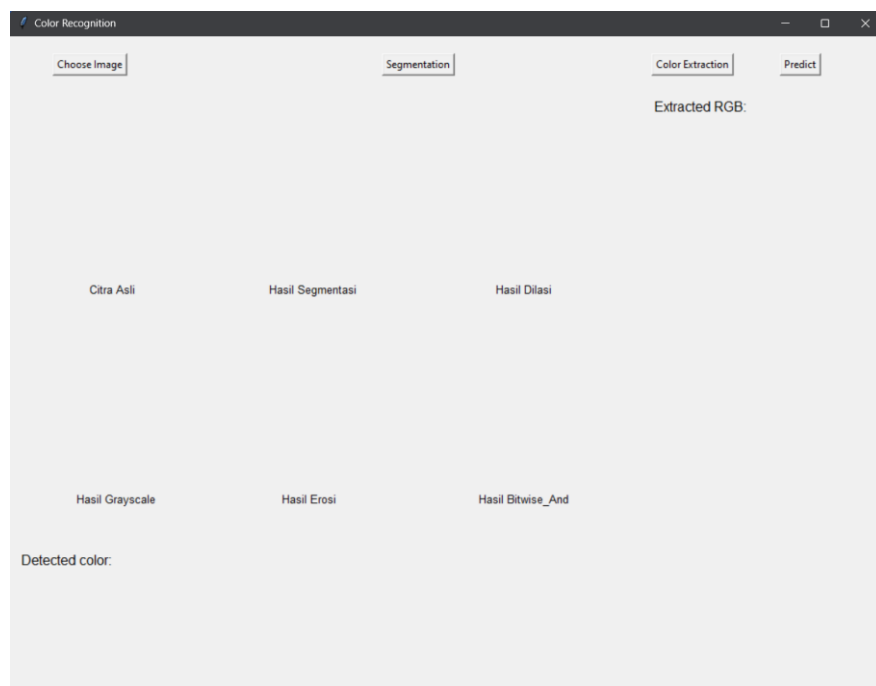


Gambar 5. Activity Diagram



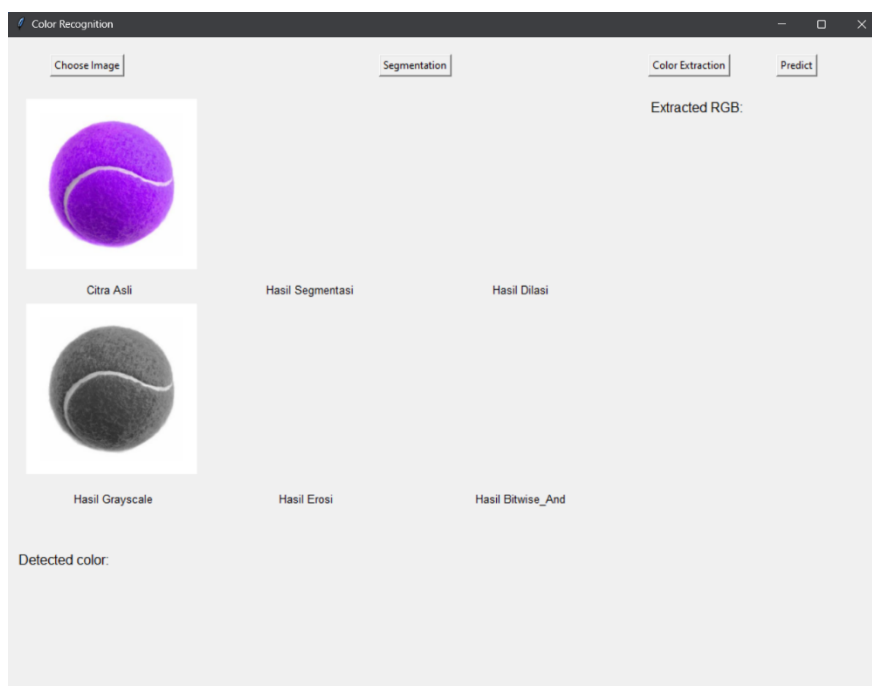
Gambar 6. Use Case Diagram

Sistem dan GUI dibangun dengan menggunakan bahasa pemrograman Python dengan library OpenCV. Ketika program dijalankan, sistem akan menampilkan tampilan antarmuka sederhana yang dapat dilihat dalam gambar berikut:



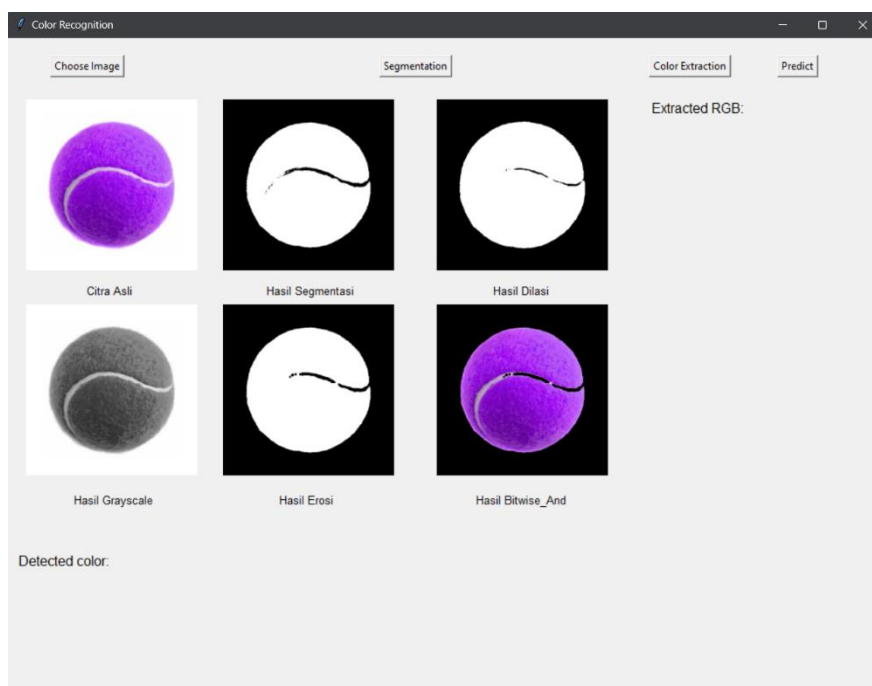
Gambar 7. Tampilan Awal Sistem

Pengguna dapat memasukkan gambar yang ingin diidentifikasi dengan cara menekan tombol *Choose Image* yang ada pada tampilan dan memilih gambar pada jendela file explorer. Sistem akan menampilkan gambar asli dan gambar yang telah diubah menjadi grayscale.



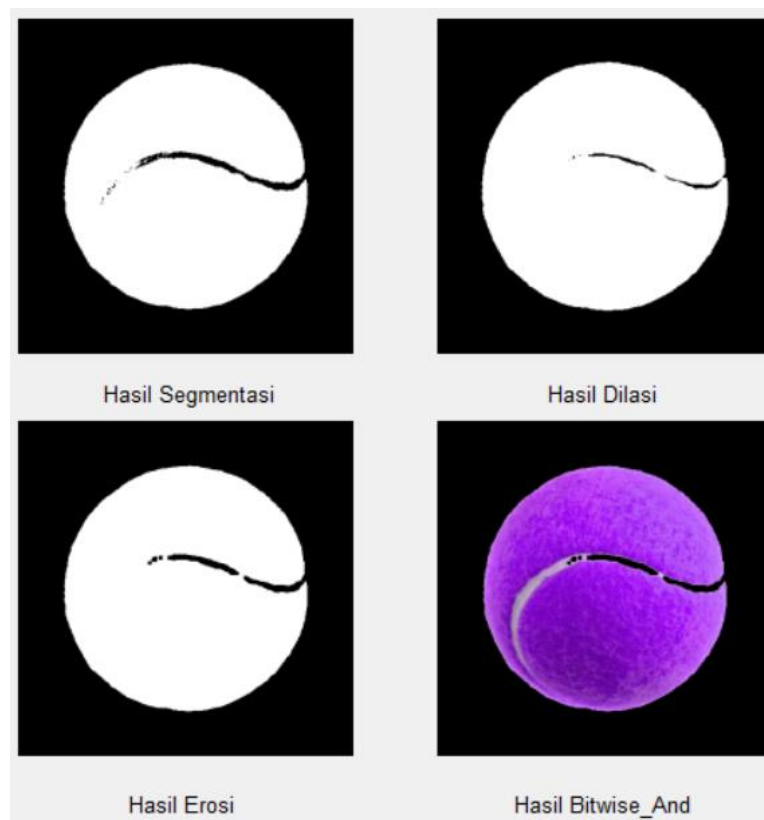
Gambar 8. Tampilan setelah memasukan image

Selanjutnya pengguna dapat menekan tombol *Segmentation* untuk melakukan proses segmentasi terhadap gambar yang sudah dimasukan sebelumnya. Tombol tersebut akan memunculkan 4 gambar dari masing-masing proses yang dilakukan, yaitu segmentasi dengan menggunakan metode *Otsu Threshold*, Dilasi, Erosi, dan penggabungan gambar awal dengan hasil proses sebelumnya (*bitwise*).



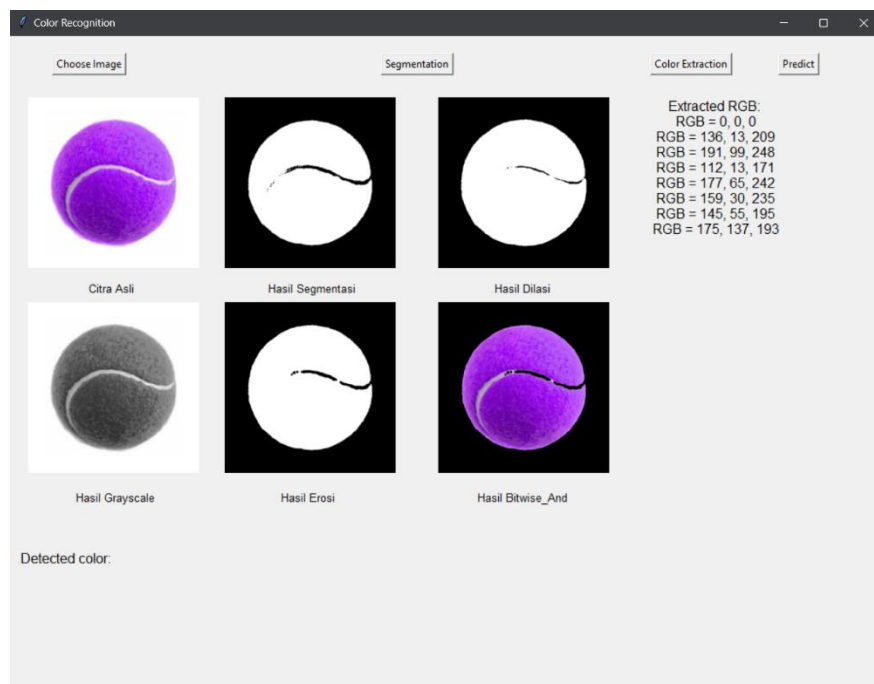
Gambar 9. Tampilan setelah segmentasi

Proses segmentasi dimulai dengan mengkonversi citra asli menjadi citra grayscale yang bertujuan untuk memudahkan analisis citra masukan. Kemudian, dilakukan proses Otsu Tresholding untuk memisahkan objek dengan latar belakangnya. Selanjutnya, dilakukan operasi morfologi, yaitu dilasi dan erosi. Operasi dilasi digunakan untuk memperluas area objek dengan suatu struktur kernel, sehingga dapat mengisi celah-celah kecil dalam objek. Sedangkan erosi dilakukan untuk merapikan tepi objek dengan mengecilkan area objek menggunakan struktur kernel yang sama. Pada penelitian ini kernel yang digunakan berbentuk persegi panjang dengan ukuran 5x5. Terakhir, citra hasil erosi dikonversi menjadi citra berwarna (RGB), lalu dilakukan operasi bitwise AND antara citra asli dengan citra hasil erosi yang bertujuan untuk mendapatkan objek yang telah tersegmentasi dengan akurat. Berikut gambar dari hasil citra segmentasi:

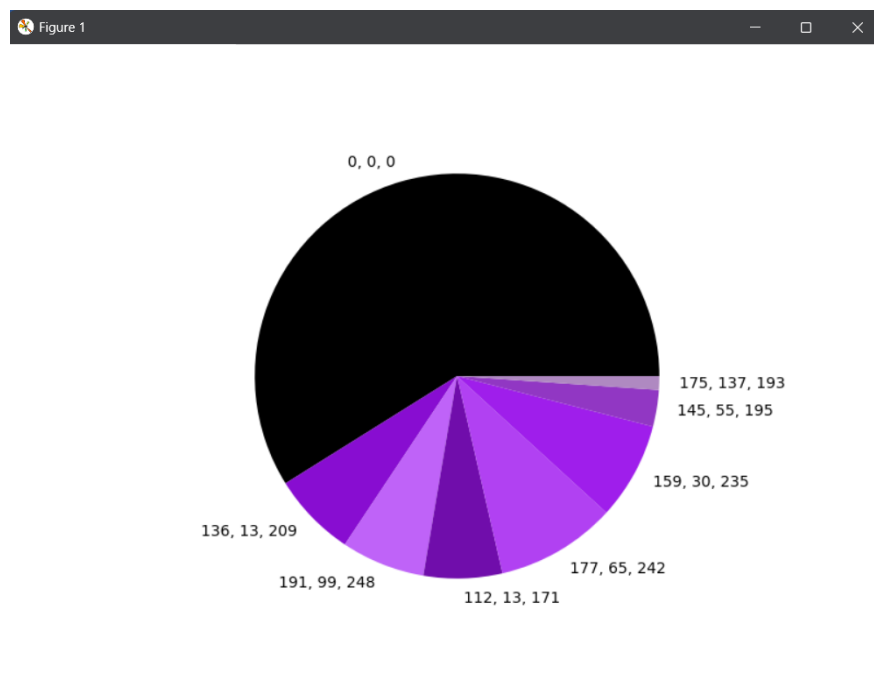


Gambar 10. Hasil segmentasi citra

Kemudian saat pengguna menekan tombol *Color Extraction*, sistem akan mengekstrak nilai RGB dari gambar hasil segmentasi dan menampilkan nilai RGB pada antarmuka, serta menampilkan diagram lingkaran warnanya. Dengan menggunakan nilai $K=8$ sebagai parameter K-means clustering, maka proses ekstraksi akan menghasilkan delapan nilai RGB yang merupakan nilai pusat setiap clusternya.

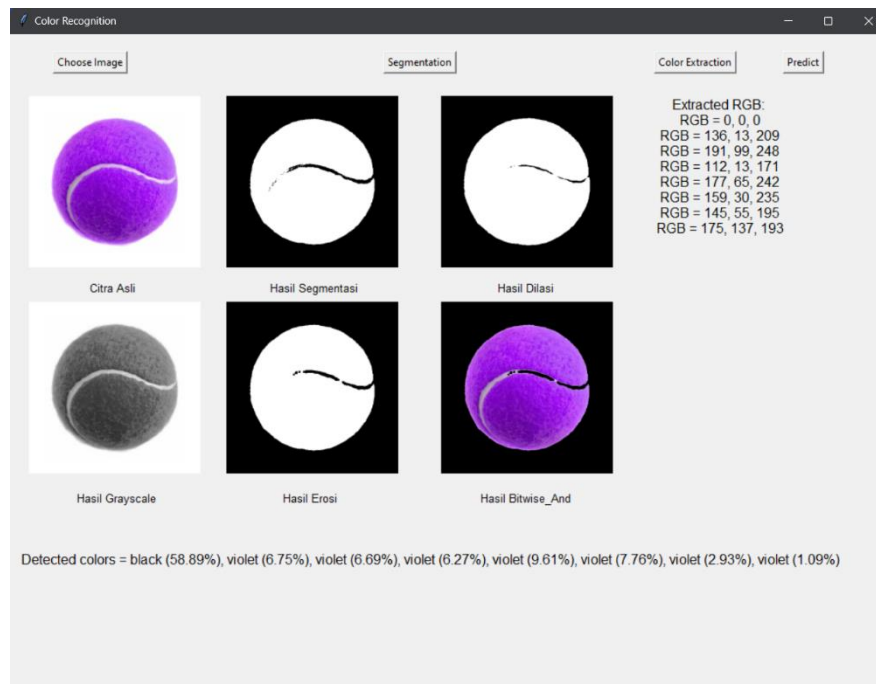


Gambar 11. Tampilan setelah ekstraksi warna



Gambar 12. Tampilan Pie Chart

Setelah pengguna menekan tombol Prediksi, sistem akan menghitung jarak antara setiap nilai RGB baru dengan dataset untuk menentukan prediksi warna dengan parameter $K=3$. Hasil prediksi beserta persentase kandungan setiap warna dalam gambar akan ditampilkan pada antarmuka.



Gambar 13. Tampilan setelah prediksi

4. KESIMPULAN

Dalam penelitian ini, berhasil dikembangkan sebuah sistem pengenalan warna berbasis K-Nearest Neighbors (KNN). Hasil eksperimen menunjukkan bahwa sistem ini mampu mengenali warna dengan tingkat akurasi yang memadai, terutama setelah dilakukan ekstraksi fitur menggunakan metode segmentasi, dilasi, erosi, dan bitwise. Kelebihan utama program ini terletak pada kemampuannya mengenali warna pada gambar, memberikan hasil ekstraksi warna, dan melakukan prediksi dengan cukup baik. Namun, kekurangan mungkin terjadi pada kondisi tertentu, seperti gambar dengan tingkat kompleksitas yang tinggi atau cahaya yang kurang. Kemungkinan pengembangan selanjutnya dapat mencakup peningkatan model KNN dengan penambahan data pelatihan, eksplorasi metode ekstraksi fitur yang lebih canggih, dan adaptasi terhadap berbagai kondisi lingkungan. Secara keseluruhan, penelitian ini memberikan dasar yang kuat untuk pengembangan lebih lanjut dalam bidang pengenalan warna menggunakan pendekatan KNN.

5. SARAN

Untuk penelitian lebih lanjut, disarankan untuk mempertimbangkan beberapa aspek yang dapat meningkatkan kualitas sistem pengenalan warna. Pertama, diperlukan penambahan dan diversifikasi data pelatihan agar model KNN dapat lebih baik menangani variasi warna dan kondisi gambar yang lebih kompleks. Kedua, eksplorasi metode ekstraksi fitur yang lebih canggih dapat meningkatkan kemampuan sistem dalam mengenali dan memahami karakteristik warna yang lebih detail. Selain itu, evaluasi lebih lanjut terhadap respons sistem terhadap perubahan kondisi pencahayaan dan latar belakang gambar dapat memberikan wawasan yang lebih mendalam. Terakhir, penelitian lebih lanjut sebaiknya mempertimbangkan integrasi dengan

teknologi pengolahan citra dan machine learning terbaru untuk meningkatkan performa dan adaptabilitas sistem dalam berbagai konteks.

UCAPAN TERIMA KASIH

Kami, sebagai penulis penelitian ini, ingin mengucapkan terima kasih kepada Bapak Novan Wijaya atas bantuan dan bimbingan yang berharga dalam proses penelitian sehingga penelitian ini dapat diselesaikan dengan baik dan mencapai hasil yang memuaskan.

DAFTAR PUSTAKA

- [1] 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.
- [2] S. Mujahidin, “Klasifikasi Warna Kulit berdasarkan Ruang Warna RGB,” *Semin. Nas. Apl. Teknol. Inf.*, vol. 1, no. 1, pp. 17–19, 2015, [Online]. Available: <http://journal.uui.ac.id/index.php/Snati/article/view/3530>
- [3] S. Saifullah, D. B. Prasetyo, Indahyani, R. Dreżewski, and F. A. Dwiyanto, “Palm Oil Maturity Classification Using K-Nearest Neighbors Based on RGB and L*a*b Color Extraction,” *Procedia Comput. Sci.*, 2023, doi: 10.1016/j.procs.2023.10.294.
- [4] K. E. Sakti, M. Mardiana, and R. A. Pradipta, “Rancang Bangun Aplikasi Web Pendeteksi Warna Pada Pixel Gambar Dengan Knn Classifier,” *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 2, 2023, doi: 10.23960/jitet.v11i2.3009.
- [5] F. Shidiq, “Penerapan Metode K-Nearest Neighbor (KNN) Untuk Menentukan Ikan Cupang Dengan Ekstraksi Fitur Ciri Bentuk Dan Canny,” *Innov. Res. Informatics*, vol. 3, no. 2, pp. 39–46, 2021, doi: 10.37058/innovatics.v3i2.3093.
- [6] N. Tamsir, “Aplikasi Citra Untuk Mendeteksi dan Mengenali Warna Menggunakan Algoritma Midpoint,” *e-Jurnal JUSITI (Jurnal Sist. Inf. dan Teknol. Informasi)*, vol. 9, no. 2, pp. 147–155, 2020, doi: 10.36774/jusiti.v9i2.769.
- [7] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. San Francisco, 2005.
- [8] Y. Zhou *et al.*, “Application of mathematical morphology operation with memristor-based computation-in-memory architecture for detecting manufacturing defects,” *Fundam. Res.*, vol. 2, no. 1, pp. 123–130, 2022, doi: 10.1016/j.fmre.2021.06.020.