

Ekstraksi Palet Warna Digital Menggunakan K-Means Clustering Berbasis Web

^{1*}Moh. Farhan Nashiruddin, ²Umi Mahdiyah, ³Wahyu Cahyo Utomo

¹ Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: ¹farhannash214@gmail.com, ²umimahdiyah@gmail.com,

³wahyu.utomo@unpkdr.ac.id

Penulis Korespondens : Moh. Farhan Nashiruddin

Abstrak— Pemilihan warna dalam desain digital sering kali bergantung pada intuisi pengguna, sehingga hasil palet warna yang diperoleh dapat kurang konsisten dan tidak merepresentasikan warna dominan dari gambar. Penelitian ini bertujuan mengembangkan sistem ekstraksi palet warna digital berbasis web menggunakan metode *K-Means Clustering*. Sistem dibangun dengan Streamlit dan Python untuk memproses citra berformat JPG, JPEG, atau PNG. Tahapan sistem meliputi unggah gambar, *resize* citra, ekstraksi piksel RGB, pengelompokan warna menggunakan *K-Means*, pemilihan warna berdasarkan *mood*, visualisasi kode warna HEX, penyuntingan warna secara manual, pembuatan *gradient theme*, serta ekspor hasil palet. Hasil implementasi menunjukkan bahwa sistem mampu menghasilkan palet warna dominan secara otomatis, interaktif, dan mudah digunakan. Sistem ini dapat membantu desainer dalam menentukan inspirasi warna secara lebih cepat, objektif, dan sesuai dengan karakter visual gambar.

Kata Kunci— *K-Means Clustering*, palet warna, pengolahan citra

Abstract— Colour selection in digital design often relies on the user's intuition, meaning that the resulting colour palettes may lack consistency and fail to represent the dominant colours of the image. This research aims to develop a web-based digital colour palette extraction system using the *K-Means Clustering* method. The system is built using Streamlit and Python to process images in JPG, JPEG, or PNG formats. The system's stages include image upload, image resizing, RGB pixel extraction, colour clustering using *K-Means*, colour selection based on mood, HEX colour code visualisation, manual colour editing, gradient theme creation, and palette export. The results of the implementation demonstrate that the system is capable of automatically generating dominant colour palettes in an interactive and user-friendly manner. This system can assist designers in determining colour inspiration more quickly, objectively, and in line with the visual character of the image.

Keywords— *K-Means Clustering*, colour palette, image processing

This is an open access article under the CC BY-SA License.



I. PENDAHULUAN

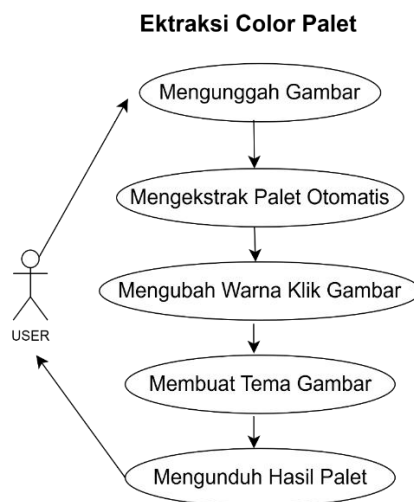
Warna merupakan elemen krusial yang melampaui sekadar sensasi visual, karena memiliki kekuatan yang memengaruhi emosi, pikiran, dan persepsi manusia secara mendalam [1]. Dalam perancangan desain, palet warna yakni sekumpulan sampel yang merepresentasikan atau menggambarkan warna utama dalam sebuah karya seni digunakan sebagai standar acuan untuk menciptakan identitas visual yang kohesif [2]. Pembuatan palet secara tradisional sering kali bergantung pada intuisi perancang, namun seiring kemajuan teknologi, ekstraksi palet warna kini dapat diotomatisasi dengan mengambil inspirasi warna langsung dari sebuah gambar atau foto digital [3].

Secara teknis, citra digital tersebut tersusun atas grid dua dimensi yang terdiri dari elemen terkecil bernama piksel, di mana masing-masing piksel menyimpan informasi spesifik mengenai intensitas cahaya dan komponen warna [4]. Variasi data warna yang sangat banyak pada piksel ini kemudian dapat dianalisis dan dikompresi dengan cara menyaring variasinya untuk menemukan sekumpulan warna yang paling dominan [5]. Untuk mencapai hal tersebut secara otomatis dan objektif, diperlukan metodologi dan teknik klasterisasi yang mampu memproses serta mengelompokkan data warna gambar ke dalam ruang warna yang terstruktur [6].

Salah satu algoritma klasterisasi *unsupervised* (tanpa pengawasan) yang paling populer dan tangguh untuk tugas ini adalah *K-Means Clustering* [7]. Algoritma ini bekerja secara iteratif dengan membagi titik-titik data warna ke dalam sejumlah klaster berdasarkan penghitungan jarak *Euclidean* terdekat, lalu menentukan sebuah titik pusat (*centroid*) yang secara efektif mewakili warna representatif dari masing-masing klaster tersebut [8]. Melalui integrasi algoritma K-Means, proses penentuan skema warna dapat dilakukan secara komputasional dan objektif, sehingga mampu menghasilkan paradigma atau palet warna yang presisi dan harmonis untuk diterapkan secara langsung pada karya grafis [9], [10].

II. METODE

Penelitian ini menggunakan pendekatan komputasi berbasis *unsupervised learning* untuk analisis dan pemrosesan citra digital. Proses penelitian dilakukan melalui beberapa tahapan, yaitu penyiapan dataset citra desain grafis, pembacaan dan ekstraksi matriks piksel warna, inisialisasi jumlah klaster warna (K), penerapan algoritma *K-Means Clustering*, serta tahap ekstraksi nilai *centroid* untuk visualisasi hasil akhir. Model yang dikembangkan diarahkan untuk mengklasifikasikan dan mengelompokkan jutaan variasi piksel warna pada sebuah karya desain grafis ke dalam beberapa kelompok warna dominan, guna menghasilkan palet warna digital yang akurat dan representatif.



Gambar 1. Alur Proses Sistem

A. Dataset Penelitian

Dataset dalam penelitian ini menggunakan data sekunder berupa citra desain grafis digital statis berformat JPG, JPEG, atau PNG yang bersumber dari platform portofolio profesional seperti Pinterest dan Behance. Pemilihan sumber data ini bertujuan untuk memperoleh sampel

visual dengan standar estetika industri dan komposisi warna yang bervariasi. Keberagaman sampel ini sangat penting untuk menguji keandalan algoritma dalam mengekstraksi skema warna pada berbagai tingkat kompleksitas, mulai dari desain minimalis hingga gradasi yang rumit. Untuk merepresentasikan implementasi desain di dunia nyata, dataset ini diklasifikasikan ke dalam tiga kategori utama, yaitu Desain Poster, Desain Produk, dan Desain Antarmuka Pengguna (UI Design).

Tabel 1. Contoh Dataset Desain Grafis

Tema	Keterangan	Contoh Desain
Desain Poster	Desain dengan tema Hari Guru Nasional	
Desain Produk	Desain produk minuman dengan warna yang <i>soft</i>	
UI Desain	Desain tampilan UI (<i>User Interface</i>)	

B. Pra-Pemrosesan

Pengecilan resolusi citra dengan batas lebar maksimal 150 piksel dilakukan pada tahap pra-pemrosesan awal untuk memangkas beban komputasi secara signifikan tanpa mengurangi akurasi warna aslinya. Proses optimasi ini menggunakan metode interpolasi *Bilinear* yang mengunci lebar gambar di angka 150 piksel, sambil otomatis menyesuaikan tingginya agar bentuk desain tetap proporsional. Langkah ini sangat krusial karena mampu mengeliminasi jutaan data piksel redundan dari gambar beresolusi tinggi, sehingga algoritma *K-Means* dapat bekerja menganalisis matriks warna dengan sangat ringan, efisien, dan instan (*real-time*) dengan proporsi warna dominan yang tetap terjaga utuh.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Gambar 2. Matriks Warna

III. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

Implementasi sistem dilakukan menggunakan bahasa pemrograman Python dengan *framework* Streamlit. *Library* yang digunakan meliputi NumPy untuk pengolahan matriks piksel, PIL untuk pembacaan dan pemrosesan citra, Scikit-Learn untuk penerapan algoritma *K-Means Clustering*, Colorsys untuk perhitungan karakter warna, serta *Streamlit Image Coordinates* untuk mendukung proses pemilihan warna secara interaktif pada citra.

Alur implementasi dimulai dari pengguna mengunggah citra berformat JPG, JPEG, atau PNG. Citra tersebut kemudian diubah ke mode RGB, diperkecil dengan batas lebar 150 piksel, dan dikonversi menjadi matriks piksel. Matriks piksel selanjutnya diproses menggunakan *K-Means* untuk membentuk sejumlah kandidat warna. Kandidat warna tersebut dinilai berdasarkan pilihan *color mood*, kemudian dipilih menjadi palet akhir dalam bentuk kode HEX dan persentase dominansi warna.

Implementasi proses ekstraksi warna pada PaletteFlow

```

1  def extract_smart_colors(image, mood='None', n_target=5):
2      max_width = 150
3      ratio = max_width / float(image.size[0])
4      h_size = int(float(image.size[1]) * float(ratio))
5      small_img = image.resize((max_width, h_size), Image.Resampling.BILINEAR)
6      img_array = np.array(small_img)
7      pixels = img_array.reshape(-1, 3)
8      kmeans = KMeans(n_clusters=max(10, n_target * 3), random_state=42, n_init=3)
9      kmeans.fit(pixels)
10     candidates_rgb = kmeans.cluster_centers_
11     scored_candidates = [{'rgb': c, 'score': calculate_mood_score(c, mood)} for c in candidates_rgb]
12     final_hex = [rgb_to_hex(c) for c in final_picks[:n_target]]
13     percentages = calculate_dominance(img_array, final_hex)
14     return final_hex, percentages

```

Gambar 3. Implementasi Proses Ekstraksi Warna Pada PaletteFlow

Pada Gambar 3 ditunjukkan bagian utama proses ekstraksi warna pada PaletteFlow. Nilai *n_clusters* ditentukan secara adaptif dengan rumus $\max(10, n_target \times 3)$ agar jumlah kandidat warna yang dianalisis lebih banyak dibandingkan jumlah warna akhir yang diminta pengguna. Setelah proses klusterisasi selesai, sistem mengubah nilai *centroid* RGB menjadi kode HEX, menghitung dominansi warna, dan menampilkan hasilnya dalam bentuk *strip* palet yang mudah digunakan pada kebutuhan desain.

B. Hasil Pengujian Antar Skenario

Pengujian antar skenario dilakukan untuk mengetahui kemampuan sistem dalam menghasilkan palet warna pada beberapa karakter citra dan pilihan *mood* yang berbeda. Skenario disusun berdasarkan variasi jenis citra, pilihan *mood*, dan jumlah warna yang diminta pengguna. Bentuk pengujian ini disesuaikan dengan karakter aplikasi PaletteFlow yang berfokus pada ekstraksi warna dominan, bukan klasifikasi citra berlabel.

Tabel 2. Hasil Pengujian Antar Skenario

Skenario	Karakteristik Pengujian	Parameter	Hasil
1	Desain poster dengan warna kontras	<i>Colorful</i> , 5 warna	Palet dominan dan kode HEX tampil
2	Desain poster dengan komposisi lembut	<i>Muted</i> , 5 warna	Palet cenderung lembut dan tidak terlalu jenuh
3	Desain produk dengan warna terang	<i>Bright</i> , 5 warna	Warna terang lebih diprioritaskan
4	Desain antarmuka pengguna sederhana	<i>None</i> , 6 warna	Palet netral mengikuti dominansi piksel
5	Citra bernuansa gelap	<i>Dark</i> , 5 warna	Warna gelap lebih dominan pada hasil akhir

Berdasarkan Tabel 2, seluruh skenario pengujian dapat menghasilkan palet warna sesuai parameter yang diberikan. Perbedaan pilihan *mood* berpengaruh terhadap pemilihan kandidat *centroid* yang ditampilkan sebagai warna akhir. Pada mode *Colorful*, sistem cenderung memilih warna dengan saturasi lebih kuat, sedangkan pada mode *Muted* sistem memilih warna yang lebih rendah saturasinya. Hal ini menunjukkan bahwa sistem tidak hanya mengambil warna dominan secara umum, tetapi juga menyesuaikan hasil dengan kebutuhan visual pengguna.

C. Hasil Evaluasi Per Kategori Citra

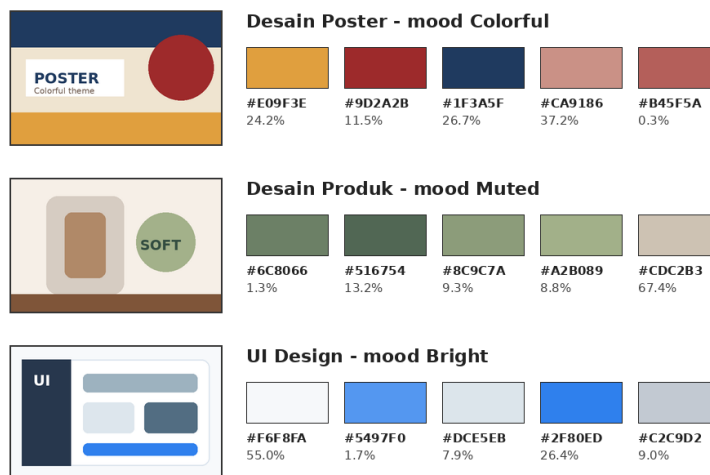
Evaluasi per kategori citra dilakukan untuk mengetahui kesesuaian hasil ekstraksi pada beberapa jenis desain grafis. Kategori yang digunakan meliputi desain poster, desain produk, dan desain antarmuka pengguna. Setiap kategori memiliki karakter visual berbeda, sehingga hasil ekstraksi dapat menunjukkan kemampuan sistem dalam menangani variasi warna dan komposisi gambar.

Tabel 3. Hasil Evaluasi Per Kategori Citra

Kategori Citra	Karakter Visual	Hasil Ekstraksi	Keterangan
Desain Poster	Warna kontras dan area visual kuat	Palet menampilkan warna utama dari latar, teks, dan objek	Sesuai untuk kebutuhan identitas visual poster
Desain Produk	Warna lembut dan komposisi minimalis	Palet cenderung menghasilkan warna <i>soft</i> dan netral	Sesuai untuk desain produk dengan nuansa sederhana
<i>UI Design</i>	Blok warna antarmuka dan elemen tombol	Palet menampilkan warna latar, panel, dan aksen	Sesuai untuk penyusunan referensi warna antarmuka

Berdasarkan Tabel 3, sistem mampu menghasilkan palet yang relevan pada setiap kategori citra. Pada desain poster, warna dominan yang dihasilkan memperlihatkan elemen warna utama sehingga dapat digunakan sebagai acuan desain lanjutan. Pada desain produk, hasil ekstraksi cenderung menampilkan warna lembut sesuai karakter visual produk. Pada *UI Design*, sistem mampu menangkap warna latar dan aksent sehingga palet dapat membantu penyusunan gaya visual antarmuka.

Perbandingan hasil ekstraksi pada tiga kategori citra

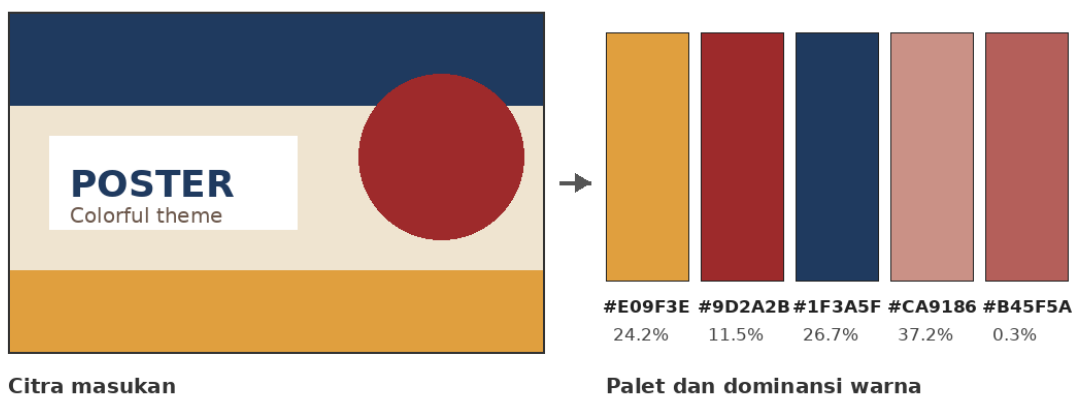


Gambar 4. Perbandingan Hasil Ekstraksi Pada Beberapa Kategori Citra

D. Analisis Dominasi Warna

Analisis dominasi warna dilakukan dengan menghitung proporsi piksel yang paling dekat terhadap setiap warna palet akhir. Perhitungan ini menggunakan jarak antarwarna dalam ruang RGB, sehingga setiap piksel pada citra dapat dikelompokkan ke warna palet yang paling sesuai. Persentase dominansi warna menjadi informasi tambahan yang penting karena pengguna dapat mengetahui warna yang paling banyak muncul pada citra.

Hasil ekstraksi palet warna



Gambar 5. Hasil Ekstraksi Palet Warna Dan Dominansi Warna

Gambar 5 menunjukkan contoh hasil ekstraksi palet warna pada citra poster. Palet yang terbentuk terdiri dari warna #E09F3E dengan dominansi 24,2%, #9D2A2B sebesar 11,5%, #1F3A5F sebesar 26,7%, #CA9186 sebesar 37,2%, dan #B45F5A sebesar 0,3%. Nilai tersebut menunjukkan bahwa warna #CA9186 menjadi warna dengan dominansi tertinggi pada contoh citra, sedangkan #B45F5A hanya muncul dalam proporsi kecil. Informasi dominansi ini membantu pengguna membedakan warna utama dan warna aksen pada desain.

E. Analisis Pengaruh *Color Mood*

Fitur *color mood* digunakan untuk mengarahkan pemilihan warna hasil *K-Means* agar lebih sesuai dengan kebutuhan desain. Sistem menyediakan beberapa pilihan *mood*, yaitu *Colorful*, *Bright*, *Muted*, *Deep*, *Dark*, dan *None*. Masing-masing *mood* memiliki cara penilaian yang berbeda berdasarkan nilai saturasi dan kecerahan warna. Dengan demikian, warna yang dipilih bukan hanya *centroid* dengan jumlah piksel besar, tetapi juga warna yang memenuhi karakter visual tertentu.

Tabel 4. Pengaruh *Color Mood* Terhadap Hasil Palet

<i>Color Mood</i>	Prioritas Warna	Kegunaan
<i>Colorful</i>	Saturasi tinggi dan warna lebih hidup	Cocok untuk poster atau desain promosi
<i>Bright</i>	Warna terang dengan tingkat kecerahan tinggi	Cocok untuk desain bersih dan ringan
<i>Muted</i>	Saturasi rendah dan warna lembut	Cocok untuk produk minimalis atau tema natural
<i>Deep</i>	Warna pekat dengan karakter kuat	Cocok untuk desain elegan dan tegas
<i>Dark</i>	Warna gelap atau rendah kecerahan	Cocok untuk tema gelap dan visual dramatis
<i>None</i>	Pemilihan warna tanpa penekanan <i>mood</i>	Cocok untuk melihat dominansi warna asli

Berdasarkan Tabel 4, pemilihan *color mood* menjadi faktor yang memengaruhi hasil akhir palet. Mode *None* digunakan ketika pengguna ingin melihat warna yang paling merepresentasikan citra secara umum. Sementara itu, mode lainnya digunakan ketika pengguna ingin mendapatkan palet dengan karakter tertentu, misalnya lebih cerah, lebih lembut, atau lebih gelap. Fitur ini membuat PaletteFlow lebih fleksibel karena dapat menyesuaikan hasil ekstraksi dengan kebutuhan konsep desain.

F. Hasil Pengujian Fungsional Sistem

Pengujian fungsional dilakukan untuk memastikan bahwa setiap fitur utama pada aplikasi berjalan sesuai rancangan. Pengujian ini berfokus pada proses interaksi pengguna, mulai dari unggah gambar sampai ekspor hasil tema warna. Hasil pengujian ditampilkan pada Tabel 5.

Tabel 5. Hasil Pengujian Fungsional Sistem

Fitur	Skenario Uji	Output yang Diharapkan	Status
<i>Upload Image</i>	Pengguna mengunggah citra JPG, JPEG, atau PNG	Citra tampil pada halaman utama	Berhasil
<i>Color Mood</i>	Pengguna memilih salah satu <i>mood</i> warna	Palet berubah mengikuti karakter <i>mood</i>	Berhasil
<i>Slider Colors</i>	Pengguna memilih jumlah warna 2 sampai 10	Jumlah <i>strip</i> palet sesuai <i>input</i>	Berhasil
Ekstraksi Otomatis	Sistem memproses citra dengan <i>K-Means</i>	Kode HEX dan dominansi warna tampil	Berhasil
Edit Warna	Pengguna memilih <i>slot</i> dan klik area citra	Warna <i>slot</i> aktif berubah sesuai piksel	Berhasil
<i>Gradient Theme</i>	Sistem menyusun palet menjadi gradasi	Gradient dan kode CSS tampil	Berhasil
<i>Download Theme</i>	Pengguna menekan tombol unduh	File JPG hasil tema dapat disimpan	Berhasil

Berdasarkan Tabel 5, seluruh fitur utama sistem berhasil dijalankan. Fitur unggah gambar menjadi tahap awal yang menentukan masukan sistem, sedangkan fitur *mood* dan slider digunakan untuk mengatur karakter serta jumlah warna yang dihasilkan. Fitur edit warna memberi keleluasaan kepada pengguna untuk memperbaiki palet secara manual, dan fitur *download theme* menjadi keluaran akhir yang dapat digunakan kembali dalam proses desain.

G. Hasil Ekspor Tema

Hasil ekspor tema merupakan keluaran akhir dari aplikasi PaletteFlow. Pada tahap ini, sistem menggabungkan citra masukan, palet warna, kode HEX, serta *gradient theme* ke dalam satu tampilan. Fitur ekspor ini diperlukan agar hasil ekstraksi tidak berhenti pada tampilan aplikasi, tetapi dapat disimpan dan digunakan kembali sebagai referensi desain.



Gambar 6. Hasil Ekspor Tema Warna Dan *Gradient Theme*

Gambar 6 menunjukkan contoh hasil ekspor tema yang dihasilkan oleh sistem. Bagian utama menampilkan citra desain, sedangkan bagian samping memperlihatkan daftar kode HEX dari palet yang terbentuk. Pada bagian bawah terdapat *gradient theme* yang disusun dari seluruh warna palet. Hasil ini memperlihatkan bahwa aplikasi mampu memberikan keluaran yang praktis karena pengguna dapat langsung memperoleh inspirasi warna, kode warna, dan gambaran gradasi dalam satu file.

H. Pembahasan

Hasil penelitian menunjukkan bahwa metode *K-Means Clustering* dapat digunakan untuk mengekstraksi palet warna dominan dari citra desain grafis. Sistem mampu mengelompokkan piksel warna ke dalam beberapa klaster, kemudian menampilkan *centroid* sebagai warna representatif. Pendekatan ini sesuai untuk aplikasi PaletteFlow karena data warna pada citra tidak memerlukan label kelas seperti pada klasifikasi citra, melainkan cukup dikelompokkan berdasarkan kemiripan nilai warna.

Pemilihan parameter jumlah warna berpengaruh terhadap keragaman palet yang dihasilkan. Jumlah warna yang sedikit menghasilkan palet yang lebih sederhana, sedangkan jumlah warna yang lebih banyak dapat menangkap variasi warna yang lebih detail. Selain itu, penggunaan *color mood* memberikan nilai tambah karena hasil palet dapat diarahkan sesuai kebutuhan desain. Hal ini membuat sistem lebih adaptif dibanding ekstraksi warna yang hanya mengambil warna dominan tanpa mempertimbangkan karakter visual.

Secara keseluruhan, PaletteFlow mampu menjalankan proses ekstraksi, visualisasi, pengubahan manual, dan ekspor tema warna dalam satu alur sistem. Hasil pengujian fungsional menunjukkan bahwa fitur utama berjalan sesuai rancangan. Keterbatasan sistem terdapat pada ketergantungan terhadap kualitas citra masukan, karena pencahayaan, resolusi, dan komposisi objek dapat memengaruhi palet akhir. Oleh karena itu, penggunaan citra dengan kualitas baik tetap diperlukan agar hasil ekstraksi lebih representatif.

IV. KESIMPULAN

Penelitian ini menghasilkan aplikasi PaletteFlow untuk ekstraksi palet warna digital menggunakan metode *K-Means Clustering*. Sistem mampu memproses citra desain grafis, mengelompokkan piksel warna, menghasilkan kode HEX, menampilkan persentase dominansi warna, serta menyediakan fitur *color mood*, edit warna manual, *gradient theme*, dan ekspor tema. Berdasarkan hasil pengujian, seluruh fitur utama berjalan sesuai rancangan dan mampu menghasilkan palet warna yang relevan pada desain poster, desain produk, dan desain antarmuka pengguna. Dengan demikian, PaletteFlow dapat digunakan sebagai alat bantu dalam menentukan referensi warna desain secara lebih cepat, objektif, dan praktis.

DAFTAR PUSTAKA

- [1] A. Yadav and P. Singh, "Penyempurnaan Pembuatan Skema Warna: Klasterisasi K-Means Iteratif dan Evaluasi ARI Terjemahan Bahasa Indonesia dari artikel: Refining Color Scheme Generation: Iterative K-Means Clustering and ARI Evaluation," *Halaman 1 Journal of Informatics Electrical and Electronics Engineering*, vol. 05, no. 112, pp. 1–12, 2024, doi: 10.54060/a2zjournals.jieec.112.
- [2] Z. Zhang, Z. Zhang, and Z. Xia, "The style correlation between color palette generated by the k-means and original picture," in *Ergonomics In Design*, AHFE International, 2022. doi: 10.54941/ahfe1001956.

- [3] Atomic Object LLC, "Palette Maker-A Tool for Extracting Color Palettes From Images," 2026. Accessed: Jun. 05, 2026. [Online]. Available: <https://github.com/mattne>
- [4] P. Nabilla, M. Farhan Saputra, and R. A. Saputra, "PERBANDINGAN RUANG WARNA RGB, HSV DAN YCBCR UNTUK SEGMENTASI CITRA IKAN KEMBUNG MENGGUNAKAN K-MEANS CLUSTERING," 2022.
- [5] A. Ma'ruf, H. Sibyan, and N. Mardiyantoro, "Terjemahan Artikel ke Bahasa Indonesia Ekstraksi Palet Warna untuk Kompresi Gambar Digital menggunakan Algoritma K-Means," 2025.
- [6] Y. Kynash and M. Semeniv, "New Approach to Dominant and Prominent Color Extraction in Images with a Wide Range of Hues," *Technologies (Basel)*, vol. 13, no. 6, Jun. 2025, doi: 10.3390/technologies13060230.
- [7] A. Cheng, E. Rosenberg, and A. Gorbunova, "K-MEANS WITH SAMPLING FOR DETERMINING PROMINENT COLORS IN IMAGES," *ICTACT Journal on Soft Computing*, vol. 13, no. 1, pp. 2813–2819, Oct. 2022, doi: 10.21917/ijsc.2022.0400.
- [8] Aryo De Wibowo Muhammad Sidik *et al.*, "Menerapkan K-Means Clustering untuk Segmentasi Gambar Database Berwarna," *Fidelity: Jurnal Teknik Elektro*, vol. 2, pp. 57–61, 2020.
- [9] Rizka Rizka, S. Nasution, F. Aulia, and Supiyandi Supiyandi, "Penerapan Metode Segmentasi Warna HSV untuk Deteksi Objek Berbasis Warna pada Citra Digital," *Router : Jurnal Teknik Informatika dan Terapan*, vol. 3, no. 4, pp. 11–24, Dec. 2025, doi: 10.62951/router.v3i4.706.
- [10] M. Sun and J. Wu, "Research on Picture Book Color Design for Children with Autism Based on K-Means Clustering Algorithm," in *Proceedings of the 2025 5th International Conference on Internet of Things and Machine Learning, IoTML 2025*, Association for Computing Machinery, Inc, Aug. 2025, pp. 130–137. doi: 10.1145/3749566.3749596.