



Pengoptimalan Kecepatan Kompresi Citra Fraktal dengan Menggunakan Jarak Variansi Kuadrat

Akhmad Sultoni ^{1*} , Nanang Susyanto ²

¹ Department of Mathematics, Universitas Gadjah Mada, Indonesia

² Department of Mathematics, Universitas Gadjah Mada, Indonesia

* Corresponding Author. E-mail: akhmad.s@mail.ugm.ac.id

ARTICLE INFO

Article History:

Received: 20-Nov. 2024

Revised: 31-Jan. 2025

Accepted: 27-Mar.2025

Keywords:

Image compression,
fractal, square variance
distance, encoding,
decoding.

ABSTRACT

Kompresi citra fraktal terdiri dari dua tahap utama, yaitu encoding dan decoding. Proses encoding umumnya membutuhkan waktu yang lebih lama karena melibatkan pencocokan antara blok domain dan blok range, sedangkan decoding berlangsung lebih cepat. Untuk meningkatkan efisiensi encoding, metode dengan jarak variansi kuadrat atau root variance distance (SVD) dikembangkan sebagai pendekatan seleksi blok domain yang lebih optimal. Metode ini menghitung perbedaan variansi antara blok domain dan blok range, sehingga hanya blok domain dengan karakteristik variansi tertentu yang dipertimbangkan dalam proses pencocokan. Dengan cara ini, jumlah perbandingan blok dapat dikurangi secara signifikan, sehingga mempercepat proses kompresi. Pengujian menggunakan citra grayscale berukuran 256×256 piksel menunjukkan bahwa metode ini mampu mempercepat waktu encoding hingga 34 kali lebih cepat dibandingkan metode tanpa variansi, dengan kualitas citra hasil kompresi yang tetap terjaga pada Peak Signal-to-Noise Ratio (PSNR) sebesar 23.0 pada ukuran blok range 8×8.

Fractal image compression consists of two main stages: encoding and decoding. The encoding process generally takes longer because it involves matching domain blocks with range blocks, whereas decoding is faster. To improve the efficiency of encoding, the squared variance distance (SVD) method was developed as a more optimal approach for domain block selection. This method calculates the variance difference between domain and range blocks, allowing only domain blocks with specific variance characteristics to be considered in the matching process. As a result, the number of block comparisons is significantly reduced, accelerating the compression process. Tests on a 256×256 pixel grayscale image demonstrate that this method can speed up encoding by up to 34 times compared to the method without variance, while maintaining the compressed image quality at a Peak Signal-to-Noise Ratio (PSNR) of 23.0 with a block range size of 8×8.

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



How to Cite:

Sultoni, A., & Susyanto, N. (2025). Pengoptimalan Kecepatan Kompresi Citra Fraktal dengan Menggunakan Jarak Variansi Kuadrat. *PYTHAGORAS Jurnal Matematika Dan Pendidikan Matematika*, 20(1). <https://doi.org/10.21831/pythagoras.v20i1.79254>

<https://doi.org/10.21831/pythagoras.v20i1.79254>

PENDAHULUAN

Di era perkembangan teknologi dan informasi, internet telah menjadi kebutuhan esensial bagi kehidupan manusia. Internet kini menghubungkan miliaran individu di seluruh dunia. Menurut data yang dirilis oleh Statista, pada Juli 2020 terdapat 4,57 miliar pengguna internet aktif, dengan 3,96 miliar di antaranya aktif di media social. Jumlah pengguna yang besar ini meningkatkan kebutuhan akan ruang penyimpanan, terutama untuk citra digital (gambar). Oleh karena itu, diperlukan metode yang mampu mengurangi penggunaan memori penyimpanan citra sembari tetap menjaga kualitas gambar.

Teknik kompresi citra umumnya menggunakan pendekatan lossy, dengan beberapa metode yang telah banyak diterapkan, seperti transformasi kosinus diskrit (DCT), yang menjadi dasar kompresi JPEG (Hou et al., 2018), serta transformasi wavelet diskrit (Bouchemel et al., 2018). Namun, metode DCT memiliki keterbatasan karena bergantung pada resolusi, sehingga kurang efektif dalam menghasilkan citra terkompresi dengan kualitas tinggi. Metode lain, seperti kuantisasi vektor yang dijelaskan oleh Horng (2012) dan Rahebi (2022), mengadopsi pendekatan "building block" dengan membagi citra menjadi beberapa bagian kecil dan hanya menyimpan sejumlah kecil representasinya.

Sebagai alternatif, kompresi citra berbasis fraktal (*Fractal Image Compression*) bisa disingkat dengan FIC menawarkan keunggulan karena tidak bergantung pada resolusi citra (Fisher, 1995). Fraktal merupakan bentuk geometris yang memiliki sifat kesamaan diri (*self-similarity*) pada berbagai skala, sehingga citra yang dikompresi dengan metode ini tetap mempertahankan kualitasnya saat diperbesar tanpa munculnya efek blok yang sering terjadi pada teknik kompresi lainnya. Konsep fraktal pertama kali diperkenalkan oleh Mandelbrot pada 1975, yang menyatakan bahwa citra kompleks dapat dihasilkan melalui transformasi persamaan yang dilakukan secara iteratif (Nawaz et al., 2024). Barnsley kemudian mengembangkan ide ini lebih lanjut dengan memperkenalkan Iterated Function System (IFS), yaitu teknik yang memungkinkan pembentukan citra dengan pola self-similarity menggunakan parameter tertentu. Metode ini menjadi dasar bagi kompresi citra fraktal, karena memungkinkan penghematan ruang penyimpanan tanpa mengorbankan kualitas visual (Gupta et al., 2020). Oleh karena itu, penelitian ini berfokus pada eksplorasi lebih lanjut terhadap kompresi citra berbasis fraktal dan potensinya dalam meningkatkan efisiensi kompresi.

Kompresi citra berbasis fraktal terus berkembang seiring waktu, dengan berbagai pendekatan untuk meningkatkan efisiensinya. Salah satu kemajuan penting dalam bidang ini adalah kontribusi A. Jacquin, yang pertama kali memperkenalkan kompresi citra fraktal pada tahun 1992 (Roy et al., 2018). Jacquin mengusulkan bahwa citra tidak perlu disimpan sebagai kumpulan piksel, melainkan sebagai kombinasi transformasi afin berulang (Jacquin, 1992). Pendekatan ini memanfaatkan beberapa pemetaan afin yang diterapkan ke seluruh citra, membentuk serangkaian transformasi yang dikenal sebagai Sistem Fungsi Iteratif (*Iterated Function System*) disingkat dengan IFS.

Namun, penggunaan IFS dalam kompresi citra awalnya dianggap kompleks dan sulit untuk diterapkan secara praktis. Untuk mengatasi kendala ini, Jacquin kemudian memperkenalkan konsep IFS terpartisi (Partitioned Iterated Function System, PIFS), yang membuat metode kompresi citra fraktal lebih fleksibel dan efisien. Perbedaan utama antara PIFS dan IFS adalah bahwa dalam PIFS, citra dibagi menjadi beberapa kelompok kecil yang disebut blok range, sementara bagian citra sementara yang digunakan untuk membentuk blok range dikenal sebagai blok domain. PIFS diterapkan secara lokal pada setiap blok ini, bukan pada keseluruhan citra (Ahmed et al., 2020). Dengan pendekatan ini, kompresi citra fraktal menjadi lebih praktis dan efisien dalam mengurangi ukuran citra tanpa kehilangan terlalu banyak detail visual.

Kompresi citra fraktal memiliki keunggulan dalam merekonstruksi citra terkompresi tanpa kehilangan detail, terlepas dari ukuran yang digunakan (Revathy & Jayamohan, 2012). Metode ini dianggap efisien, sehingga menarik perhatian banyak peneliti untuk mengembangkan model yang dapat meningkatkan rasio kompresi (CR), independensi resolusi, serta mempercepat proses decoding (Sheeba & Rahiman, 2016). Teknik kompresi ini sangat cocok untuk berbagai aplikasi, termasuk solusi tanda tangan citra (Al-Saidi, 2012), segmentasi tekstur (Marrón, 2018), pencarian citra (Qin et al., 2015), serta penggunaan khusus dalam pemrosesan citra berbasis MRI dan ECG (Marusina et al., 2017). Selain itu, kompresi fraktal juga diterapkan dalam aspek pemrosesan citra lainnya, seperti penghilangan noise (Kachhia & Parmar, 2024) dan solusi watermarking digital (Awarayi et al., 2021).

Namun, kompresi citra fraktal menghadapi kelemahan utama, yaitu kompleksitas, waktu pengodean yang lama, dan tantangan dalam menemukan kecocokan blok terbaik di antara kumpulan blok domain yang besar (Ahmed et al., 2020). Oleh karena itu, penting untuk mempercepat proses pengodean, dan banyak upaya telah dilakukan untuk meningkatkan efisiensi kompresi citra fraktal dengan berbagai metode. Beberapa teknik fokus pada pengurangan waktu pengodean (Revathy & Jayamohan, 2012). Sebagai contoh, Zhu et al. (2010) mengusulkan algoritma yang memanfaatkan distribusi piksel dan triangulasi, menggunakan segitiga siku-siku sama kaki untuk membagi blok domain dan blok range. He Chuanjiang et al. (2007) memperkenalkan algoritma encoding citra fraktal cepat dengan metode block cross traces, sementara Wang Weiwei et al. (2017) mengembangkan

algoritma klasifikasi fitur menggunakan block library induk. Liu Shuqun et al. (2013) memperbaiki metode kompresi ini dengan menerapkan klasifikasi Fisher dan pemetaan spasial, dan Zhang Jing et al. (2013) mengusulkan algoritma kombinasi ganda serta klasifikasi fitur. Terakhir, Wang dan Liu (2020) mengembangkan algoritma berbasis fitur sentroid.

Meskipun berbagai metode kompresi citra telah dikembangkan, peningkatan kecepatan proses kompresi masih belum optimal. Salah satu faktor yang memengaruhi efisiensi kompresi adalah proses pemilihan blok domain sebelum pengujian kemiripan, yang sering kali membutuhkan waktu komputasi yang besar. Untuk mengatasi permasalahan ini, penulis mengusulkan metode baru yang memanfaatkan variansi dalam proses kompresi citra, dengan menerapkan *The Squared Variance Distance* (SVD). Metode ini bertujuan untuk menyaring blok domain berdasarkan perbedaan variansi, sehingga hanya blok dengan karakteristik yang paling relevan yang akan diproses lebih lanjut. Dengan cara ini, jumlah perhitungan yang diperlukan dalam tahap pengujian kemiripan dapat dikurangi secara signifikan, sehingga mempercepat proses kompresi tanpa mengorbankan kualitas hasil. Diharapkan, pendekatan ini dapat meningkatkan efisiensi waktu secara keseluruhan dan menjadi solusi yang lebih optimal dalam pengolahan citra digital.

METODE

Pada bagian ini, penulis membagi pembahasan menjadi tiga subbagian. Subbagian pertama membahas teori yang digunakan dalam algoritma pemrosesan, subbagian kedua mengulas algoritma yang diterapkan dalam proses kompresi citra fraktal, dan subbagian ketiga membahas metrik PSNR yang digunakan untuk mengukur kualitas citra yang dihasilkan dari proses kompresi.

TEORI YANG MENDASARI ALGORITMA

Sistem Fungsi Iterasi (SFI)

Pengkonstruksian fraktal tidak hanya dapat dilakukan dengan satu pemetaan, melainkan memerlukan sejumlah pemetaan, khususnya pemetaan kontraksi. Dalam konteks ini, kumpulan pemetaan kontraksi yang membentuk fraktal dikenal sebagai Sistem Fungsi Iterasi (SFI). Hal ini menunjukkan bahwa untuk membangun fraktal yang kompleks, berbagai pemetaan perlu diiterasikan secara sistematis. Perhatikan Definisi 1 yang penulis rujuk dari artikel Husain, A., et al (2022), Falconer (2014), dan Barnsley and Michael F. (2000).

Definisi 1 Misalkan (X, d) adalah metrik lengkap. Sistem Fungsi Iterasi (Hiperbolik) adalah koleksi berhingga pemetaan kontraksi $w_n: X \rightarrow X$, yang dinyatakan sebagai:

$$SFI = \{w_n : n = 1, 2, \dots, N\} \quad (1)$$

dengan faktor kontraksinya ditetapkan sebagai $s = \max \{s_n : n = 1, 2, \dots, N\}$. Istilah "Hiperbolik" dalam Definisi 1 diletakkan dalam tanda kurung karena terkadang kata tersebut diabaikan, lebih lengkap terkait hiperbolik lihat Navascues et al (2014). Pemetaan kontraksi berhingga yang membentuk SFI pada dasarnya sudah menunjukkan bahwa SFI bersifat Hiperbolik. Dengan kata lain, sifat kontraksi ini adalah kunci dalam menciptakan pola-pola fraktal yang menarik dan kompleks.

Teorema Collage

Teorema Collage, yang memberikan landasan matematis untuk memahami bagaimana fraktal dapat dibangun melalui iterasi fungsi kontraksi. Dalam konteks ruang metrik lengkap, Teorema Collage menjelaskan bahwa dengan menggunakan himpunan SFI, kita dapat mendekati atraktor fraktal dengan ketelitian yang diinginkan melalui jarak Hausdorff (Almeida, 2021).

Teorema 2. Diberikan (X, d) ruang metrik lengkap dan $L \in H(X)$. Untuk sebarang bilangan $\epsilon > 0$, diambil suatu $SFI = \{X: w_n, n = 1, 2, \dots, N\}$ dengan faktor kontraksi $0 \leq s < 1$ sehingga

$$h\left(L, \bigcup_{n=1}^N w_n(L)\right) \leq \epsilon$$

maka

$$h(L, A) \leq (1 - s)^{-1} h\left(L, \bigcup_{n=1}^N w_n(L)\right) \leq \frac{\epsilon}{1 - \epsilon} \quad (2)$$

dengan A adalah atraktor SFI. Teorema Collage menjelaskan bahwa jika jarak antara atraktor dan hasil gabungan dari fungsi kontraksi cukup kecil, maka kita dapat menghasilkan representasi yang akurat dari citra asli. Ini berarti bahwa fraktal dapat digunakan untuk menyimpan citra dengan lebih sedikit data, tanpa kehilangan detail yang signifikan.

Square Variance Distance (SVD)

Ukuran jarak variansi kuadrat merujuk pada artikel Landsman, Z., & Shushi, T. (2022), dan He *et al*, (2004) sebagai berikut:

Definisi 3. Diberikan *range block* $R \in \mathbb{R}^{m \times n}$ dan *domain block* $D \in \mathbb{R}^{m \times n}$, $var(R)$ didefinisikan sebagai square variance distance dari blok range R dan blok domain D sebagai berikut:

$$\begin{aligned} var(R) &= \frac{1}{m \times n} \left\| R - \bar{R} \cdot \mathbf{1} \right\|_2^2 \\ var(D) &= \frac{1}{m \times n} \left\| D - \bar{D} \cdot \mathbf{1} \right\|_2^2 \end{aligned} \quad (3)$$

di mana $\mathbf{1}$ adalah matriks konstan yang entri-entrinya 1, $\bar{R}$ dan $\bar{D}$ adalah nilai rata-rata dari blok range dan blok domain.

Root Mean Squared (RMS)

Ukuran mean squared error atau juga disebut jarak Euclidean kuadrat dirujuk pada artikel Hua *et al* (2019), Hossain, M., & Abufardeh, S. (2019), dan Wang & Bi (2016) sebagai berikut:

Definisi 4. Diberikan *range block* $R \in \mathbb{R}^{m \times n}$ dan *domain block* $D \in \mathbb{R}^{m \times n}$, RMS didefinisikan sebagai root mean squared error atau *squared euclidean distance* antara blok R dan blok D sebagai berikut:

$$RMS = \frac{1}{m \times n} \left\| (R - (sD + o\mathbf{1})) \right\|_2 \quad (4)$$

dengan

$$\begin{aligned} s &= \frac{\langle R - \bar{R}\mathbf{1}, D - \bar{D}\mathbf{1} \rangle}{(m \times n)var(D)} \\ o &= \bar{R} - s\bar{D} \end{aligned}$$

notasi $\|\cdot\|_2$ menunjukkan norm 2, s dan o berturut-turut merepresentasikan transformasi affine intensitas untuk penyesuaian kontras dan kecerahan. Notasi $\mathbf{1}$ adalah matriks konstan yang entri-entrinya 1, $\bar{R}$ dan $\bar{D}$ adalah nilai rata-rata dari blok range dan blok domain.

Hubungan SVD dan RMS

Teorema berikut memberikan representasi hubungan antara SVD dengan SED.

Teorema 5. Diberikan SVD pada definisi 3 dan RMS pada definisi 4, maka berlaku

$$RMS^2 = var(R) - s^2 var(D)$$

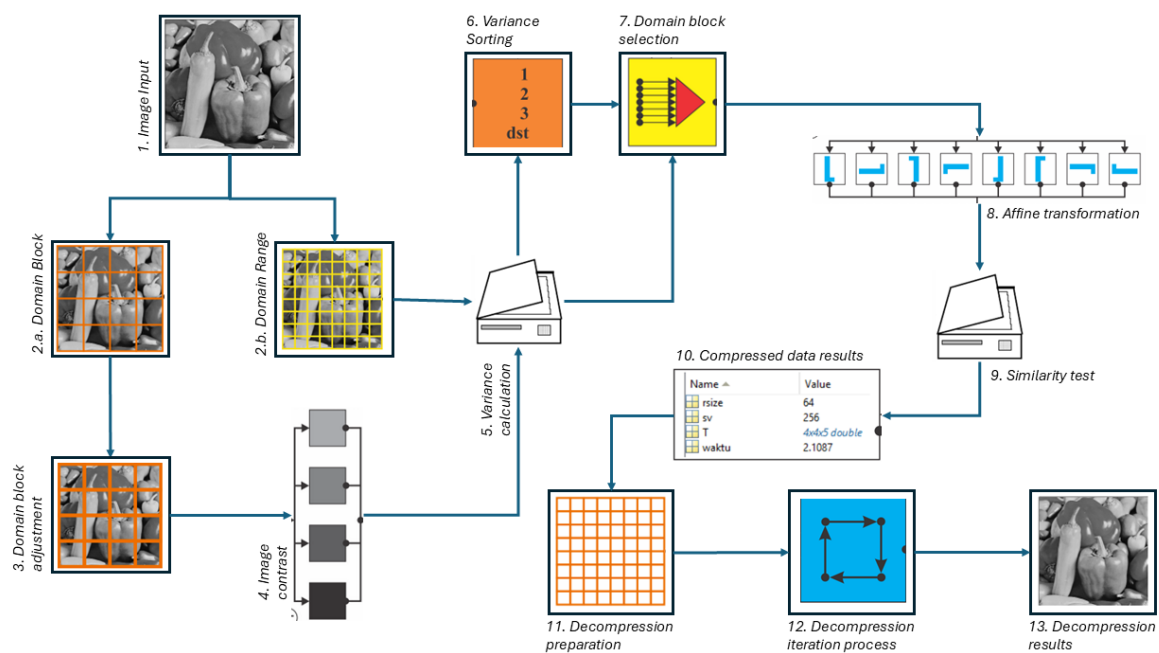
sehingga

$$RMS^2 \leq var(R) \quad (5)$$

Pertidaksamaan (5) ini menjadi dasar dalam pemilihan blok domain D , sehingga hanya menyisakan beberapa kandidat yang jauh lebih sedikit dibandingkan dengan keseluruhan blok domain. Penggunaan SVD sebelum melakukan uji kesamaan juga berperan penting dalam mempercepat waktu encoding secara signifikan.

ALGORITMA PROSES KOMPRESI CITRA FRAKTAL DENGAN SVD

Metode penelitian ini menggunakan pendekatan algoritmik yang sistematis, dibagi menjadi dua kelompok proses, encoding dan decoding. Gambaran proses kompresi citra fractal dengan SVD adalah disajikan dalam Gambar (1).



Gambar 1. Gambaran Proses Kompresi Citra Fraktal dengan SVD.

Terlihat bahwa kompresi citra fractal dengan SVD hampir sama dengan kompresi Metode Sederhana. Perbedaan antara kedua metode tersebut terdapat pada proses *encoding* yaitu terdapat penyeleksian blok domain D_i sebelum memasuki uji kemiripan. Sedangkan proses *decoding* sama persis dengan Metode Sederhana. Berikut algoritma encoding yang terdiri dari sepuluh langkah, yaitu:

1. **Input Citra (1):** Input citra dengan ukuran $N \times N$ sehingga menjadi bentuk *array* yang akan dikompresi.
2. **Blok Domain dan Blok Range (2):** Array citra dipartisi menjadi blok domain, dengan ukuran $2n \times 2n$, disimbolkan dengan $D_r, j = 1, 2, \dots, N_d$, $N_d = \left(\frac{N-2n}{\delta} + 1\right)^2$, δ merupakan perpindahan D_i ke D_{i+1} . Dan array citra dipartisi menjadi blok *range* yang tidak tumpang tindih, dengan ukuran $n \times n$, dilambangkan dengan $R_i, i = 1, 2, 3, \dots, N_r$, $N_r = \left(\frac{N}{n}\right)^2$.
3. **Sesuaikan Blok Domain (3):** Ukuran blok domain D_i disesuaikan dengan ukuran blok range.
4. **Hitung Kontras Citra (4):** Hitung empat kontras dari citra untuk setiap blok, yaitu kontras 0.45, 0.60, 0.80, dan 0.99.
5. **Perhitungan Variansi (5):** Hitung variansi semua blok domain dan blok range.
6. **Pengurutan Variansi (6):** Urutkan nilai variansi dari perhitungan Langkah (5).

$$\begin{cases} MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (R(i,j) - P(i,j))^2 \\ PSNR(R,P) = 10 \times \log_{10} \frac{255^2}{MSE} \end{cases} \quad (6)$$

di mana M dan N adalah ukuran citra asli, sedangkan m dan n mewakili ukuran citra yang dikompresi terenkripsi. R dan P masing-masing mewakili citra hasil rekonstruksi dan citra asli.

HASIL PENELITIAN

Proses encoding dan decoding kompresi citra fraktal dalam penelitian ini dilakukan menggunakan aplikasi pemrograman pada *personal computer* (PC) dengan spesifikasi AMD A6-9220 RADEON R4, 5 COMPUTE CORES 2C+3G, 2.50 GHz, RAM 4.00 GB, dan sistem operasi 64-bit. Penggunaan PC dapat mempengaruhi kecepatan waktu pemrosesan program. Namun, dalam pengujian ini, peneliti menggunakan PC yang sama untuk setiap pengujian, baik untuk Metode Sederhana maupun Metode Variansi, sehingga dapat meminimalkan error yang disebabkan oleh perbedaan perangkat.

Dalam bagian ini, pembahasan akan dibagi menjadi empat subbagian. Subbagian pertama adalah File Hasil Encoding, yang kedua adalah Ukuran Byte Citra Kompresi, yang ketiga Waktu Running Encoding, dan yang keempat Kualitas Citra Kompresi. Dua metode yang diuji adalah kompresi citra fraktal dengan metode sederhana, tanpa adanya penyeleksian variansi, dan kompresi citra fraktal dengan Metode Variansi, dengan menggunakan penyeleksian variansi. Objek penelitian berupa citra digital grayscale berukuran 256 piksel. Berikut akan dijelaskan secara rinci hasil dan pembahasan penelitian ini.

1. File Hasil Encoding

Misalkan diberikan suatu citra grayscale berukuran 256×256 piksel. Setelah dilakukan proses encoding, dihasilkan file berformat .mat yang memuat informasi tentang jumlah *looping* (disimbolkan dengan **lp**), ukuran blok range yang digunakan (**rsz**), ukuran citra (**sv**), parameter transformasi (**T**), dan waktu *running* (**waktu**). Isi file tersebut ditampilkan pada Gambar 3.

Name	Value	Min	Max
lp	49168	49168	49168
rsz	32	32	32
sv	256	256	256
T	8x8x5 double	-33.46...	111.30...
waktu	3.9630	3.9630	3.9630

doc. Akhmad Sultoni

Gambar 3. File Hasil Proses Encoding yang Disimpan dalam Format .mat

Gambar 3 menunjukkan bahwa parameter **lp** bernilai 49168, **rsz** bernilai 32, dan **sv** bernilai 256. Ini berarti file tersebut adalah hasil *encoding* menggunakan blok range 32×32 pada citra grayscale berukuran 256×256 piksel, dengan *looping* sebanyak 49168 kali. Dalam penelitian ini, digunakan beberapa ukuran **rsz**, yaitu 64×64 , 32×32 , 16×16 , 8×8 , dan 4×4 . Parameter **waktu** bernilai 3,9630, yang berarti waktu *running encoding* adalah 3,96 detik. Parameter utama, yaitu **T**, memiliki ukuran $8 \times 8 \times 5$ double, yang menunjukkan adanya 5 komponen penting, masing-masing disajikan dalam matriks berukuran 8×8 . Ukuran matriks ini diperoleh dengan membagi ukuran citra asli (**sv**) dengan ukuran blok range (**rsz**). Kelima komponen dalam **T** meliputi $lk = (x_0; y_0)$ yang merupakan lokasi blok domain D_i terbaik, w_i adalah transformasi terbaik, α_i adalah skala terbaik, dan β_i adalah faktor pencahayaan (*luminance*). Parameter **T** inilah yang akan menjadi generator dalam proses decoding untuk menghasilkan citra yang mendekati citra asli.

2. Ukuran File Citra Hasil Kompresi

Penelitian ini membandingkan tiga citra, yaitu Perahu, Babon, dan Orang, dengan ukuran masing-masing 47 KB, 57 KB, dan 65 KB terhadap file hasil encoding. Citra-citra tersebut diambil dari GitHub, yang merupakan sumber yang sering digunakan dalam penelitian kompresi gambar, dan disajikan pada Gambar 4.



Gambar 4. Tiga Citra Objek Penelitian

File hasil encoding ini disimpan dalam memori dan memerlukan proses decoding untuk mengembalikan citra yang mirip dengan aslinya. Oleh karena itu, perbandingan dilakukan antara ukuran citra asli dan file hasil encoding. Hasil disajikan pada Tabel 5.

Tabel 5. Perbandingan Ukuran Citra per Kilo Byte (KB)

Citra	KB	Ukuran citra dengan tanpa SVD				
		64 × 64	32 × 32	16 × 16	8 × 8	4 × 4
Perahu	47	1	2	3	8	31
Babon	57	1	2	3	8	31
Orang	65	1	2	3	8	31

Citra	KB	Ukuran citra dengan metode SVD				
		64 × 64	32 × 32	16 × 16	8 × 8	4 × 4
Perahu	47	1	1	3	7	23
Babon	57	1	1	2	6	21
Orang	65	1	1	3	7	24

Berdasarkan hasil pengujian yang tercantum dalam Tabel 5, dapat disimpulkan bahwa kompresi citra menggunakan kedua metode, yaitu Metode Sederhana dan Metode SVD, menunjukkan efisiensi yang signifikan dalam mengurangi ukuran file. Misalnya, citra Perahu yang memiliki ukuran awal 47 KB, dapat dikompresi menggunakan Metode Sederhana menjadi antara 1,52 hingga 47 kali lebih kecil, sedangkan dengan Metode SVD, file tersebut terkompresi menjadi antara 2,04 hingga 47 kali lebih kecil. Demikian pula, citra Babon dengan ukuran awal 57 KB, setelah kompresi dengan Metode Sederhana berkurang menjadi antara 1,84 hingga 57 kali lebih kecil, sedangkan Metode SVD mampu mengurangi ukuran file hingga 2,71 hingga 5,7 kali lebih kecil. Hasil yang serupa ditemukan pada citra Orang, yang berukuran 65 KB. Kompresi menggunakan Metode Sederhana menghasilkan pengurangan ukuran file antara 2,1 hingga 65 kali, sedangkan Metode SVD mengurangi ukuran file antara 2,71 hingga 65 kali. Hal ini menunjukkan bahwa kedua metode kompresi, baik Metode Sederhana maupun Metode SVD, sangat efektif dalam mengurangi ukuran file citra secara signifikan. Pengurangan ukuran file ini juga diperkirakan akan serupa jika diterapkan pada citra grayscale dengan resolusi 256 × 256 lainnya. Secara keseluruhan, hasil penelitian ini menegaskan bahwa penggunaan kompresi citra fraktal dapat secara efisien mengurangi ukuran file.

3. Waktu Running Encoding

Waktu pemrosesan pada tahap encoding dipengaruhi oleh beberapa faktor, antara lain spesifikasi PC, algoritma yang digunakan, dan ukuran piksel citra. Dalam penelitian ini, faktor PC diminimalkan dengan melakukan semua pengujian pada PC yang sama. Ukuran piksel citra juga dibuat konstan dengan menggunakan objek yang sama, yaitu citra grayscale berukuran 256 × 256 piksel, baik untuk Metode tanpa Variansi maupun Metode SVD. Oleh karena itu, perbandingan kecepatan encoding difokuskan pada aspek algoritma, terutama pada jumlah loop yang digunakan. Selain itu, penulis juga mengevaluasi ukuran blok range (rsize) karena hal ini berpengaruh langsung terhadap jumlah loop dalam proses encoding.

Metode tanpa perhitungan SVD adalah pendekatan tradisional dalam kompresi citra fraktal yang menguji kesamaan antara blok domain R_i yang telah ditransformasi affine dengan keseluruhan blok range R_i . Namun, metode ini memiliki kelemahan utama dalam hal efisiensi waktu, karena banyaknya loop yang harus dilakukan selama proses encoding. Dalam penelitian ini, penulis memperkenalkan metode yang lebih efisien dengan

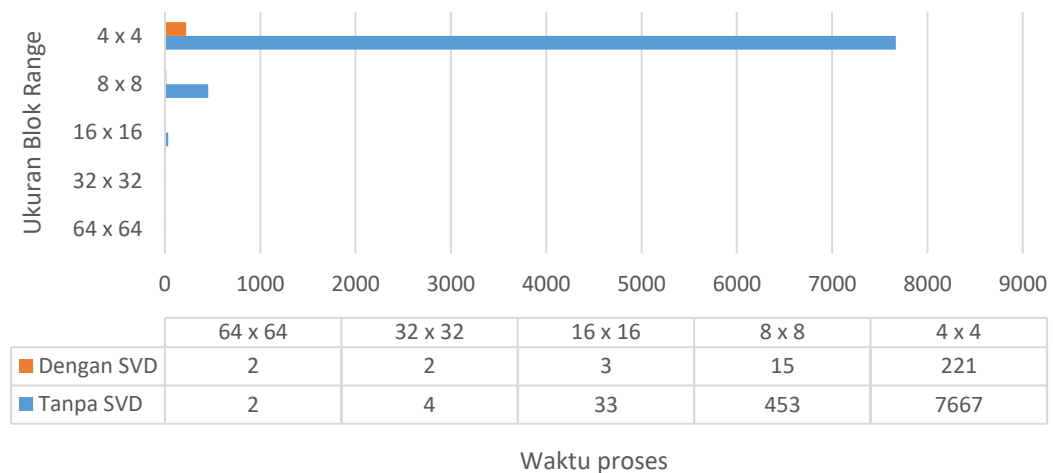
menggunakan SVD untuk menyaring blok domain D_i sebelum memasuki subproses uji kesamaan. Seleksi awal ini sangat penting karena memungkinkan pengurangan jumlah loop secara signifikan dibandingkan metode tanpa perhitungan SVD, sehingga mengoptimalkan kecepatan proses encoding secara substansial.

Tabel 6. Perbandingan banyaknya Loop Proses Encoding Kompresi Citra Fraktal

Ukuran Blok Range	Loop Metode tanpa SVD	Loop Metode dengan SVD	Rasio*
64×64	18436	16604	0,9006
32×32	49168	18533	0,3769
16×16	540736	54769	0,1013
8×8	8405248	666598	0,0793
4×4	134235136	11820337	0,0881

Rasio* := $\frac{\text{Metode dengan SVD}}{\text{Metode tanpa SVD}}$

Hasil penelitian menunjukkan keunggulan yang jelas dari metode menggunakan SVD, sebagaimana ditampilkan dalam Tabel 6. Jumlah loop yang diperlukan pada metode ini jauh lebih sedikit dibandingkan metode tradisional, yang diuji pada citra **babon.jpg** (Gambar 4). Selain itu, faktor ukuran blok range turut mempengaruhi jumlah loop yang diperlukan, semakin kecil ukuran blok range, semakin banyak loop yang dibutuhkan untuk kedua metode, namun metode dengan SVD tetap menunjukkan efisiensi yang lebih baik. Penulis juga mengevaluasi dampak ukuran blok range terhadap waktu pemrosesan encoding, dengan hasil yang divisualisasikan pada Gambar 7.



Gambar 7. Perbandingan waktu proses encoding

Ketika ukuran blok range adalah 64×64, kedua metode menunjukkan waktu pemrosesan encoding yang hampir setara. Namun, seiring pengecilan ukuran blok range, keunggulan metode yang menggunakan SVD semakin menonjol. Terbukti bahwa metode ini mampu mempercepat waktu pemrosesan encoding antara 2 hingga 34,7 kali lebih cepat dibandingkan metode tanpa perhitungan SVD.

4. Kualitas Citra Kompresi

Kualitas citra kompresi yang dimaksud di sini merujuk pada tingkat kesamaan antara citra hasil kompresi dengan citra aslinya. Mengukur kesamaan ini tidak mudah dilakukan secara langsung menggunakan panca indra, karena persepsi terhadap kesamaan citra bersifat subjektif. Untuk mendapatkan pengukuran yang objektif, diperlukan suatu metrik tertentu. Dalam penelitian ini, penulis menggunakan nilai Peak Signal-to-Noise Ratio (PSNR) yang didefinisikan pada Persamaan 6. PSNR dapat secara kuantitatif mengevaluasi perbedaan antara citra hasil rekonstruksi dan citra aslinya. Semakin tinggi nilai PSNR dalam satuan desibel (dB), semakin baik kualitas

rekonstruksi citra tersebut (Liu, X. *et al*, 2023). Semakin tinggi nilai PSNR, semakin baik kualitas citra fraktal yang dihasilkan.

Gambar 8. Perbandingan waktu proses dan Kualitas Citra

Ukuran Blok Range	Waktu Proses (<i>detik</i>)			Kualitas Citra (dB)		
	Tanpa SVD	SVD	Rasio*	Tanpa SVD	SVD	Rasio*
64×64	2	2	1	18,3	17,9	0,978
32×32	4	2	0,5	19,6	19,3	0,985
16×16	33	3	0,09	21,2	20,4	0,962
8×8	453	15	0,033	23,0	21,1	0,917
4×4	7667	221	0,029	27,5	23,0	0,836

Rasio* := $\frac{\text{Metode dengan SVD}}{\text{Metode tanpa SVD}}$

Hasil penelitian mengenai kualitas kompresi citra menggunakan Metode tanpa SVD dan Metode dengan SVD dapat dilihat pada Tabel 8. Tabel tersebut menunjukkan bahwa kualitas kompresi citra sangat dipengaruhi oleh ukuran blok range yang dipilih. Semakin kecil ukuran blok range, semakin baik hasil kualitas kompresi citra. Ketika kedua metode dibandingkan, Metode tanpa SVD memberikan kualitas kompresi yang lebih unggul dibandingkan Metode dengan SVD. Pada ukuran blok range 4×4, Metode tanpa SVD menghasilkan PSNR sebesar 27,4 dB, sementara Metode dengan SVD hanya mencapai PSNR sebesar 23,0 dB. Walaupun kualitas kompresi Metode dengan SVD sedikit lebih rendah, keunggulan utama metode ini terletak pada kecepatan kompresinya yang jauh lebih baik. Metode dengan SVD mampu mencapai kecepatan kompresi hingga 34 kali lipat lebih cepat daripada Metode tanpa SVD. Oleh karena itu, meskipun ada sedikit penurunan kualitas, Metode dengan SVD menawarkan efisiensi waktu kompresi yang signifikan.

SIMPULAN

Penelitian ini telah membandingkan performa kompresi citra fraktal antara Metode tanpa SVD dan Metode dengan SVD. Dengan menggunakan ukuran blok *range* yang sama, penelitian ini mampu menyajikan perbandingan yang adil antara kedua metode tanpa dipengaruhi faktor eksternal lain. Hasilnya menunjukkan bahwa Metode dengan SVD secara signifikan lebih efisien dalam hal waktu kompresi, dengan rasio waktu proses mencapai 0,029 atau sekitar 34,7 kali lebih cepat dibandingkan Metode tanpa SVD, sambil tetap mempertahankan kualitas citra yang dihasilkan. Pada tahap decoding, waktu proses tidak diperhitungkan karena algoritma yang digunakan pada kedua metode sama, sehingga waktu proses pada tahap ini juga sama.

Ke depan, penelitian ini dapat dikembangkan lebih lanjut. Saat ini, citra masih dipartisi menjadi blok *range* dan blok domain yang seragam dan berbentuk persegi. Penggunaan partisi dengan ukuran dan bentuk yang lebih bervariasi untuk blok *range* dan blok domain diharapkan dapat meningkatkan fleksibilitas metode kompresi. Selain itu, Kualitas citra yang dihasilkan oleh metode yang menggunakan SVD masih lebih rendah dibandingkan dengan metode tanpa SVD. Hal ini disebabkan oleh adanya pengurangan informasi citra akibat penerapan *threshold*. Pengembangan lebih lanjut pada Metode dengan SVD diharapkan dapat meningkatkan kualitas citra sehingga hasil kompresi yang dihasilkan lebih optimal.

DAFTAR PUSTAKA

- Ahmed, Z. J., George, L. E., & Abduljabbar, Z. S. (2020). Fractal image compression using block indexing technique: A review. *Iraqi Journal of Science*, 1798-1810. <https://doi.org/10.24996/ij.s.2020.61.7.29>
- Al-Saidi, N. M. (2012). An efficient signcryption method using fractal image coding scheme. *International Journal of Applied Mathematics and Informatics*, 6(4), 189-197.
- Awarayi, N. S., Appiah, O., Weyori, B. A., & Ninfaakang, C. B. (2021). A Digital Image Watermarking Using Dwt and L-shaped Tromino Fractal Encryption. *International Journal of Image, Graphics and Signal Processing*, 12(3), 33. <https://doi.org/10.5815/ijigsp.2021.03.03>
- Barnsley, Michael F. 2000. *Fractal Everywhere*. Edisi ke-2. Morgan Kaufman: San Fransisco, USA.
- Bouchemel, A., Abed, D., & Moussaoui, A. (2018). Enhancement of compressed image transmission in WMSNs using modified μ -nonlinear transformation. *IEEE Communications Letters*, 22(5), 934–937. <https://doi.org/10.1109/LCOMM.2018.2812821>
- Falconer, K. (2014). *Fractal geometry: mathematical foundations and applications*. John Wiley & Sons.
- Fisher, Y. (1995) Fractal image compression: theory and application. Springer Verlag, New York
- G. G. C. de Almeida, et al. (2021). "Collage theorems, invertibility and fractal functions." *Soft Computing*, vol. 25, no. 21, pp. 13649-13658. DOI: [10.1007/s00500-021-05544-6](https://doi.org/10.1007/s00500-021-05544-6).
- Gupta, R., Mehrotra, D., & Tyagi, R. K. (2020). Computational complexity of fractal image compression algorithm. *IET Image Processing*, 14(17), 4425-4434. <https://doi.org/10.1049/iet-ipr.2019.0489>
- He, C., Yang, S. X., and Huang, X. 2004. Variance-based accelerating scheme for fractal image encoding. *Electronics Letters*. 40(2): 115-116.
- He, C. J., & Shen, X. N. (2007). Improved cross trace-based algorithm for fractal image coding. *Chinese Journal of Computers*, 30(12), 2156–2163. <https://doi.org/10.3321/j.issn:0254-4164.2007.12.012>
- Hossain, M., & Abufardeh, S. (2019). A New Method of Calculating Squared Euclidean Distance (SED) Using pTree Technology and Its Performance Analysis. In CATA (pp. 45-54). <https://experts.umn.edu/en/publications/a-new-method-of-calculating-squared-euclidean-distance-sed-using->
- Horng, M. H. (2012). Vector quantization using the firefly algorithm for image compression. *Expert Systems with Applications*, 39(1), 1078-1091. <https://doi.org/10.1016/j.eswa.2011.07.108>
- Hou, D., Wang, H., Zhang, W., & Yu, N. (2018). Reversible data hiding in JPEG image based on DCT frequency and block selection. *Signal Processing*, 148, 41-47. <https://doi.org/10.1016/j.sigpro.2018.02.002>
- Husain, A., Nanda, M. N., Chowdary, M. S., & Sajid, M. (2022). Fractals: an eclectic survey, part-I. *Fractal and Fractional*, 6(2), 89. <https://doi.org/10.3390/fractalfract6020089>
- Jacquin, A. E. (1992). Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on image processing*, 1(1), 18-30. <https://doi.org/10.1109/83.128028>
- Kachhia, K. B., & Parmar, P. P. (2024). A novel fractional mask for image denoising based on fractal–fractional integral. *Partial Differential Equations in Applied Mathematics*, 11, 100833. <https://doi.org/10.1016/j.padiff.2024.100833>

- Landsman, Z., & Shushi, T. (2022). The location of a minimum variance squared distance functional. *Insurance: Mathematics and Economics*, 105, 64-78. <https://doi.org/10.1016/j.insmatheco.2022.03.006>
- Liu, S. Q., & Pan, Z. R. (2013). Fractal image coding method based on fisher classification and space mapping. *Journal of Computer Applications*, 33(12), 3552–3554. <https://doi:10.11772/j.issn.1001-9081.2013.12.3552>
- Liu, X., Tong, X., Zhang, M., Wang, Z., & Fan, Y. (2023). Image compression and encryption algorithm based on uniform non-degeneracy chaotic system and fractal coding. *Nonlinear Dynamics*, 111(9), 8771-8798. <https://doi.org/10.1007/s11071-023-08281-5>
- Marrón, B. (2018). Texture filters and fractal dimension on image segmentation. *Journal of Signal and Information Processing*, 9(3), 229-238. <https://doi.org/10.4236/jsip.2018.93014>
- Marusina, M. Y., Mochalina, A. P., Frolova, E. P., Satikov, V. I., Barchuk, A. A., Kuznetsov, V. I., ... & Tarakanov, S. A. (2017). MRI image processing based on fractal analysis. *Asian Pacific Journal of Cancer Prevention: APJCP*, 18(1), 51. <https://doi.org/10.22034/APJCP.2017.18.1.51>
- Nawaz, B., Ullah, K., & Gdawiec, K. (2024). Generation of Mandelbrot and Julia sets by using M-iteration process. *Chaos, Solitons & Fractals*, 188. <https://doi.org/10.1016/j.chaos.2024.115516>
- Qin, Z., Xiaoqing, H., & Wenbo, L. (2015). An effective image retrieval method based on fractal dimension using kernel density estimation. In *Advanced Computer and Communication Engineering Technology: Proceedings of the 1st International Conference on Communication and Computer Engineering* (pp. 987-999). Springer International Publishing. https://doi.org/10.1007/978-3-319-07674-4_92
- Rahebi, J. (2022). Vector quantization using whale optimization algorithm for digital image compression. *Multimedia Tools and Applications*, 81(14), 20077-20103. <https://doi.org/10.1007/s11042-022-11952-x>
- Revathy, K., & Jayamohan, M. (2012). Dynamic domain classification for fractal image compression. *International Journal of Computer Science & Information Technology*, 4(2), 95. <https://doi.org/10.5121/ijcsit.2012.4208>
- Roy, S. K., Kumar, S., Chanda, B., Chaudhuri, B. B., & Banerjee, S. (2018). Fractal image compression using upper bound on scaling parameter. *Chaos, Solitons & Fractals*, 106, 16-22. <https://doi.org/10.1016/j.chaos.2017.11.013>
- Sheeba, K., & Rahiman, A. (2016). Iteration less Wavelet-Fractal Image Compression Applicable in Cellular Mobile Communication System. *International Journal of Computer Applications*, 136(8), 40-44. <https://doi.org/10.5120/ijca2016908543>
- Wang, Q., & Bi, S. (2016). Prediction of the PSNR Quality of Decoded Images in Fractal Image Coding. *Mathematical Problems in Engineering*, 2016. <https://doi.org/10.1155/2016/2159703>
- Wang, L., & Liu, Z. L. (2020). A fractal image compression algorithm based on centroid features and important sensitive area classification. *Computer Engineering and Science*, 42(5), 869–876. <https://doi:10.3969/j.issn.1007-130X.2020.05.014>
- Wang, W. W., Zhang, A. H., & Tang, T. T. (2017). A fast fractal coding algorithm with feature of parent block. *Computer Technology and Development*, 27(4), 51–54. <https://doi:10.3969/j.issn.1673-629X.2017.04.012>
- Yang, J., Feng, X., Chen, Y., Yan, P., & Zhang, H. (2024). A secure fractal compression scheme based on irregular Latin square, Julia and 2D-FCICM. *Digital Signal Processing*, 155, 104725. <https://doi.org/10.1016/j.dsp.2024.104725>

- Zhang, J., Zhang, A. H., Wang, W. W., & Tang, T. T. (2013). Investigation on fast fractal image encoding with sum of double cross eigenvalues. *Computer Technology and Development*, 27(3), 159–162. <https://doi:10.3969/j.issn.1673-629X.2017.03.033>
- Zhao, Q., Zhang, H., & Yang, S. (2021). Hybrid wavelet-fractal image compression algorithm. *IEEE Transactions on Image Processing*, 30, 5462-5475.
- Zhu, Z. L., Zhao, Y. L., & Yu, H. (2010). Efficient fractal image compression based on pixels distribution and triangular segmentation. *Journal of Computer Applications*, 30(2), 337–340. <https://doi:JournalArticle/5af5a07fc095d718d827576f>