

Analisis Komparatif Kinerja DCT, DWT, dan PCA pada Teknik Kompresi Citra Lossy

Fris Alvin¹, Sofyan Saif Pahlevi²

^{1,2}Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Kota Semarang

Artikel Info

Kata kunci:

Discrete Cosine Transform
Discrete Wavelet Transform
Kompresi Citra Lossy
Principal Component Analysis
Study Komparatif

ABSTRAK

Dalam era digital saat ini, kebutuhan pengelolaan dan penyimpanan data citra secara efisien menjadi semakin penting, terutama di sektor seperti fotografi, kedokteran, dan media digital. Kompresi lossy sendiri bisa menawarkan penghematan ukuran file yang signifikan, cocok untuk distribusi multimedia dengan batasan bandwidth. Dalam penelitian ini membahas perbandingan antara tiga algoritma populer untuk kompresi citra lossy, yaitu Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), dan Principal Component Analysis (PCA) untuk mengetahui dalam hal efisiensi kompresi, kualitas citra, dan waktu pemrosesan. Antarmuka pengguna grafis (GUI) berbasis MATLAB digunakan untuk memudahkan analisis performa setiap algoritma. GUI yang dikembangkan memungkinkan pengguna untuk mengatur parameter kompresi, mengevaluasi performa secara real-time, dan memilih algoritma yang paling sesuai dengan kebutuhan, baik dari segi kualitas citra maupun efisiensi ukuran citra. Hasil penelitian menunjukkan bahwa algoritma DCT memberikan efisiensi kompresi tertinggi dan terus stabil pada citra dengan format .jpg yang mencapai 88,935% dan 55,859%,. Algoritma DWT unggul dalam segi kompresi citra dengan format .png karena dari perbandingan antara DCT dan PCA algoritma DWT memberikan hasil kompresi yang sangat baik untuk citra lossy yaitu mencapai 77,2096% dan 63,1969%. Sementara algoritma PCA menunjukan tingkat efisiensi kompresi citra yang bervariasi tergantung pada jenis format citra yang digunakan, dimana saat kompresi format .png kb citra yang dihasilkan malah lebih tinggi dari citra aslinya dan untuk format .jpg ada dimana algoritma PCA unggul dari algoritma DCT dan DWT. Kesimpulannya, algoritma DCT cocok untuk aplikasi multimedia karena efisiensi kompresinya, DWT unggul untuk citra dengan detail tinggi, dan PCA bermanfaat untuk pengurangan dimensi data tertentu.

Penulis Korespondensi :

Fris Alvin,
Program Studi Teknik Informatika
Fakultas Ilmu Komputer
Universitas Dian Nuswantoro, Kota Semarang, 50131
Email: 111202213986@mhs.dinus.ac.id

1. PENDAHULUAN

Dalam perkembangan era digital saat ini, kompresi citra menjadi salah satu teknik yang sangat dibutuhkan, terutama dalam mengoptimalkan penyimpanan dan mengelola kualitas dari citra digital [1], [2]. Kompresi citra bertujuan untuk mengurangi ukuran file citra digital, sehingga memungkinkan efisiensi dari penggunaan ruang penyimpanan dan proses transfer data menjadi lebih cepat [3]. Keberadaan teknologi kompresi sangat penting di berbagai bidang, seperti media digital, kedokteran, dan ilmu forensik, di mana data visual dalam jumlah besar harus disimpan dan diakses dengan cepat tanpa mengurangi kualitas informasi yang diperlukan [4], [5]. Metode kompresi lossy banyak digunakan dalam menawarkan solusi yang efisien untuk mengurangi ukuran file dengan mengorbankan sebagian detail dari citra yang terkompresi [6]. Pendekatan ini sangat relevan untuk aplikasi yang tidak memerlukan ketelitian visual tingkat tinggi, seperti

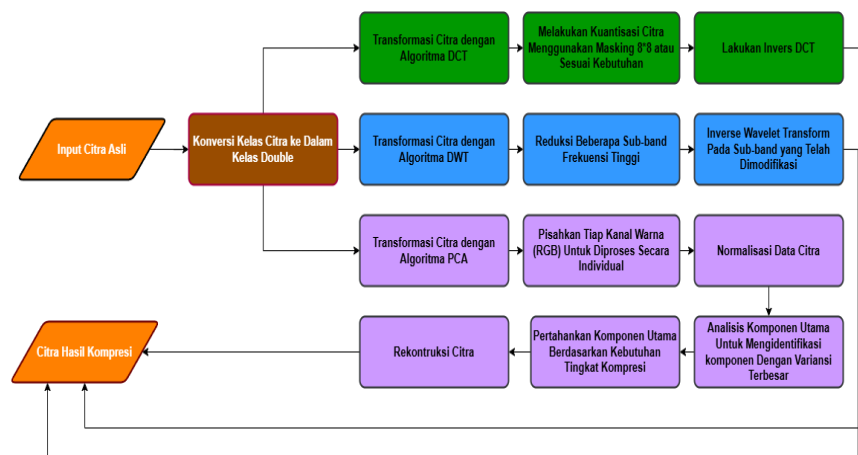
distribusi multimedia, penyiaran video, atau pengarsipan gambar dalam skala besar [7]. Dalam penelitian ini beberapa algoritma akan digunakan untuk penelitian dan juga mengevaluasi algoritma terbaik dalam hal kompresi citra lossy. Algoritma yang akan digunakan dalam penelitian kompresi lossy adalah Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), dan Principal Component Analysis (PCA) [8], [9].

Algoritma DCT menjadi salah satu pilihan dalam penelitian kali ini, efisiensi algoritma DCT terletak pada kemampuannya dalam memisahkan frekuensi rendah dan tinggi pada citra. Bagian yang kurang signifikan secara visual dapat dihapus untuk mengurangi ukuran file tanpa mengorbankan kualitas terlalu banyak. Algoritma DWT menggunakan pendekatan berbasis skala yang memanfaatkan teknik filterasi digital [10]. Metode ini menghasilkan kualitas visual yang lebih baik, meskipun membutuhkan lebih banyak sumber daya komputasi. Sementara itu, algoritma PCA berfokus pada reduksi dimensi data dimana dengan mentransformasikan citra ke ruang baru yang mempertahankan informasi paling penting, namun sering kali disertai penurunan kualitas visual yang lebih jelas dibandingkan algoritma DCT atau DWT [11]. Hingga saat ini belum ada penelitian secara khusus membandingkan untuk ketiga algoritma DCT, DWT, dan PCA sekaligus dalam hal kompresi citra lossy.

Didukung dengan penggunaan perangkat lunak berupa MATLAB, dengan antarmuka berbasis GUI (Graphical User Interface), menyediakan platform yang memudahkan dalam proses kompresi citra [12]. Dengan GUI pengguna dapat merubah detail dan warna pada citra serta menghasilkan citra dengan ukuran file yang lebih kecil sesuai dengan kebutuhan dan hasil kompresi dapat langsung dilihat secara visual dan real time. Pada penelitian ini, dilakukan perbandingan dari penggunaan algoritma DCT, DWT, dan PCA untuk kompresi citra lossy. Dengan melihat sejauh mana efisiensi pengurangan ukuran file citra, kualitas citra setelah terkompresi, dan waktu pemrosesan untuk setiap algoritma. Melalui penelitian ini, diharapkan dapat diperoleh wawasan mendalam mengenai kelebihan dan kekurangan masing-masing algoritma, sehingga pengguna dapat memilih algoritma yang paling sesuai dengan kebutuhan spesifik masing-masing. Pendekatan ini juga diharapkan dapat memberikan kontribusi terhadap pengembangan lebih lanjut teknologi kompresi citra digital yang efisien, relevan, dan dapat diimplementasikan secara praktis di berbagai bidang aplikasi.

2. METODE

Pada penelitian ini, dilakukan penerapan dan perbandingan tiga algoritma untuk kompresi citra lossy, yaitu Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), dan Principal Component Analysis (PCA). Penelitian ini bertujuan untuk menganalisa kinerja dari ketiga algoritma yang digunakan, dalam segi hasil kompresi yang paling efisien untuk pengurangan ukuran citra untuk berbagai format seperti BMP, JPG, dan PNG serta tingkat efisiensi waktu dalam proses kompresi berlangsung. Dilakukan dengan menggunakan perangkat lunak MATLAB untuk membangun antarmuka pengguna grafis (GUI) yang memfasilitasi proses kompresi dan analisis. GUI ini dirancang untuk memberikan kemudahan bagi pengguna dalam mengamati hasil kompresi secara langsung, baik dari segi tampilan citra maupun ukuran citra yang dihasilkan. Pada penelitian ini akan dijelaskan cara kerja dari ketiga algoritma DCT, DWT, dan PCA dalam melakukan kompresi terhadap citra lossy agar dalam proses kompresi bisa mendapatkan hasil yang terbaik dalam hal efisiensi penyimpanan file citra, akan dijelaskan dalam bagan alur atau flowchart pada Gambar 1.



Gambar 1. Bagan alur kompresi citra dengan algoritma DCT, DWT, dan PCA

2.1. Discrete Cosine Transform (DCT)

Pada penelitian ini, algoritma Discrete Cosine Transform (DCT) digunakan untuk memisahkan frekuensi rendah dan tinggi pada citra digital yang akan dikompresi, dengan berfokus pada optimalisasi ukuran file [13], [14]. Proses dimulai dengan membagi citra asli menjadi blok-blok kecil berukuran 8x8 piksel. Setiap blok kemudian diubah dari domain spasial ke domain frekuensi menggunakan transformasi DCT. Komponen frekuensi rendah, yang membawa informasi utama citra akan dipertahankan, sementara komponen frekuensi tinggi yang kurang signifikan secara visual akan diabaikan. Algoritma ini memungkinkan pengurangan ukuran file yang signifikan, terutama pada aplikasi seperti pengiriman gambar atau streaming multimedia. Namun, karena beberapa informasi visual pada citra dihilangkan, kualitas citra hasil kompresi sangat bergantung pada tingkat agresivitas matriks yang digunakan dalam proses kompresi yang diterapkan.

2.2. Discrete Wavelete Transform (DWT)

Algoritma kedua yang digunakan untuk melakukan perbandingan pada kompresi citra lossy adalah algoritma Discrete Wavelet Transform (DWT) dimana algoritma ini menggunakan pendekatan berbasis skala untuk memecah citra menjadi berbagai lapisan frekuensi sesuai dengan tingkatan transformasi yang diinginkan [15]. Dengan teknik ini, algoritma DWT dapat menangkap detail halus dan struktur citra secara lebih efektif dibandingkan algoritma DCT dalam kompresi citra lossy. Prosesnya melibatkan dekomposisi citra ke dalam beberapa tingkat, di mana setiap tingkat menyimpan informasi spesifik tentang frekuensi tertentu [16]. Keunggulan utama dari algoritma DWT adalah kemampuannya untuk mempertahankan detail visual yang dianggap penting pada tingkat kompresi yang lebih tinggi, meskipun membutuhkan waktu pemrosesan yang lebih lama. Hal ini menjadikan algoritma ini salah satu pilihan yang baik untuk aplikasi yang memerlukan kualitas visual tinggi, seperti pencitraan medis atau fotografi.

2.3. Principal Component Analysis (PCA)

Algoritma terakhir yang digunakan untuk melakukan perbandingan pada kompresi citra lossy yaitu algoritma Principal Component Analysis (PCA) dimana algoritma PCA ini mengambil pendekatan yang berbeda dengan berfokus pada pengurangan dimensi data yang dimiliki [17]. Algoritma PCA bekerja dengan mentransformasi citra ke ruang baru yang hanya mempertahankan komponen utama yang memiliki variansi terbesar. Komponen minor yang dianggap kurang signifikan akan diabaikan, sehingga ukuran file dapat dikurangi secara signifikan. Namun, kekurangannya adalah kualitas citra yang dihasilkan sering kali lebih terpengaruh dibandingkan dengan algoritma seperti DCT atau DWT, karena beberapa detail penting bisa hilang dalam proses ini. Dengan itu algoritma PCA ini biasanya sangat cocok untuk kebutuhan yang mengutamakan efisiensi penyimpanan, misalnya dalam pengelolaan kumpulan data citra yang besar. Meski begitu, penurunan kualitas dari kompresi citra yang dihasilkan akan lebih buruk sebagai bagian dari kompromi yang dihasilkan oleh algoritma ini.

2.4. Evaluasi Performa

Dalam penelitian ini, kualitas citra hasil kompresi dievaluasi menggunakan beberapa parameter, yaitu Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), dan Structural Similarity (SSIM). Parameter PSNR digunakan untuk mengukur perbedaan intensitas antara citra asli dan citra terkompresi. Nilai PSNR yang lebih tinggi mengindikasikan kualitas citra kompresi yang lebih baik karena perbedaan intensitas antara citra asli dan kompresi lebih kecil. Berikut adalah rumus untuk menghitung nilai PSNR suatu citra.

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \quad (1)$$

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

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3)$$

$$\text{Persentasi Kompresi} = \left(1 - \frac{\text{Size}_{\text{kompresi}}}{\text{Size}_{\text{asli}}} \right) \times 100\% \quad (4)$$

PSNR merupakan ukuran yang sering digunakan untuk mengevaluasi kualitas kompresi citra [18]. PSNR dihitung berdasarkan nilai MSE dan nilai maksimum piksel dalam citra, seperti 255 untuk citra 8-bit. Semakin tinggi nilai PSNR, semakin baik kualitas citra hasil kompresi karena menunjukkan bahwa tingkat noise atau distorsi pada citra kompresi relatif rendah dibandingkan dengan citra asli.

MSE digunakan untuk menghitung rata-rata kesalahan kuadrat antara citra asli dan citra hasil kompresi [19]. Nilai MSE memberikan indikasi seberapa jauh perbedaan antara kedua citra tersebut. Rumus MSE melibatkan ukuran panjang (M) dan lebar (N) citra, serta nilai piksel pada posisi tertentu di citra asli $I(i, j)$ dan citra kompresi $K(i, j)$. Nilai MSE yang lebih rendah menunjukkan bahwa hasil kompresi memiliki kualitas yang lebih baik, dengan perbedaan minimal antara citra asli dan hasil kompresinya.

SSIM digunakan untuk mengukur kesamaan struktur antara citra asli dan citra hasil kompresi dengan mempertimbangkan tiga faktor utama: luminans, kontras, dan struktur [20]. Rumus SSIM melibatkan rata-rata piksel (μ_x dan μ_y), variansi piksel (σ_x^2 dan σ_y^2), serta kovariansi (σ_{xy}) antara citra asli dan kompresi. Nilai SSIM berkisar antara 0 hingga 1, dengan nilai mendekati 1 menunjukkan kesamaan struktur yang sangat tinggi antara kedua citra.

Selain pengukuran kualitas visual, ukuran file kompresi juga dihitung untuk mengevaluasi efisiensi kompresi. Persentase ukuran file kompresi dihitung menggunakan rasio antara ukuran file asli ($\text{Size}_{\text{asli}}$) dan ukuran file hasil kompresi ($\text{Size}_{\text{kompresi}}$). Semakin kecil ukuran file setelah kompresi, semakin efisien proses kompresi tersebut. Namun, efisiensi ukuran harus tetap seimbang dengan kualitas citra agar hasil kompresi tidak mengorbankan kualitas visual secara signifikan.

3. PEMBAHASAN HASIL

Pada penelitian ini, dilakukan perbandingan dari ketiga algoritma yaitu DCT, DWT, dan PCA pada kompresi citra lossy melalui GUI yang dikembangkan di MATLAB. Dalam penelitian ini, saat merancang GUI banyak aspek yang diperlukan seperti tampilan untuk ukuran citra asli dan citra hasil, tampilan gambar citra asli dan citra hasil, persen kompresi, tampilan PSNR, tampilan MSE, tampilan SSIM, dan fungsional lainnya dalam menunjang agar GUI yang dibuat dapat berjalan dengan baik.

GUI ini dirancang dengan beberapa elemen interaktif yang memungkinkan untuk lebih mudah menggunakan dan memahami hasil dari perbandingan ketiga algoritma yang akan dilakukan untuk mengetahui seberapa efisien algoritma DCT, DWT, dan PCA dalam hal efisiensi ukuran citra. Setiap komponen GUI ini dikaitkan langsung dengan fungsi-fungsi spesifik dalam kode untuk memastikan proses kompresi berjalan sesuai dengan parameter yang telah ditentukan. Misalnya, tombol "Browse Image" terhubung dengan fungsi untuk memuat gambar asli, yang kemudian ditampilkan pada panel "Citra Asli". Tombol "Compress Image" mengaktifkan fungsi kompresi yang sesuai dengan algoritma kompresi citra lossy yang digunakan dan menghasilkan gambar terkompresi di panel "Citra Hasil". Sementara itu, komponen teks di bagian bawah GUI secara otomatis menampilkan nilai-nilai metrik kualitas seperti PSNR, MSE, SSIM, dan persentase kompresi berdasarkan perbandingan antara citra asli dan citra hasil kompresi. Penjelasan bagaimana algoritma DCT, DWT, dan PCA bekerja untuk perbandingan kompresi citra lossy dengan menggunakan MATLAB dan dibuat agar bisa berfungsi secara optimal akan di jelaskan kedalam pseudocode berikut ini.

Start

```

Ambil citra asli dari input yang diberikan (variabel "handles.img").
Dapatkan informasi tentang file citra (variabel "handles.fileinfo").
Buat matriks kosong (variabel "img_comp") untuk menyimpan citra hasil kompresi dengan ukuran yang sama seperti gambar asli.
Untuk setiap kanal warna dalam citra (R, G, B), lakukan langkah-langkah berikut:
    • Ubah kanal warna ke tipe angka desimal (double) agar bisa dihitung dengan lebih presisi.
    • Buat matriks transformasi DCT ukuran 8x8, yang nantinya dipakai untuk mengubah data citra ke domain frekuensi.
    • Terapkan transformasi DCT ke citra dalam blok-blok kecil berukuran 8x8.
    • Buat pola mask (matriks 8x8) yang menentukan data mana yang akan dipertahankan saat kompresi.
    • Gunakan mask tersebut untuk membuang data yang tidak penting di tiap blok.
    • Setelah data citra direduksi, ubah kembali citra ke bentuk aslinya dengan menerapkan DCT invers.
    • Simpan hasil pemrosesan ke dalam (variabel "img_comp") untuk citra hasil kompresi.
Gabungkan semua kanal warna untuk menghasilkan citra yang sudah terkompresi.

```

End

Langkah-langkah dari pseudocode diatas menunjukan pemakaian algoritma DCT dalam kompresi citra lossy dimana fungsi ini memproses citra asli dengan cara membagi citra menjadi blok-blok kecil berukuran 8x8, kemudian menerapkan transformasi DCT pada setiap blok. Transformasi ini dilakukan menggunakan matriks transformasi DCT yang dihasilkan oleh fungsi `dctmtx (8)`. Setelah transformasi DCT dilakukan, setiap blok hasil transformasi dikalikan dengan sebuah matriks masking berukuran 8x8. Matriks ini berfungsi untuk menghilangkan komponen frekuensi tinggi yang dianggap kurang signifikan, sehingga ukuran file citra dapat dikurangi secara signifikan. Matriks masking pada kode yang dibuat hanya mempertahankan nilai-nilai pada beberapa komponen frekuensi rendah, sementara komponen lainnya diabaikan. Proses masking ini dilakukan dengan fungsi `blkproc` yang menerapkan operasi pada blok-blok citra.

Setelah masking diterapkan, blok-blok yang telah diatur ulang tersebut kemudian diubah kembali ke domain spasial dengan menggunakan transformasi DCT invers. Proses ini memastikan bahwa citra hasil kompresi tetap berada dalam format yang dapat dilihat secara visual. Akhirnya, citra hasil kompresi disimpan dalam variabel `Img_comp`, dengan struktur yang sama seperti citra asli. kode ini menggambarkan bagaimana algoritma DCT dapat bekerja untuk mengompresi citra dengan menghilangkan informasi yang tidak terlalu penting, sekaligus mempertahankan kualitas visual citra sesuai dengan kebutuhan. Implementasi dari algoritma DCT ini dapat dikembangkan lebih lanjut untuk menyesuaikan tingkat kompresi dengan mengubah matriks masking sesuai kebutuhan.

Start

Ambil citra asli dari input yang diberikan (variabel "handles.Image").
 Dapatkan informasi tentang file citra (variabel "handles.fileinfo").
 Pilih tipe wavelet yang akan digunakan, misalnya 'haar' atau 'db1'.
 Buat matriks kosong (variabel "Img_comp") untuk menyimpan citra hasil kompresi dengan ukuran yang sama seperti gambar asli.

Untuk setiap kanal warna dalam citra (R, G, B), lakukan langkah-langkah berikut:

- Ubah kanal warna ke tipe angka desimal (double) agar bisa dihitung dengan lebih presisi.
- Terapkan Wavelet Decomposition untuk membagi citra menjadi beberapa sub-band frekuensi:
 - Sub-band cA (Approximation) menyimpan informasi utama citra.
 - Sub-band cH (Horizontal), cV (Vertical), dan cD (Diagonal) menyimpan detail-detail citra.
- Buang informasi dari sub-band cH, cV, dan cD dengan mengatur nilainya menjadi nol, untuk mengurangi ukuran data citra.
- Setelah sub-band cH, cV, dan cD dihilangkan, ubah kembali citra ke bentuk aslinya dengan menerapkan Inverse Wavelet Transform.
- Simpan hasil pemrosesan ke dalam (variabel "Img_comp") untuk citra hasil kompresi.

Gabungkan semua kanal warna untuk menghasilkan citra yang sudah terkompresi.

End

Pseudocode diatas menunjukan pemakaian algoritma DWT dalam kompresi citra lossy, dimulai dengan membaca citra asli dan mengonversinya ke dalam tipe data double agar lebih mudah diolah. Setiap citra berwarna memiliki tiga kanal warna seperti merah, hijau, dan biru atau biasa disebut rgb yang akan diproses secara terpisah menggunakan transformasi wavelet dua dimensi, dengan tipe wavelet yang digunakan adalah haar. Transformasi ini menghasilkan empat bagian yaitu koefisien aproksimasi (cA), horizontal (cH), vertikal (cV), dan diagonal (cD), yang masing-masing mewakili tingkat detail yang berbeda dalam citra. Untuk mengurangi ukuran file, detail pada sub-band horizontal, vertikal, dan diagonal dihapus dengan mengatur nilainya menjadi nol, sementara koefisien aproksimasi dipertahankan karena memuat informasi utama citra.

Setelah proses ini, citra dikembalikan ke bentuk aslinya menggunakan transformasi wavelet invers. Ukuran hasil rekonstruksi disesuaikan agar sesuai dengan dimensi citra asli. Hasil kompresi untuk setiap kanal warna disimpan dalam variabel `Img_comp`, yang merepresentasikan citra hasil kompresi. Pendekatan ini efektif untuk mengurangi ukuran file dengan tetap mempertahankan fitur utama citra, sehingga sangat cocok untuk aplikasi yang memerlukan efisiensi ruang penyimpanan tanpa terlalu banyak kehilangan informasi visual yang penting.

Start

Ambil citra asli dari input yang diberikan (variabel “handles.img”).

Dapatkan dimensi gambar: jumlah baris (rows), kolom (cols), dan kanal warna (channels).

Buat matriks kosong (variabel “Img_comp”) untuk menyimpan citra hasil kompresi dengan ukuran yang sama seperti gambar asli.

Tentukan jumlah komponen utama (variabel “numComponents”) yang akan dipertahankan.

Untuk setiap kanal warna dalam citra (R, G, B), lakukan langkah-langkah berikut:

- Ambil kanal warna ke-n dari citra asli (I).
- Ubah kanal warna ke bentuk matriks 2D (reshape).
- Hitung rata-rata dari setiap piksel dalam kanal (I_mean).
- Kurangi rata-rata dari data untuk melakukan centering data (I_centered).
- Lakukan PCA pada data yang telah di centered untuk mendapatkan: koefisien (coeff) dan skor proyeksi (score).
- Pilih sejumlah komponen utama (numComponents) untuk mempertahankan informasi penting.
- Rekonstruksi kanal warna dari komponen utama yang dipertahankan.
- Simpan kanal warna yang direkonstruksi ke dalam (variabel “Img_comp”) untuk citra hasil kompresi.

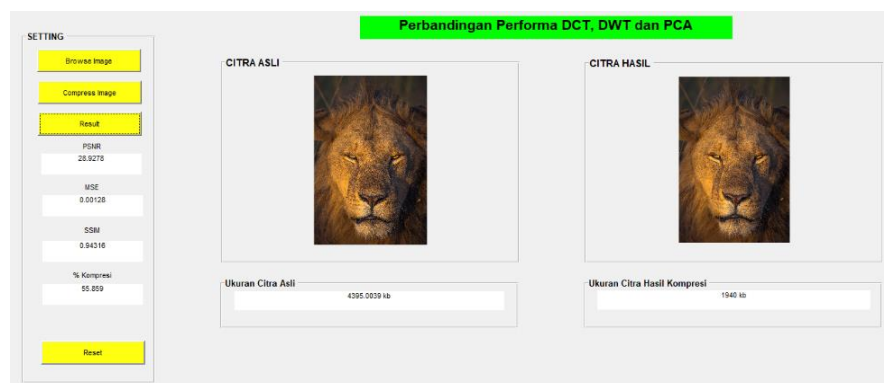
Gabungkan semua kanal warna untuk menghasilkan citra yang sudah terkompresi.

End

Pseudocode diatas menunjukkan pemakaian algoritma PCA dalam kompresi citra lossy. Dengan mempertahankan sejumlah komponen utama tertentu. Pertama, citra asli diambil dari variabel handles.img, dan dimensinya yang berupa baris, kolom, dan kanal warna dihitung. Sebuah matriks kosong Img_comp digunakan untuk menyimpan hasil citra yang telah dikompresi. Setiap kanal warna merah, hijau, dan biru diproses secara terpisah. Kanal warna diubah menjadi matriks dua dimensi, dan rata-rata data dikurangi untuk memusatkan nilai-nilai piksel. PCA kemudian dihitung menggunakan fungsi bernama pca yang menghasilkan komponen utama dan skor proyeksi. Kompresi citra dilakukan dengan mempertahankan hanya sejumlah komponen utama tertentu, sesuai nilai yang ditentukan oleh variabel numComponents. Citra kemudian direkonstruksi menggunakan komponen utama yang dipertahankan, dan nilai rata-rata ditambahkan kembali agar hasil rekonstruksi mendekati citra asli. Proses ini dilakukan untuk setiap kanal warna, dan hasilnya disimpan di matriks Img_comp. Dengan PCA, penggunaan kode ini memungkinkan pengurangan ukuran file citra sambil tetap menjaga sebagian besar informasi visual yang penting.



(a) Dataset



(b) Data Uji

Gambar 2. Dataset dan data uji

Hasil dari implementasi kode yang dibuat akan diuji menggunakan beberapa dataset atau data uji yang dimiliki, guna untuk melihat perbandingan kinerja dari segi efisiensi dalam kompresi ukuran citra dengan tiga algoritma yang digunakan yaitu DCT, DWT, dan PCA untuk kompresi citra lossy yang akan dilakukan. Berikut ini adalah contoh beberapa dataset yang digunakan dan tampilan GUI yang telah dibuat, beserta hasil uji perbandingan dari penggunaan ketiga algoritma DCT, DWT, dan PCA untuk kompresi citra lossy dengan penggunaan data uji yang diambil dari internet sebanyak 5 yang termasuk dalam dataset publik, data yang digunakan memiliki format file yang berbeda seperti jpg, png, dan bmp. Penjelasan lebih lanjut dapat dilihat pada Gambar 2 (a), Gambar 2 (b), Tabel 1, Tabel 2, dan Tabel 3.

Tabel 1. Uji data algoritma DCT citra lossy

Dataset	Citra Asli (Kb)	Citra Kompresi (Kb)	MSE	PSNR	SSIM	Kompresi (%)
alam.bmp	3198,8848	3198,8027	0,0056869	22,4513	0,82108	0,0025644
sapi.png	190,4199	90,2783	0,0025444	25,9442	0,8802	52,5899
rog.png	1314,4785	817,0449	0,00049694	33,037	0,92849	37,8427
mri.jpg	362,4609	42,0635	0,0098796	20,0526	0,76835	88,935
singa.jpg	4395,0039	1940	0,00128	28,9278	0,94316	55,859

Tabel 2. Uji data algoritma DWT citra lossy

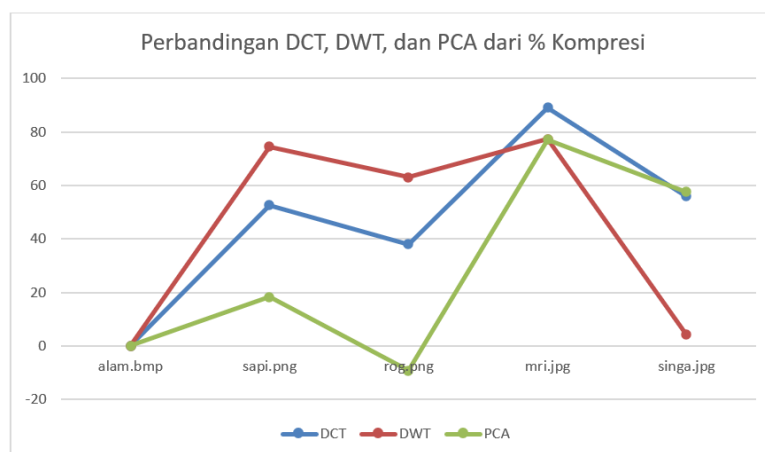
Dataset	Citra Asli (Kb)	Citra Kompresi (Kb)	MSE	PSNR	SSIM	Kompresi (%)
alam.bmp	3198,8848	3198,8027	0,0035281	24,5246	0,89276	0,0025644
sapi.png	190,4199	48,5908	0,001129	29,473	0,95162	74,4823
rog.png	1314,4785	483,7686	0,00027423	35,6189	0,96819	63,1969
mri.jpg	362,4609	82,6064	0,0039724	24,0095	0,92845	77,2096
singa.jpg	4395,0039	4211,0879	0,00060637	32,1726	0,97343	4,1847

Tabel 3. Uji data algoritma PCA citra lossy

Dataset	Citra Asli (Kb)	Citra Kompresi (Kb)	MSE	PSNR	SSIM	Kompresi (%)
alam.bmp	3198,8848	3198,8027	0,0054389	22,6449	0,80244	0,0025644
sapi.png	190,4199	155,7109	0,00051834	32,8538	0,91598	18,2276
rog.png	1314,4785	1435,583	0,00060824	32,1593	0,82027	-9,2131
mri.jpg	362,4609	83,1152	0,0046156	23,3577	0,70439	77,0692
singa.jpg	4395,0039	1871,6084	0,0031163	25,0637	0,88872	57,4151

Pada gambar 3 menunjukkan tampilan GUI berhasil di jalankan dan dapat difungsikan secara optimal untuk kelima dataset yang digunakan yang diambil dari internet. Sementara pada tabel 1 2, dan 3 dari lima data yang telah diuji dalam membandingkan penggunaan algoritma DCT, DWT, dan PCA pada kompresi citra lossy. Diketahui penggunaan algoritma DCT lebih efisien dalam kompresi citra dengan format JPG karena sifat algoritma ini yang secara khusus dirancang untuk memanfaatkan karakteristik visual manusia. DCT memfokuskan proses kompresinya pada penghapusan komponen frekuensi tinggi yang kurang signifikan secara visual, sehingga menghasilkan ukuran file yang lebih kecil tanpa mengorbankan kualitas visual secara drastis. Hal ini membuat algoritma DCT sangat cocok untuk format JPG, yang secara inheren menggunakan metode kompresi serupa.

Sebaliknya, algoritma DWT menunjukkan hasil yang lebih baik saat digunakan pada format PNG. Hal ini disebabkan oleh pendekatannya yang berbasis wavelet, yang mampu mempertahankan detail penting pada citra meskipun telah melalui proses kompresi. Dengan membagi citra menjadi beberapa level frekuensi, penggunaan algoritma DWT menjadi lebih fleksibel dalam menangani format gambar yang membutuhkan kejelasan detail, seperti citra dengan format PNG, sehingga menghasilkan kualitas kompresi yang lebih optimal.



Gambar 3. Grafik hasil uji perbandingan DCT, DWT, dan PCA

Untuk PCA, algoritma ini menunjukkan tingkat efisiensi kompresi citra yang bervariasi tergantung pada jenis format citra yang digunakan. Pada kasus tertentu, seperti dataset rog.png, algoritma ini bahkan menghasilkan persentase kompresi negatif yaitu -9,2131%, yang mengindikasikan bahwa penggunaan algoritma PCA tidak selalu efektif untuk semua jenis citra, terutama citra dengan struktur atau tekstur yang kompleks. Selain itu, kualitas citra hasil kompresi menggunakan algoritma PCA cenderung lebih rendah

dibandingkan algoritma DWT dan DCT, seperti yang terlihat dari nilai PSNR yang lebih rendah pada dataset mri.jpg yaitu 23,3577 dan penurunan nilai SSIM. Meskipun algoritma PCA memiliki kelebihan dalam analisis dimensi dan pengurangan redundansi data, algoritma ini kurang optimal jika diterapkan untuk kompresi citra, terutama jika kualitas visual harus dipertahankan dengan baik. Untuk memahami gambaran perbandingan dari algoritma DCT, DWT, dan PCA dapat dilihat melalui grafik gambar 3 diatas.

4. KESIMPULAN

Dalam penelitian ini, diperoleh kesimpulan bahwa algoritma DCT, DWT, dan PCA memiliki keunggulan masing-masing dalam kompresi citra lossy. Algoritma DCT menunjukkan efisiensi tinggi pada format gambar JPG dengan menghasilkan tingkat kompresi yang signifikan tanpa terlalu banyak mengorbankan kualitas visual. Hal ini menjadikannya cocok untuk aplikasi yang memerlukan ukuran file kecil, seperti pada pengiriman gambar melalui internet. DWT, di sisi lain, memberikan hasil terbaik untuk format PNG karena kemampuannya dalam mempertahankan detail citra, terutama pada bagian-bagian yang halus, sehingga kualitas visual tetap terjaga meski ukuran file berkurang. Sementara itu, algoritma PCA menunjukkan performa yang lebih bervariasi, dengan tingkat kompresi yang cenderung tidak konsisten, seperti kasus persentase kompresi negatif pada beberapa dataset. Namun, algoritma PCA memiliki kelebihan dalam reduksi dimensi data dan lebih cocok untuk aplikasi pengolahan data skala besar atau analisis statistik, di mana efisiensi pengelolaan data menjadi prioritas, meskipun kualitas visual citra terkompresi mengalami penurunan. Sebagai saran, dalam penelitian selanjutnya agar lebih untuk mengeksplorasi kombinasi dari beberapa algoritma, seperti menggabungkan DWT dan PCA, untuk memanfaatkan kelebihan masing-masing metode. Selain itu, dapat juga diterapkan teknik optimasi kuantisasi atau penggunaan wavelet dengan tipe yang lebih kompleks untuk meningkatkan efisiensi kompresi. Penelitian lebih lanjut juga diharapkan mengevaluasi performa algoritma pada format gambar lainnya, seperti TIFF atau BMP, untuk memberikan panduan lebih komprehensif dalam memilih algoritma kompresi yang sesuai dengan kebutuhan tertentu. Dengan demikian, diharapkan dapat ditemukan keseimbangan yang lebih baik antara ukuran file dan kualitas citra yang dihasilkan.

REFERENCES

- [1] M. Habek, Y. Genc, N. Aytas, A. Akkoc, E. Afacan, and E. Yazgan, "Digital image encryption using elliptic curve cryptography: A review," in *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, IEEE, 2022, pp. 1–8.
- [2] E. R. Pramudya *et al.*, "Optimization of image encryption using fractal Tromino and polynomial Chebyshev based on chaotic matrix," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 22, no. 6, p. 1529, Aug. 2024, doi: 10.12928/telkomnika.v22i6.26080.
- [3] N. Gupta and R. Vijay, "Hybrid image compression-encryption scheme based on multilayer stacked autoencoder and logistic map," *China Communications*, vol. 19, no. 1, pp. 238–252, 2022, doi: 10.23919/JCC.2022.01.017.
- [4] C. A. Sari, P. Purwanto, E. H. Rachmawanto, and A. Syukur, "An integration of quantum systems using BB84 for enhanced security in aeroponic smart farming," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 22, no. 6, p. 1491, Dec. 2024, doi: 10.12928/telkomnika.v22i6.26450.
- [5] N. R. D. Cahyo, C. A. Sari, E. H. Rachmawanto, C. Jatmoko, R. R. A. Al-Jawry, and M. A. Alkhafaji, "A Comparison of Multi Class Support Vector Machine vs Deep Convolutional Neural Network for Brain Tumor Classification," in *2023 International Seminar on Application for Technology of Information and Communication (iSemantic)*, IEEE, Sep. 2023, pp. 358–363. doi: 10.1109/iSemantic59612.2023.10295336.
- [6] R. Ezhilarasi, K. Venkatalakshmi, and B. P. Khanth, "Enhanced approximate discrete cosine transforms for image compression and multimedia applications," *Multimed Tools Appl*, vol. 79, pp. 8539–8552, 2020.
- [7] C. Umam, L. B. Handoko, C. A. Sari, E. H. Rachmawanto, and L. A. R. Hakim, "Kombinasi Vigenere dan Autokey Cipher dalam Proses Proteksi SMS Berbasis Android," *Prosiding Sains Nasional dan Teknologi*, vol. 12, no. 1, p. 492, Nov. 2022, doi: 10.36499/psnst.v12i1.7108.
- [8] A. D. Krismawan and E. H. Rachmawanto, "Principal Component Analysis (PCA) dan K-Nearest Neighbor (KNN) dalam Deteksi Masker pada Wajah," *Prosiding Sains Nasional dan Teknologi*, vol. 12, no. 1, p. 382, Nov. 2022, doi: 10.36499/psnst.v12i1.7066.
- [9] E. A. Sofyan, C. A. Sari, H. Rachmawanto, and R. D. Cahyo, "High-Quality Evaluation for Invisible Watermarking Based on Discrete Cosine Transform (DCT) and Singular Value Decomposition

- (SVD),” *Advance Sustainable Science, Engineering and Technology (ASSET)*, vol. 6, no. 1, 2024, doi: 10.26877/asset.v6i1.17186.
- [10] Mohammed Hassan Abd and Osamah Waleed Allawi, “Secured Mechanism Towards Integrity of Digital Images Using DWT, DCT, LSB and Watermarking Integrations,” *Ibn AL-Haitham Journal For Pure and Applied Sciences*, vol. 36, no. 2, pp. 454–468, Apr. 2023, doi: 10.30526/36.2.3088.
- [11] R. Noor, A. Khan, A. Sarfaraz, Z. Mehmood, and A. M. Cheema, “Highly robust hybrid image watermarking approach using Tchebichef transform with secured PCA and CAT encryption,” *Soft comput*, vol. 23, no. 20, pp. 9821–9829, Oct. 2019, doi: 10.1007/s00500-019-03838-2.
- [12] W. Y. Yang *et al.*, *Applied numerical methods using MATLAB*. John Wiley & Sons, 2020.
- [13] A. Tiwari, D. Awasthi, and V. K. Srivastava, “Image security enhancement to medical images by RDWT-DCT-Schur decomposition-based watermarking and its authentication using BRISK features,” *Multimed Tools Appl*, vol. 83, no. 22, pp. 61883–61912, 2024, doi: 10.1007/s11042-023-15878-w.
- [14] A. J. G. Busson, P. R. C. Mendes, D. de S. Moraes, A. M. da Veiga, A. L. V. Guedes, and S. Colcher, “Video Quality Enhancement Using Deep Learning-Based Prediction Models for Quantized DCT Coefficients in MPEG I-frames,” in *2020 IEEE International Symposium on Multimedia (ISM)*, IEEE, Dec. 2020, pp. 29–32. doi: 10.1109/ISM.2020.00012.
- [15] J. Khandelwal, V. K. Sharma, D. Singh, and A. Zaguia, “Dwt-svd based image steganography using threshold value encryption method,” *Computers, Materials and Continua*, vol. 72, no. 2, pp. 3299–3312, 2022, doi: 10.32604/cmc.2022.023116.
- [16] A. Alzahrani, “Enhanced invisibility and robustness of digital image watermarking based on DWT-SVD,” *Appl Bionics Biomech*, vol. 2022, 2022.
- [17] Priyanka and D. Kumar, “Feature Extraction and Selection of kidney Ultrasound Images Using GLCM and PCA,” in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 1722–1731. doi: 10.1016/j.procs.2020.03.382.
- [18] C. A. Sari, M. H. Dzaki, E. H. Rachmawanto, R. R. Ali, and M. Doheir, “High PSNR Using Fibonacci Sequences in Classical Cryptography and Steganography Using LSB,” *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 4, pp. 568–580, 2023, doi: 10.22266/ijies2023.0831.46.
- [19] C. A. Sari, E. H. Rachmawanto, and C. A. Haryanto, “Cryptography Triple Data Encryption Standard (3DES) for Digital Image Security,” *Scientific Journal of Informatics*, vol. 5, no. 2, pp. 2407–7658, 2018, [Online]. Available: <http://journal.unnes.ac.id/nju/index.php/sji>
- [20] P. N. Andono and C. A. Sari, “Remove Blur Image Using Bi-Directional Akamatsu Transform and Discrete Wavelet Transform,” *Scientific Journal of Informatics*, vol. 9, no. 2, pp. 179–188, Nov. 2022, doi: 10.15294/sji.v9i2.34173.