

Implementasi Algoritma DCT (Discrete Cosine Transform) dan K-Means Clustering untuk Kuantisasi Warna pada Konversi Citra JPG ke Format GIF

Sumita Wardani^{1*}, Andi Zulherry², Karina Andriani³, Ichsan Firmansyah⁴

¹ Fakultas Teknik, Teknik Elektro, Universitas Al-Azhar, Medan, Indonesia

² Fakultas Ilmu Komputer dan Teknologi Informasi, Sains Data, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia

³ Manajemen Informatika, STMIK Triguna Dharma, Medan, Indonesia

⁴ Sistem Informasi, STMIK Triguna Dharma, Medan, Indonesia

INFORMASI ARTIKEL

Diterima Redaksi: 30 Juni 2026

Revisi Akhir: 10 Juli 2026

Diterbitkan Online: 22 Juli 2026

KATA KUNCI

Discrete Cosine Transform

K-Means Clustering

Kompresi Citra

Kuantisasi Warna

Konversi JPG ke GIF

KORESPONDENSI (*)

Phone: +62 823-7965-4108

E-mail: sumita.wardani88@gmail.com

A B S T R A K

Meningkatnya kebutuhan akan penyimpanan dan transmisi citra digital yang efisien mendorong penelitian di bidang kompresi citra dan teknik kuantisasi warna. Penelitian ini bertujuan untuk mengimplementasikan algoritma Discrete Cosine Transform (DCT) yang dikombinasikan dengan K-Means Clustering untuk kuantisasi warna dalam proses konversi citra JPG ke format GIF. Algoritma DCT digunakan untuk mereduksi komponen frekuensi tinggi pada citra guna meningkatkan efisiensi data, sementara algoritma K-Means Clustering diterapkan untuk menghasilkan palet warna yang optimal sesuai dengan batasan format GIF, yaitu maksimal 256 warna. Penelitian ini menggunakan pendekatan eksperimen kuantitatif dan mengimplementasikan sistem menggunakan bahasa pemrograman Python pada platform Google Colab. Parameter evaluasi yang digunakan meliputi Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), dan ukuran file. Hasil penelitian menunjukkan bahwa proses DCT berhasil mengurangi ukuran file dari 1347,83 KB menjadi 1198,46 KB, dengan nilai MSE sebesar 42,3175 dan nilai PSNR sebesar 31,8692 dB. Proses kuantisasi K-Means dengan 256 kluster menghasilkan nilai MSE sebesar 47,6381 dan nilai PSNR sebesar 31,3504 dB, dengan ukuran file sebesar 5124,70 KB. Eksperimen lanjutan dengan jumlah kluster yang berbeda menunjukkan bahwa pengurangan jumlah kluster menurunkan ukuran file namun juga menurunkan kualitas citra; GIF dengan 8 kluster menghasilkan ukuran file terkecil, yaitu 684,35 KB, dengan nilai PSNR sebesar 25,9143 dB. Temuan ini mengindikasikan bahwa kombinasi algoritma DCT dan K-Means dapat secara efektif mengkonversi citra JPG ke format GIF, meskipun terdapat pertukaran (trade-off) antara efisiensi ukuran file dan kualitas visual citra.

PENDAHULUAN

Kemajuan pesat teknologi informasi dan komunikasi telah meningkatkan secara signifikan permintaan terhadap pengolahan citra digital di berbagai bidang. Citra digital digunakan secara luas dalam multimedia, desain grafis, kecerdasan buatan, aplikasi berbasis web, dan platform media sosial. Seiring dengan bertambahnya volume citra digital, kebutuhan akan penyimpanan dan transmisi yang efisien menjadi semakin penting, sehingga kompresi citra merupakan komponen kritis dalam pemrosesan citra modern [2]. JPG (JPEG) tetap menjadi salah satu format citra yang paling banyak digunakan, menggunakan teknik kompresi lossy berbasis Discrete Cosine Transform (DCT) yang mengurangi ukuran file sambil mempertahankan kualitas visual yang relatif baik [1], [10]. Transformasi ini mengkonversi domain spasial citra ke domain frekuensi, memungkinkan komponen frekuensi tinggi dikurangi secara selektif dengan dampak persepsi yang minimal bagi pengguna [1].

Sebaliknya, GIF (Graphics Interchange Format) mengandalkan warna terindeks dengan batas ketat 256 warna, yang sering mengakibatkan penurunan kualitas visual ketika citra RGB berompleksitas warna tinggi dikonversi ke format ini, khususnya pada area gradien halus dan detail tekstural yang rinci [5]. Keterbatasan ini menuntut teknik kuantisasi warna yang efektif untuk menghasilkan palet warna yang optimal. Salah satu pendekatan yang sudah mapan adalah algoritma K-Means Clustering, yang mempartisi ruang warna RGB dengan mengelompokkan piksel secara iteratif ke dalam sejumlah klaster yang telah ditentukan berdasarkan kesamaan warna, meminimalkan variansi intra-klaster untuk menghasilkan palet warna yang kompak dan representatif [3], [4].

Beberapa penelitian terdahulu telah mengkaji kompresi citra JPG berbasis DCT dan kuantisasi warna untuk konversi ke format GIF secara terpisah. Amanati dkk. menerapkan DCT untuk kompresi citra digital guna meningkatkan efisiensi penyimpanan, sementara Kusuma dan Lestari mengevaluasi kualitas kompresi citra menggunakan metrik PSNR dan MSE pada berbagai algoritma transformasi [1], [2]. Di sisi lain, penelitian mengenai kuantisasi warna untuk format GIF juga telah banyak dilakukan; Putra dan Santoso menerapkan K-Means Clustering untuk kuantisasi warna citra digital berformat GIF, Gifani mengusulkan perbaikan algoritma K-Means untuk efisiensi encoding GIF, sementara Halim dan Wahono melakukan studi komparatif berbagai teknik kuantisasi warna dalam pembuatan GIF [4], [6], [7]. Penelitian lain oleh Sundari dan Riyadi turut mengkaji pengaruh jumlah warna terhadap kualitas hasil konversi GIF, serta Rahmadi dan Nugroho serta Wibowo dan Santoso menganalisis karakteristik format JPG dan GIF pada aplikasi multimedia dan sistem penyimpanan modern [5], [8], [9].

Studi-studi tersebut umumnya membahas algoritma DCT untuk kompresi JPG dan algoritma K-Means untuk kuantisasi warna GIF sebagai dua topik yang berdiri sendiri, tanpa mengintegrasikan keduanya dalam satu alur proses konversi JPG ke GIF yang utuh, serta belum banyak mengevaluasi secara sistematis trade-off antara ukuran file dan kualitas citra pada berbagai jumlah klaster warna. Kebaruan (novelty) penelitian ini terletak pada integrasi algoritma DCT untuk kompresi citra JPG dengan algoritma K-Means Clustering untuk kuantisasi warna dalam satu pipeline konversi JPG ke GIF, disertai eksperimen sistematis pada variasi jumlah klaster (256 hingga 8 klaster) yang dievaluasi secara kuantitatif menggunakan metrik MSE, PSNR, dan ukuran file secara bersamaan, sehingga menghasilkan gambaran trade-off yang lebih komprehensif dibandingkan penelitian-penelitian sebelumnya yang mengkaji kedua aspek tersebut secara terpisah.

Berdasarkan latar belakang tersebut, penelitian ini berfokus pada penerapan algoritma DCT dalam pemrosesan citra JPG dan pemanfaatan K-Means Clustering untuk pembentukan palet dalam konversi ke format GIF, dengan tujuan menghasilkan proses konversi yang lebih optimal dalam hal kualitas citra maupun ukuran file. Ruang lingkup penelitian dibatasi pada citra JPG sebagai format masukan, mengingat JPG merupakan jenis citra RGB yang paling umum digunakan dalam sistem digital, sementara keluarannya dibatasi pada format GIF dengan maksimal 256 warna sesuai karakteristik warna terindeks. Algoritma K-Means Clustering diterapkan dalam bentuk standar tanpa modifikasi, dan analisis difokuskan pada kualitas citra dan ukuran file, bukan waktu komputasi. Sistem diimplementasikan menggunakan bahasa pemrograman Python.

Permasalahan yang dibahas dalam penelitian ini dapat dirumuskan sebagai berikut: bagaimana algoritma DCT dapat diterapkan dalam pemrosesan citra JPG, bagaimana K-Means Clustering dapat diterapkan dalam pembentukan palet warna GIF, dan bagaimana kombinasi kedua algoritma tersebut memengaruhi kualitas citra dan ukuran file selama proses konversi JPG ke GIF. Tujuan penelitian ini adalah mengimplementasikan algoritma DCT untuk kompresi citra JPG dan menerapkan K-Means Clustering untuk kuantisasi warna menuju format GIF. Penelitian ini diharapkan dapat memperkaya kajian akademik di bidang pengolahan citra digital, khususnya mengenai penerapan DCT dan K-Means Clustering dalam konversi JPG ke GIF, serta menjadi referensi bagi peneliti selanjutnya.

TINJAUAN PUSTAKA

Discrete Cosine Transform (DCT)

DCT merupakan teknik transformasi matematika yang banyak digunakan dalam kompresi citra digital. DCT mengkonversi sinyal dari domain spasial ke domain frekuensi, memungkinkan pemisahan informasi penting dan tidak penting. Pada proses kompresi, koefisien frekuensi tinggi yang kurang signifikan secara persepsi dihilangkan, sehingga menghasilkan ukuran file yang lebih kecil [1]. DCT diterapkan pada blok piksel berukuran 8×8 , yang merupakan dasar

dari standar kompresi JPEG [6]. Secara matematis, transformasi DCT dua dimensi untuk blok piksel berukuran 8×8 dirumuskan sebagai berikut:

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right]$$

di mana $f(x,y)$ menyatakan nilai intensitas piksel pada koordinat spasial (x,y) , dan $F(u,v)$ menyatakan koefisien DCT pada domain frekuensi (u,v) .

K-Means Clustering

K-Means Clustering adalah algoritma pengelompokan berbasis jarak yang bekerja dengan cara menginisialisasi sejumlah k pusat kluster secara acak, kemudian secara iteratif mengalokasikan setiap piksel ke kluster terdekat dan memperbarui pusat kluster hingga konvergensi tercapai. Dalam konteks kuantisasi warna, setiap piksel citra direpresentasikan sebagai titik dalam ruang warna RGB, dan K-Means mengelompokkannya ke dalam k kluster yang mewakili warna-warna dominan pada citra [3], [4].

$$J = \sum_{i=1}^k \sum_{x \in C_i} |x - \mu_i|^2$$

Di mana k menyatakan jumlah kluster, C_i menyatakan himpunan titik data (piksel) pada kluster ke- i , x menyatakan vektor warna piksel dalam ruang RGB, dan μ_i menyatakan titik pusat (*centroid*) kluster ke- i . Proses iteratif K-Means bertujuan meminimalkan nilai J dengan memperbarui μ_i pada setiap iterasi hingga konvergen.

Format GIF dan Kuantisasi Warna

Format GIF menggunakan mode warna terindeks yang membatasi jumlah warna hingga 256 warna per frame. Oleh karena itu, citra RGB yang memiliki jutaan warna harus dikuantisasi terlebih dahulu sebelum disimpan dalam format GIF. Kualitas hasil konversi sangat bergantung pada algoritma kuantisasi warna yang digunakan [5], [7]. Semakin sedikit jumlah warna yang digunakan, semakin kecil ukuran file yang dihasilkan, namun kualitas visual citra juga akan menurun [8].

METODOLOGI

Penelitian ini mengadopsi pendekatan eksperimen kuantitatif, karena keseluruhan proses dirancang untuk menguji langsung kinerja algoritma DCT yang diterapkan pada konversi citra JPG ke format GIF, dikombinasikan dengan K-Means Clustering sebagai teknik kuantisasi warna. Pendekatan ini dipilih karena proses kompresi citra melibatkan sejumlah parameter yang dapat diukur, seperti ukuran file, tingkat distorsi, kualitas visual, dan efisiensi komputasi, yang semuanya dapat dievaluasi melalui metrik objektif.

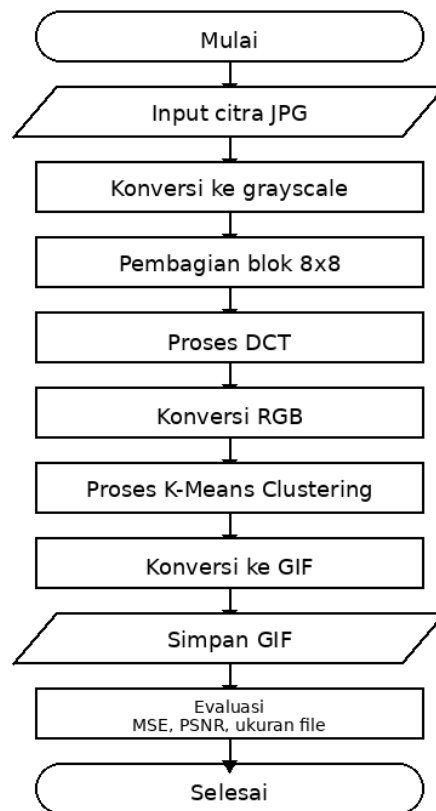
Perangkat Penelitian

Perangkat keras yang digunakan dalam penelitian ini berupa laptop dengan prosesor Intel Core i7, RAM 16 GB, dan layar 15,6 inci. Lingkungan perangkat lunak mencakup sistem operasi Windows 11, Python versi 3.10 atau yang lebih baru, pustaka Pillow, NumPy, dan Scikit-learn, serta Google Colab sebagai lingkungan pengembangan. Data penelitian terdiri dari beberapa file citra JPG dengan ukuran dan resolusi yang bervariasi, yang digunakan sebagai materi uji untuk proses konversi ke format GIF.

Perancangan Sistem

Sistem dalam penelitian ini dirancang untuk melakukan konversi citra JPG ke format GIF melalui penerapan kombinasi algoritma DCT dan K-Means Clustering, disusun sedemikian rupa sehingga setiap tahap pemrosesan dapat diamati, dianalisis, dan dievaluasi secara komprehensif. Alur proses keseluruhan dimulai saat pengguna mengunggah citra JPG ke

dalam sistem. Citra kemudian dikonversi ke grayscale, dibagi menjadi blok 8×8 piksel, diproses menggunakan DCT, dan koefisien frekuensi tingginya direduksi. Blok yang telah dimodifikasi direkonstruksi menggunakan IDCT, kemudian dikonversi kembali ke RGB untuk proses kuantisasi warna menggunakan K-Means Clustering. Setelah kuantisasi, citra disimpan dalam format GIF dan sistem melakukan evaluasi akhir berupa perhitungan MSE, PSNR, dan perbandingan ukuran file.



Gambar 1. Bagan Alir Sistem Konversi JPG ke GIF

Implementasi Sistem

Sistem diimplementasikan menggunakan bahasa pemrograman Python pada platform Google Colab. Pustaka yang digunakan meliputi NumPy untuk operasi matriks, SciPy untuk implementasi DCT dan IDCT, Pillow untuk manipulasi citra, Matplotlib untuk visualisasi hasil, Scikit-learn untuk K-Means Clustering, dan Scikit-image untuk komputasi metrik evaluasi MSE dan PSNR.

Pada tahap implementasi DCT, citra terlebih dahulu dikonversi ke grayscale, kemudian dibagi menjadi blok 8×8 piksel. Setiap blok diproses menggunakan DCT untuk mentransformasikan representasi citra dari domain spasial ke domain frekuensi. Koefisien frekuensi tinggi kemudian direduksi dengan menetapkan sebagian nilainya menjadi nol, mensimulasikan proses kompresi dengan menghilangkan detail yang kurang signifikan. Pada tahap implementasi K-Means Clustering, citra hasil DCT dikonversi kembali ke format RGB. Algoritma K-Means mengelompokkan warna piksel ke dalam kluster dengan memperbarui pusat kluster secara iteratif hingga konvergensi tercapai; penelitian ini menguji beberapa variasi jumlah kluster, yaitu 256, 128, 64, 32, dan 8 kluster.

Teknik Analisis Data

Teknik analisis data yang diterapkan bersifat kuantitatif, membandingkan hasil pemrosesan citra sebelum dan sesudah penerapan DCT dan K-Means Clustering. Data yang dianalisis mencakup ukuran file, nilai MSE, dan nilai PSNR, yang diperiksa pada setiap tahap pemrosesan dan dibandingkan berdasarkan variasi jumlah kluster yang digunakan. Analisis deskriptif juga dilakukan dengan mengamati perubahan visual pada citra untuk menilai kualitas secara subjektif. Teknik analisis data yang diterapkan bersifat kuantitatif, membandingkan hasil pemrosesan citra sebelum dan sesudah penerapan DCT dan K-Means Clustering.

Mean Squared Error (MSE) digunakan untuk mengukur rata-rata kuadrat selisih antara citra asli dan citra hasil pemrosesan. Nilai MSE yang semakin kecil menunjukkan bahwa citra hasil pemrosesan semakin mendekati citra asli. Persamaan MSE ditunjukkan sebagai berikut:

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [I(i,j) - K(i,j)]^2$$

dengan:

MMM = jumlah baris citra,

NNN = jumlah kolom citra,

$I(i,j)I(i,j)I(i,j)$ = nilai piksel citra asli pada koordinat $(i,j)(i,j)(i,j)$,

$K(i,j)K(i,j)K(i,j)$ = nilai piksel citra hasil kompresi/konversi pada koordinat $(i,j)(i,j)(i,j)$.

Peak Signal-to-Noise Ratio (PSNR) digunakan untuk mengukur kualitas citra hasil pemrosesan berdasarkan nilai MSE. Semakin tinggi nilai PSNR, maka kualitas citra hasil pemrosesan semakin baik dan semakin mendekati citra asli. Nilai PSNR dihitung menggunakan persamaan berikut:

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right)$$

Nilai MAX menunjukkan nilai intensitas piksel maksimum yang dapat direpresentasikan oleh citra digital. Dalam penelitian ini digunakan citra 8-bit sehingga nilai maksimum intensitas piksel adalah 255, sedangkan MSE (Mean Squared Error) merupakan nilai rata-rata kuadrat kesalahan yang diperoleh dari hasil perbandingan antara citra asli dan citra hasil pemrosesan. Berdasarkan persamaan tersebut, nilai PSNR berbanding terbalik dengan nilai MSE, sehingga semakin kecil nilai MSE yang dihasilkan maka semakin besar nilai PSNR. Kondisi ini menunjukkan bahwa kualitas citra hasil pemrosesan semakin baik karena tingkat distorsi terhadap citra asli semakin rendah.

HASIL DAN PEMBAHASAN

Algoritma K-Means Clustering kemudian diterapkan pada citra hasil DCT untuk mengurangi jumlah warnanya menjadi 256 warna. Hal ini dicapai dengan mengelompokkan seluruh piksel ke dalam 256 kluster menggunakan algoritma K-Means dengan inisialisasi acak, mengganti setiap piksel dengan warna pusat kluster terdekatnya, dan menghasilkan citra yang telah dikuantisasi K-Means, yang disimpan dalam format GIF, menampilkan citra dengan palet warna yang kompak dan optimal untuk format GIF.

Evaluasi Kualitas Citra

Kualitas citra dievaluasi menggunakan MSE dan PSNR, membandingkan setiap citra hasil pemrosesan dengan citra grayscale referensi asli, di mana nilai MSE yang lebih rendah dan PSNR yang lebih tinggi menunjukkan kualitas yang lebih baik. Citra JPEG hasil proses DCT menghasilkan MSE sebesar 42,3175 dan PSNR sebesar 31,8692 dB, sedangkan GIF K-Means 256 kluster menghasilkan MSE sebesar 47,6381 dan PSNR sebesar 31,3504 dB. Hasil DCT sedikit lebih baik, menunjukkan bahwa kuantisasi warna memperkenalkan penurunan kualitas tambahan yang kecil akibat batasan 256 warna; namun selisih antara kedua metode tersebut kecil, mengindikasikan bahwa keluaran K-Means tetap mendekati kualitas citra yang dikompresi DCT.

Pemeriksaan visual yang diperbesar mendukung temuan ini. Citra asli mempertahankan detail penuh dan transisi warna yang halus sebagai referensi. Setelah DCT, detail halus dan tekstur sedikit berkurang, dengan sedikit blurring dan artefak blok awal, meskipun objek utama tetap dapat dikenali dengan jelas. Setelah kuantisasi K-Means menjadi 256 kluster, distribusi warna sedikit bergeser, dengan wilayah warna yang lebih seragam muncul di area gradien dan pencahayaan, sementara objek tetap dapat diidentifikasi dengan jelas. Hal ini mengkonfirmasi bahwa DCT terutama memengaruhi detail bentuk dan tekstur, sedangkan K-Means Clustering terutama memengaruhi distribusi warna, dan keduanya mempertahankan struktur inti citra.

Perbandingan Ukuran File

Ukuran file dibandingkan pada setiap tahap pemrosesan untuk menilai efisiensi kompresi. Proses DCT, yang disimpan sebagai JPEG, mengurangi ukuran file dari 1347,83 KB menjadi 1198,46 KB, yaitu pengurangan sebesar 11,08% yang dapat dikaitkan dengan penghilangan komponen frekuensi tinggi yang kurang signifikan bagi persepsi manusia. Mengkonversi hasil DCT menjadi GIF 256 klaster melalui K-Means justru meningkatkan ukuran file secara tajam menjadi 5124,70 KB, peningkatan sebesar 280,22% dari aslinya. Ini menunjukkan bahwa GIF tidak selalu menghasilkan file yang lebih kecil dari JPG, khususnya untuk citra dengan kompleksitas warna tinggi.

Eksperimen Variasi Jumlah Klaster

Eksperimen lanjutan mengkaji bagaimana jumlah klaster, yaitu 256, 128, 64, 32, dan 8 klaster, memengaruhi kualitas dan ukuran file citra hasil DCT setelah kuantisasi K-Means. Pada 256 klaster, kualitas tetap sangat mendekati aslinya, dengan transisi halus pada detail wajah, pakaian, dan latar belakang. Pada 128 klaster, perubahan gradasi dan tekstur kecil mulai terlihat, meskipun kualitas keseluruhan masih baik. Pada 64 klaster, transisi warna menjadi lebih kasar dan posterisasi mulai muncul. Pada 32 klaster, kualitas menurun lebih lanjut, dengan nada yang lebih datar, blok warna yang lebih terlihat, dan hilangnya detail halus. Pada 8 klaster, degradasi paling parah, dengan gradasi hampir hilang dan posterisasi kuat, meskipun bentuk objek utama masih dapat dikenali. Rangkuman lengkap ukuran file, MSE, dan PSNR untuk semua konfigurasi yang diuji disajikan pada Tabel 1.

Tabel 1. Hasil Eksperimen K-Means Clustering

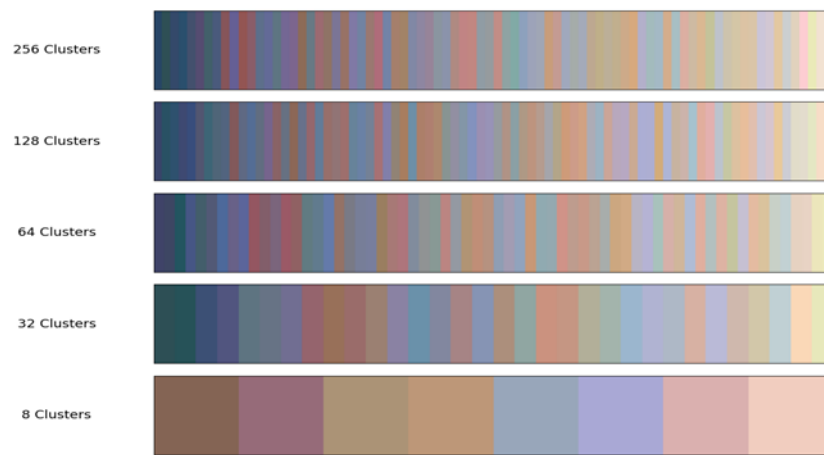
Konfigurasi	Ukuran File (KB)	MSE	PSNR (dB)
JPG Asli	1347,83	-	-
Setelah DCT (JPEG)	1198,46	42,3175	31,8692
K-Means GIF (256 Klaster)	5124,70	47,6381	31,3504
K-Means GIF (128 Klaster)	4012,38	51,9247	30,9813
K-Means GIF (64 Klaster)	2874,61	59,4813	30,3891
K-Means GIF (32 Klaster)	1943,27	74,8562	29,3917
K-Means GIF (8 Klaster)	684,35	146,8934	25,9143

Sebagaimana ditunjukkan pada Tabel 1, jumlah klaster sangat memengaruhi ukuran file dan kualitas: semakin sedikit klaster, semakin sedikit informasi warna yang perlu disimpan, sehingga ukuran file lebih kecil, dengan GIF 8 klaster menghasilkan ukuran terkecil, yaitu 684,35 KB, bahkan lebih kecil dari JPG asli maupun JPEG hasil DCT.

Namun hal ini berdampak pada kualitas: MSE naik dan PSNR turun seiring berkurangnya jumlah klaster, mencapai MSE sebesar 146,8934 dan PSNR sebesar 25,9143 dB pada 8 klaster. Pada 256 klaster, kualitas tetap lebih dekat ke aslinya (MSE 47,6381, PSNR 31,3504 dB), meskipun ukuran file jauh lebih besar. Posterisasi, yang terlihat sebagai blok warna kasar yang menggantikan gradien halus, tampak pada 32 dan 8 klaster.



Gambar 2. Perbandingan Visual Hasil K-Means pada Berbagai Jumlah Klaster



Gambar 3. Perbandingan Palet Warna Kluster K-Means

Hasil eksperimen secara keseluruhan menunjukkan bahwa kombinasi algoritma DCT dan K-Means Clustering dapat secara efektif melakukan konversi JPG ke GIF, meskipun terdapat pertukaran antara kualitas citra dan efisiensi ukuran file. Tahap DCT mengurangi komponen frekuensi tinggi untuk membuat data citra lebih efisien tanpa secara substansial mengorbankan struktur utamanya, sementara tahap K-Means Clustering mengurangi jumlah warna sesuai persyaratan format GIF dengan mengelompokkan piksel ke dalam palet berbasis kluster yang kompak. Semakin sedikit kluster yang digunakan, ukuran file berkurang, namun kualitas visual juga menurun, dengan efek posterisasi yang terlihat pada jumlah kluster rendah seperti 32 dan 8 kluster.

KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, algoritma DCT dan K-Means Clustering berhasil diimplementasikan dalam bahasa Python untuk konversi citra JPG ke format GIF, di mana DCT berhasil menurunkan ukuran file dari 1347,83 KB menjadi 1198,46 KB dengan MSE 42,3175 dan PSNR 31,8692 dB, sementara K-Means Clustering melakukan kuantisasi warna dengan jumlah kluster yang sangat memengaruhi hasil: pada 256 kluster kualitas tetap mendekati aslinya (PSNR 31,3504 dB) namun ukuran file meningkat signifikan, sedangkan pada 8 kluster ukuran file turun ke nilai terkecil, 684,35 KB, dengan konsekuensi penurunan kualitas berupa posterisasi dan hilangnya detail warna; secara keseluruhan kombinasi kedua algoritma ini terbukti efektif untuk konversi JPG ke GIF dan menunjukkan bahwa GIF tidak selalu menghasilkan ukuran file lebih kecil dari JPG, sehingga pemilihan jumlah kluster yang tepat menjadi faktor kunci dalam menyeimbangkan kualitas dan efisiensi ukuran file, dan penelitian selanjutnya dapat mengeksplorasi metode kuantisasi warna alternatif seperti Median Cut, Octree, atau NeuQuant serta menganalisis waktu komputasi dan kinerja sistem.

DAFTAR PUSTAKA

- [1] N. Amanati, W. Setyawan, dan D. Sari, "Implementasi Discrete Cosine Transform (DCT) pada kompresi citra digital untuk peningkatan efisiensi penyimpanan," *Jurnal Teknologi Informasi dan Terapan*, vol. 9, no. 2, hlm. 145–154, 2021.
- [2] Y. Kusuma dan D. Lestari, "Evaluasi kualitas kompresi citra menggunakan metrik PSNR dan MSE pada berbagai algoritma transformasi," *Jurnal Multimedia dan Komputer*, vol. 5, no. 2, hlm. 77–86, 2021.
- [3] R. Dewi dan F. Pratama, "Optimasi pengolahan citra menggunakan metode Discrete Cosine Transform pada media digital," *Jurnal Informatika dan Komputer*, vol. 10, no. 3, hlm. 189–198, 2021.
- [4] I. Putra dan A. Santoso, "Implementasi K-Means Clustering pada proses kuantisasi warna citra digital berformat GIF," *Journal of Digital Imaging Technology*, vol. 4, no. 1, hlm. 11–22, 2022.
- [5] A. Wibowo dan H. Santoso, "Analisis perbandingan format citra digital JPG dan GIF pada aplikasi multimedia," *Jurnal Teknologi dan Sistem Informasi*, vol. 3, no. 1, hlm. 33–42, 2022.
- [6] R. Gifani, "An improved K-Means color quantization for efficient GIF encoding," *International Journal of Image Processing*, vol. 17, no. 1, hlm. 55–64, 2023.
- [7] C. Halim dan R. Wahono, "Studi komparatif teknik kuantisasi warna dalam pembuatan GIF," *Jurnal Sistem dan Teknologi Informasi*, vol. 8, no. 4, hlm. 301–312, 2020.

- [8] B. Rahmadi dan S. Nugroho, "Analisis format JPG dan GIF pada sistem penyimpanan multimedia modern," *Jurnal Teknologi Media Digital*, vol. 6, no. 1, hlm. 41–52, 2023.
- [9] M. Sundari dan R. Riyadi, "Pengaruh jumlah warna terhadap kualitas hasil konversi GIF pada citra digital," *Jurnal Informatika Nusantara*, vol. 3, no. 2, hlm. 98–108, 2020.
- [10] Rangkuti, A.R.S., & Sari, I.P. (2026). Penggunaan Fitur Warna Cielab Dalam Segmentasi Kematangan Buah Kakao Dengan Algoritma K-Means Clustering. *Hello World Jurnal Ilmu Komputer* 5 (1), 36-46
- [11] Sari, I.P., Basri, M., & Syafrayani, P.R. (2025). Implementasi Sistem Aplikasi Pengolahan Teks pada Gambar Menggunakan Modifikasi Metode LSB dan ROT13. *Hello World Jurnal Ilmu Komputer* 4 (2), 90-97
- [12] Sari, I. P., Al-Khowarizmi, & Batubara, I. H. (2021). Cluster analysis using K-Means algorithm and fuzzy C-Means clustering for grouping students' abilities in online learning process. *Journal of Computer Science, Information Technology and Telecommunication Engineering*, 2(1), 139–144.
- [13] Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th ed.). Pearson.
- [14] Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2004). *Digital Image Processing Using MATLAB*. Pearson Prentice Hall.
- [15] ain, A. K. (1989). *Fundamentals of Digital Image Processing*. Prentice Hall.
- [16] Pratt, W. K. (2007). *Digital Image Processing: PIKS Scientific Inside* (4th ed.). John Wiley & Sons.
- [17] Pennebaker, W. B., & Mitchell, J. L. (1993). *JPEG: Still Image Data Compression Standard*. Van Nostrand Reinhold.
- [18] MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.
- [19] Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), 129–137.
- [20] Salomon, D. (2007). *Data Compression: The Complete Reference* (4th ed.). Springer.
- [21] Bovik, A. C. (Ed.). (2005). *Handbook of Image and Video Processing* (2nd ed.). Academic Press.