

Perbandingan Kinerja Algoritma Prefix Code C2 dan LZW dalam Mengompresi File Gambar

Kevin Raihan¹

¹Universitas Budidarma, Sisingamangaraja, Sumatera Utara, Indonesia
E-Mail : raihankev5@gmail.com¹

Pristiwanto²

²Universitas Budidarma, Sisingamangaraja, Sumatera Utara, Indonesia
E-Mail : 4nt0.82@gmail.com²

ABSTRACT

Data compression plays a crucial role in reducing file sizes, particularly for image files, to conserve storage space and accelerate data transmission. This study evaluates the performance of two widely-used compression algorithms: Prefix Code C2 and LZW, focusing on their effectiveness in compressing image files. The research involves a comparative analysis of the Prefix Code C2 and LZW algorithms based on several performance metrics, such as file size post-compression, compression speed, and decompression speed. The experimental results highlight that each algorithm has specific strengths and weaknesses depending on the image type and characteristics. The Prefix Code C2 algorithm demonstrates superior compression ratios for images with high levels of repetition, whereas the LZW algorithm outperforms in both compression and decompression speed for images with more intricate color variations. These insights can assist in selecting the most suitable compression algorithm for different types of image files.

Keywords: Data Compression, Prefix Code C2 Algorithm, LZW Algorithm, Image Files

PENDAHULUAN

Kompresi data berperan penting dalam pertukaran dan penyimpanan data karena membantu meningkatkan efisiensi dan efektivitas. Di era digital saat ini, penggunaan serta pertukaran file digital sudah menjadi hal yang lumrah. Namun, ukuran file digital yang besar seringkali menghabiskan ruang penyimpanan dan membutuhkan waktu lebih lama untuk ditransfer. Oleh karena itu, penggunaan algoritma kompresi untuk mengecilkan ukuran file menjadi sangat penting [1]. Terdapat berbagai algoritma kompresi yang dapat digunakan untuk memperkecil ukuran file, di antaranya adalah algoritma Prefix Code C2 dan LZW (Lempel-Ziv-Welch). Kedua algoritma ini memiliki pendekatan yang berbeda dalam mengompresi data dan kinerjanya dapat bervariasi, bergantung pada jenis data yang dikompresi.

Dalam penelitian ini, perbandingan kinerja antara dua algoritma kompresi, yaitu Prefix Code C2 dan LZW, akan dilakukan untuk mengompresi file gambar. Kinerja kedua algoritma akan dievaluasi menggunakan beberapa metrik, seperti ukuran file setelah kompresi, kecepatan kompresi, dan kecepatan dekompresi. Latar belakang penelitian ini didasarkan pada pentingnya membandingkan

dua metode kompresi data. penerapan metode kompresi data berguna untuk menemukan metode kompresi yang tepat dan optimal untuk digunakan [2].

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja kedua algoritma kompresi dalam konteks file gambar. Diharapkan hasil penelitian dapat memberikan pemahaman mengenai algoritma mana yang lebih efektif dan efisien dalam situasi tertentu. Temuan ini berpotensi memiliki dampak praktis yang signifikan, terutama bagi aplikasi yang memerlukan penggunaan intensif file gambar, seperti di industri multimedia, pengolahan citra medis, dan komunikasi visual pada umumnya.

BAHAN DAN METODE

2.1 Bahan Penelitian

Penelitian ini menggunakan beberapa file gambar digital sebagai dataset yang dikompresi. Dataset terdiri dari gambar dengan berbagai format, ukuran dan kompleksitas warna seperti gambar dengan pola berulang dan variasi warna yang kompleks.

2.2 Metode Penelitian

Berikut merupakan tahapan yang dilakukan dalam penelitian ini.

1. Proses Kompresi

Setiap file gambar dikompresi menggunakan kedua algoritma, yaitu Prefix Code C2 dan LZW. Proses kompresi selesai, ukuran file hasil kompresi dicatat untuk analisis lebih lanjut.

2. Pengukuran Kinerja

Mengukur perubahan ukuran file sebelum dan sesudah kompresi untuk menghitung rasio kompresi. Waktu yang dibutuhkan setiap algoritma untuk mengompresi file gambar akan diukur dan dibandingkan. Waktu yang dibutuhkan untuk mengembalikan file gambar yang telah dikompresi ke bentuk aslinya juga akan dicatat.

Analisis Data

Kinerja kedua algoritma dievaluasi berdasarkan hasil kompresi dan dekompresi pada gambar dengan karakteristik yang berbeda, untuk menentukan situasi di mana masing-masing algoritma lebih unggul.

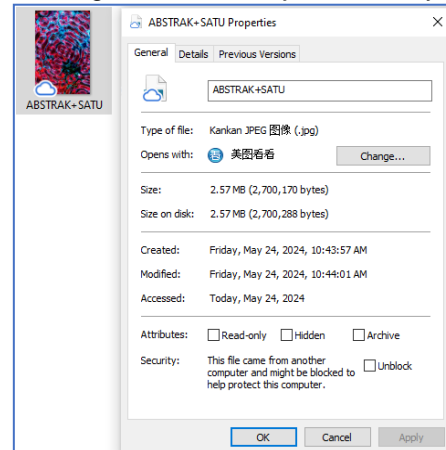
HASIL DAN PEMBAHASAN

Tahapan kompresi dengan algoritma prefix code C2 dengan LZW hingga proses analisis kinerja menggunakan pendekatan eksponensial penulis uraikan dalam bentuk tahapan dan gambarkan dalam bentuk diagram, sebagai berikut:

1. Menetapkan dan mengakses file gambar jenis ekstensi *.JPG.
2. Melakukan ekstraksi nilai *hexadecimal* terhadap objek gambar menggunakan aplikasi *binary view* untuk digunakan dalam proses penerapan rumus algoritma prefix code C2 dan LZW.
3. Melakukan penerapan rumus dan tahapan prefix code C2 terhadap nilai *hexadecimal* file gambar yang digunakan sebagai data sampel untuk proses kompresi.
4. Melakukan penerapan rumus dan tahapan LZW terhadap nilai *hexadecimal* file gambar yang digunakan sebagai data sampel untuk proses kompresi.
5. Masing-masing hasil kompresi prefix code C2 dan LZW kemudian di analisis perbandingan kinerjanya dengan menggunakan pendekatan metode eksponensial.
6. Menuliskan kesimpulan terkait dengan algoritma kompresi dengan kinerja kompresi terbaik antara prefix code C2 dan LZW dari hasil proses penerapan metode eksponensial

Gambar 1. Properties Sampel Gambar

Contoh data berupa tampilan nilai hexadecimal dari file gambar di atas diambil untuk menyederhanakan proses. Penulis memilih 16 nilai hexadecimal yang berada di alamat 0001A0 sebagai sampel data, di mana setiap byte terdiri dari 8 bit. Nilai hexadecimal yang digunakan sebagai sampel data adalah "20 43 6F 6D 70 61 6E 79 00 00 64 65 73 63 00 00". Dengan demikian, jumlah bit yang



akan diproses dapat dihitung, yaitu $16 \times 8 \text{ bit} = 128 \text{ bit}$. Jika dikonversi ke dalam satuan byte, hasilnya adalah $128/8 = 16 \text{ byte}$. Setelah mendapatkan nilai hexadecimal dari data gambar, nilai tersebut kemudian dapat disajikan dalam tabel di bawah ini.

Tabel 1. Frekuensi Nilai Hexa Sampel Gambar

Nilai Hexa	Frek	ASCII Decimal	Biner	Bit	Frek X Bit
20	1	32	00100000	8	8
43	1	67	01000011	8	8
6F	1	111	01101111	8	8
6D	1	109	01101101	8	8
70	1	112	01110000	8	8
61	1	97	01100001	8	8
6E	1	110	01101110	8	8
79	1	121	01111001	8	8
00	4	0	00000000	8	32
64	1	100	01100100	8	8
65	1	101	01100101	8	8
73	1	115	01110011	8	8
63	1	99	01100011	8	8
Jumlah bit x frekuensi					128 bit

Pembacaan nilai pada file gambar *.JPG dilakukan berdasarkan sampel nilai yang telah ditetapkan, yaitu 20 43 6F 6D 70 61 6E 79 00 00 64 65 73 63 00 00. Setelah proses pembacaan nilai selesai, dibuat tabel yang menunjukkan frekuensi kemunculan masing-masing nilai tersebut.

Tabel 2. Nilai Hexa Yang Belum dikompresi

Nilai Hexadecimal	Frekuensi Kemunculan
20	1
43	1
6F	1
6D	1

70	1
61	1
6E	1
79	1
00	4
64	1
65	1
73	1
63	1
Jumlah	16

Pembuatan kode bilangan menggunakan prefix code telah dijelaskan pada bagian landasan teori di bab sebelumnya. Dalam penelitian ini, prefix code yang digunakan adalah versi C2.

Tabel 3. Nilai Hexa dikompresi dengan kode C2

Nilai Hexa	Frek	Codeword (C1)	Bit	Frekuensi x Bit
00	4	0	1	4
20	1	100	3	3
43	1	110	3	3
6F	1	10100	5	5
6D	1	10110	5	5
70	1	11100	5	5
61	1	11110	5	5
6E	1	1010100	7	7
79	1	1010110	7	7
64	1	1011100	7	7
65	1	1011110	7	7
73	1	1110100	7	7
63	1	1110110	7	7
Total				72 bit

File gambar berhasil dikompresi menggunakan algoritma Prefix Code C2, dan format file tersebut diubah sesuai standar yang telah ditetapkan. Perubahan format ini menandakan bahwa file tersebut telah melalui proses kompresi dengan aplikasi yang menggunakan algoritma Prefix Code C2. Untuk mengevaluasi kinerja algoritma Prefix Code C2 dalam proses kompresi, penulis akan mengukur hasil kompresi file *.JPG berdasarkan parameter yang ditentukan. Semakin kecil ukuran file hasil kompresi, semakin baik kinerja kompresi, begitu pula sebaliknya.

Berikut merupakan parameter untuk mengukur kinerja algoritma Prefix Code C2.

1. *Ratio Of Comprassion (RC)*
 $RC = \frac{128}{72} = 1,778$
2. *Compression Ratio (CR)*
 $CR = \frac{72}{128} \times 100\% = 56,25\%$
3. *Redudancy (Rd)*

$$Rd = \frac{128-72}{128} \times 100\% = 43,75\%$$

4. *Space Saving (SS)*
 $SS = 100\% - CR = 100\% - 56,25\% = 43,75\%$

File gambar berhasil dikompresi menggunakan algoritma LZW, dan format file tersebut akan disesuaikan dengan standar yang berlaku. Perubahan format ini menandakan bahwa file telah melalui proses kompresi menggunakan aplikasi yang menerapkan algoritma LZW. Format file yang digunakan adalah *.JPG. Untuk menilai kinerja algoritma LZW dalam proses kompresi, penulis akan mengukur hasil kompresi file gambar *.JPG berdasarkan parameter yang telah ditentukan. Semakin kecil ukuran file setelah kompresi, semakin baik kinerja kompresinya, dan sebaliknya.

1. *Ratio Of Comprassion (RC)*

$$RC = \frac{128}{110} = 1,16$$

2. *Compression Ratio (CR)*

$$CR = \frac{110}{128} \times 100\% = 86\%$$

3. *Redudancy (Rd)*

$$Rd = \frac{128-110}{128} \times 100\% = 14\%$$

4. *Space Saving (SS)*

$$SS = 100\% - CR = 100\% - 86\% = 14\%$$

Menentukan prioritas keputusan berdasarkan nilai dari masing-masing alternatif. Hasil penentuan prioritas keputusan membandingkan kedua algoritma seperti tabel keputusan berikut ini.

Tabel 4. Prioritas Keputusan

Alternatif	Total Nilai	Rangking
Algoritma Prefix Code C2	3750541,33	1
Algoritma Lempel Ziv Welch	48557,16	2

Berdasarkan analisis perbandingan kinerja antara algoritma Prefix Code C2 dan algoritma Lempel-Ziv-Welch menggunakan metode eksponensial, dapat disimpulkan bahwa algoritma Prefix Code C2 merupakan pilihan yang lebih efektif dan optimal dalam mengompresi file gambar dibandingkan dengan algoritma Lempel-Ziv-Welch (LZW).

KESIMPULAN

Penelitian ini dengan topik "Perbandingan Kinerja Algoritma Prefix Code C2 dan LZW dalam Mengompresi File Gambar" menghasilkan beberapa kesimpulan sebagai berikut:

1. Algoritma Prefix Code C2 menunjukkan efisiensi kompresi yang lebih baik dalam mengurangi ukuran file gambar dibandingkan dengan LZW. Berdasarkan

- perbandingan yang dilakukan menggunakan metode eksponensial, Algoritma Prefix Code C2 menempati posisi teratas.
2. Algoritma LZW memiliki kecepatan kompresi yang lebih tinggi dibandingkan dengan Prefix Code C2, sehingga lebih sesuai untuk aplikasi yang membutuhkan proses kompresi cepat meskipun tidak mengutamakan ukuran file yang paling kecil.
 3. Kualitas gambar hasil dekompresi dari kedua algoritma tidak menunjukkan perbedaan yang signifikan. Namun, untuk gambar dengan detail tinggi, Prefix Code C2 cenderung lebih stabil dalam mempertahankan kualitas.

UCAPAN TERIMKASIH

Penulis mengucapkan terimakasih ke pembimbing yang telah banyak memberikan masukan dalam penyelesaian penelitian ini. Ucapan terima kasih juga penulis ucapkan ke Dekan Fakultas Ilmu Komputer setra program studi yang telah memberikan kesempatan untuk melakukan penelitian ini. Terimakasih juga penulis ucapkan kepada rekan-rekan yang telah membantu penyelesaian penelitian ini.

DAFTAR PUSTAKA

- [1] K. Sayood, *Introduction to data compression*. Inggris: Morgan Kaufmann Publishers, 2000.
 - [2] E. S. Sitorus, "Analisis Perbandingan Algoritma Goldbach G0 Dan Algoritma Goldbach G1 Dalam Mengkompresikan File Gambar," *Nasional Teknologi Informasi dan Komputer*, vol. 6, no. 1, 2022, doi: 10.30865/komik.v6i1.5684.
 - [3] D. Asdini and D. Putro Utomo, "Analisis Perbandingan Kinerja Algoritma Huffman Dan Algoritma Levenstein Dalam Kompresi File Dokumen Format .RTF," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 6, no. 1, 2022, doi: 10.30865/komik.v6i1.5739.
 - [4] A. Pradana, "Analisa Perbandingan Algoritma Elias Gamma Code Dan Algoritma Goldbach Code Pada Kompresi File Dokumen," *Nasional Teknologi Informasi dan Komputer*, vol. 6, no. 1, 2022, doi: 10.30865/komik.v6i1.5741.
 - [5] A. Hartama, "Analisis Perbandingan Algoritma Prefix Code Dengan Elias Omega Code Dalam Merancang Aplikasi Pengkompresi File Video Avi," *Nasional Teknologi Informasi dan Komputer*, vol. 6, no. 1, 2022, doi: 10.30865/komik.v6i1.5683.
 - [6] A. N. Alim, H. Yuana, dan F. Febrinita, "Aplikasi Kompresi Citra Dengan Menggunakan Algoritma Lempel Ziv Welch (LZW)," 2022.
 - [7] F. Fauzan, "Penerapan Algoritma Additive Code Dalam Mengkompresi Record Database," *Jurnal Sains dan Teknologi Informasi*, vol. 3, no. 2, hlm. 68–78, Mar 2024, doi: 10.47065/jussi.v3i2.4799.
 - [8] I. Arizki, A. Triawan, dan F. Zayid, "Penerapan Algoritma Lempel Ziv Welch (LZW) Untuk Kompresi Data," vol. 14, no. 1, hlm. 66–73, 2024, doi: 10.36350/jbs.v14i1.
 - [9] M. Alfarizi dan S. Aripin, "Penarapan Algoritma Prefix Code Dalam Kompresi File Video," vol. 4, no. 1, 2020, doi: 10.30865/komik.v4i1.2686.
 - [10] A. Sihotang, "Implementasi Algoritma Prefix Codes untuk Kompresi File Video Hasil Ekstra Aplikasi Kinemaster," 2022. [Daring]. Tersedia pada: <https://journal.grahamitra.id/index.php/biostech>
 - [11] E. N. Simanjuntak, "Penerapan Algoritma Prefix Code Pada Kompresi File Gambar ARTICLE INFO," *Media Online*, vol. 1, no. 3, hlm. 96–100, 2021.
- K. Y. Sarumaha, M. Syahrizal, dan E. R. Siagian, "Analisis Perbandingan Algoritma Elias Delta Code Dengan Algoritma Prefix Code Dalam