

Analisa Perbandingan Algoritma Goldbach dan LZW pada File WAV

Muhammad Bima Arya Irawan

Universitas Budidarma, Sisingamangaraja, Sumatera Utara, Indonesia

E-Mail : bima.bimaaryawan111111@gmail.com

Pristiwanto²

² Universitas Budidarma, Sisingamangaraja, Sumatera Utara, Indonesia

E-Mail : 4nt0.82@gmail.com²

ABSTRACT

WAV stands for Waveform Audio Format, an audio file standard developed by Microsoft and IBM. It is a variant of the RIFF bitstream format, similar to the IFF and AIFF formats used on Amiga and Macintosh systems. Both WAV and AIFF are compatible with Windows and Macintosh platforms. Although WAV files can contain compressed audio, they are typically used for uncompressed audio. The WAV format is part of Microsoft's RIFF specification for multimedia file storage. Compression is the process of reducing file size without significantly compromising data quality. The effectiveness of compression is evaluated based on factors such as compression time, memory usage, output quality, and the final file format. This research suggests that compressing audio files can help optimize hard disk space. Various compression algorithms are available, and reducing audio file size can enhance storage efficiency.

Keywords: Compression, WAV Audio Files, Goldbach, LZW

PENDAHULUAN

Format WAV (*.WAV) adalah salah satu format file audio yang umum digunakan pada sistem operasi Windows, terutama untuk keperluan game dan multimedia. WAV merupakan format audio mentah, di mana suara direkam langsung dan diubah menjadi bentuk digital melalui proses kuantisasi. Format ini sederhana untuk dibuat dan diproses, namun secara default tidak mendukung kompresi, sehingga dikenal dengan nama PCM (Pulse Code Modulation). Tantangannya adalah ukuran file WAV yang sangat besar, sehingga jarang digunakan di internet. Misalnya, sebuah lagu dengan kualitas asli CD audio (stereo, 16-bit, 44.100 Hz) akan membutuhkan ruang penyimpanan sekitar 176.400 byte per detik [1][2]. Oleh karena itu, file audio WAV berdurasi 60 detik dapat memakan sekitar 10,584 MB, dan untuk lagu berdurasi 4 menit, kapasitas yang dibutuhkan bisa mencapai 40 MB. Kompresi adalah proses mengurangi ukuran data dari yang besar menjadi lebih kecil tanpa mengorbankan kualitas data secara signifikan. Beberapa parameter yang menentukan kualitas kompresi meliputi waktu yang dibutuhkan untuk proses kompresi, efisiensi penggunaan memori, kualitas hasil kompresi, dan format keluaran yang dihasilkan[3][4][5]. Banyak metode yang dapat digunakan untuk melakukan kompresi file, akan tetapi penelitian ini untuk melakukan perbandingan metode dalam melakukan kompresi file WAV. Berbagai jenis file dapat

dikompresi untuk mengurangi ukuran dan menghemat ruang penyimpanan atau mempermudah pengiriman. Kompresi pada setiap tipe file dapat menggunakan metode lossless (tanpa kehilangan kualitas) atau lossy (dengan pengorbanan kualitas), tergantung pada format dan kebutuhan pengguna [6][7][8]. Penelitian ini bisa memberikan kontribusi pada pemahaman cara kerja algoritma dalam konteks yang berbeda dari biasanya. Ini juga bisa mendorong penelitian lebih lanjut dalam aplikasi algoritma non-tradisional dalam pemrosesan sinyal atau kompresi data. Dengan membandingkan algoritma yang sangat berbeda, dapat mendapatkan wawasan yang lebih dalam tentang bagaimana dan mengapa algoritma tertentu bekerja dengan cara tertentu dalam konteks aplikasi yang berbeda. Ini bisa membuka pintu untuk inovasi baru dalam teknik kompresi atau pengolahan data.

METODE PENELITIAN

2.1 Algoritma Goldbach

Algoritma Goldbach merujuk pada algoritma yang digunakan untuk membuktikan atau memverifikasi Konjektur Goldbach, salah satu masalah terkenal dalam matematika. Konjektur Goldbach, yang diajukan oleh matematikawan Christian Goldbach pada 1742, menyatakan bahwa setiap bilangan bulat genap yang lebih besar dari 2 dapat dinyatakan sebagai jumlah dua bilangan prima. Algoritma Goldbach bekerja berdasarkan *Hipotesis Goldbach*, yang

merupakan sebuah dugaan matematika yang menyatakan bahwa setiap bilangan genap lebih besar dari 2 dapat dinyatakan sebagai jumlah dari dua bilangan prima. Algoritma ini tidak membuktikan hipotesis tersebut, tetapi mencoba menemukan dua bilangan prima yang menjumlahkan bilangan genap tertentu[9].

2.2 Algoritma LZW

Sedangkan Algoritma LZW (Lempel-Ziv-Welch) adalah algoritma kompresi data yang populer dan tanpa kehilangan (lossless), dikembangkan oleh Terry Welch pada tahun 1984. Algoritma ini merupakan penyempurnaan dari metode LZ78 yang diciptakan oleh Abraham Lempel dan Jacob Ziv pada akhir 1970-an. LZW bekerja dengan menggantikan pola berulang dalam data dengan kode yang lebih pendek, mengurangi ukuran data tanpa mengorbankan kualitas. Algoritma LZW beroperasi dengan prinsip membangun kamus (dictionary) yang berisi pola karakter (string) yang sering muncul dalam data. Ketika pola ditemukan lagi, algoritma menggantinya dengan entri yang sesuai dari kamus, sehingga mengurangi ukuran keseluruhan data. Tabel kode awal berisi semua simbol dasar (karakter) dari data yang akan dikompresi, masing-masing dengan kode unik. Misalnya, untuk data berbasis teks, tabel awal akan berisi semua karakter ASCII standar dengan kode 0 hingga 255 [10].

HASIL DAN PEMBAHASAN

Tahap awal yang dilakukan melibatkan pengambilan nilai heksadesimal menggunakan aplikasi binary viewer, kemudian melanjutkan dengan proses kompresi sesuai dengan aturan dan langkah-langkah algoritma yang digunakan. Setelah proses kompresi selesai, hasil kompresi dapat dikembalikan ke nilai awal melalui proses dekompresi.

Masalah umum yang sering terjadi adalah ukuran file WAV yang cukup besar, yang dapat menyebabkan proses transmisi data memerlukan waktu lama dan alokasi penyimpanan yang besar. Oleh karena itu, dengan melakukan kompresi pada file WAV, dapat membantu mengurangi ruang penyimpanan dan mempercepat transmisi data. Untuk ukuran file WAV yang akan dikompres sebesar 40 MB. Untuk keperluan perhitungan manual, diambil sampel sebanyak 7 karakter dari nilai heksadesimal file `dwsample.wav`. Nilai sampel heksadesimal yang diambil adalah: 69 73 6F 6D 70 34 32.

Tabel 1. Nilai Hexadesimal

Nilai	Frekuensi
69	1
73	1
6F	3

6D	3
70	1
34	1
32	1

Pembacaan frekuensi dilakukan dengan menghitung jumlah kemunculan setiap nilai. Kemudian, karakter-karakter diurutkan berdasarkan frekuensi kemunculannya, mulai dari karakter dengan frekuensi tertinggi hingga yang terendah.

Tabel 2. Hexa Yang Belum dikompresi

Hexa	Binner	Frekuensi	Bit	Frekuensi x Bit
69	01101001	1	8	8
73	01110011	1	8	8
6F	01101111	3	8	24
6D	01101101	3	8	24
70	01110000	1	8	8
34	00110100	1	8	8
32	00110010	1	8	8
TOTAL BIT				88

Berdasarkan tabel di atas, nilai (karakter) heksadesimal setara dengan 8 bit dalam biner. Dengan demikian, 16 digit heksadesimal memiliki nilai biner sebesar 128 bit. Untuk mengonversi jumlah bit menjadi byte, bagi total bit dengan 8. Jadi, $128/8 = 16$ byte akan dihasilkan.

Tabel 3. Codeword Goldbach

N	2(n+3)	Primes	Codeword
1	8	3+5	11
2	10	3+7	101
3	12	5+7	011
4	14	3+11	1001
5	16	5+11	0101
6	18	7+11	0011
7	20	7+13	00101
8	22	5+17	010001
9	24	11+13	00011
10	26	7+19	0010001
11	28	11+17	000101
12	30	13+17	000011
13	32	13+19	0000101
14	34	11+23	00110001

Karakter yang diubah menjadi codeword adalah karakter yang telah diurutkan sebelumnya berdasarkan frekuensinya.

Tabel 4. Kompresi Nilai WAV Goldbach

Karakter	Frek	2[n+3]	Prima	Codeword	Bit	Frekx Bit
69	1	8	3+5	11	2	2
73	1	10	3+7	101	3	3
6F	3	12	5+7	011	3	3
6D	3	14	3+11	1001	4	12
70	1	16	5+11	0101	4	4
34	1	18	7+11	0011	4	4
32	1	20	7+13	00101	5	5

Total Bit	39
-----------	----

Metode LZW akan memulai dengan inisialisasi berdasarkan kode ASCII, yaitu dari 0 hingga 255, dan indeks dalam kamus akan dimulai dari 256 dan seterusnya. Berikut adalah tahapan penerapan kompresi menggunakan metode LZW.

Tabel 5. Inisialisasi

Sampel	Decimal	Sampel	Decimal
69	105	00	00
73	115	06	6
6F	111	57	87
6D	109	0A	12
70	112	6D	109
34	52	6F	111
32	50	76	118

Selanjutnya dilakukan proses sesuai dengan tahapan yang ada pada algoritma LZW. Sehingga menghasilkan kompresi sebagai berikut ini.

Tabel 6. Kompresi

INPUT= P	NxCH=C	OUTPUT	ADDTODICTIONARY	
		INDEX	INDEX	STRING
00	00	0	256	0000
00	00			
00	18	0	257	0018
18	66	24	258	1866
66	74	102	259	6674
74	79	116	260	7479
79	70	121	261	7970
70	64	112	262	7064
64	61	100	263	6461
61	73	97	264	6173
73	68	115	265	7368
68	00	2	266	6800
00	00			
00	00			
0000	00	256	267	000000
00	69	0	268	0069
69	73	105	269	6973
73	6F	115	270	736F
6F	36	111	271	6F36
36	6D	54	272	366D
6D	70	109	273	6D70
70	34	112	274	7034
34	31	52	275	3400
31	00	49	276	3100
00	00			
0000	02	0	277	000002
02	60	2	278	0260
60	6D	96	279	606D

6D	6F	109	280	6D6F
6F	6F	111	281	6F6F
6F	76	111	282	6F76
76	-	118	283	76

Berdasarkan analisis perbandingan kinerja algoritma Goldbach dan algoritma Lempel-Ziv-Welch, dapat disimpulkan bahwa algoritma Goldbach terbukti lebih efektif dan optimal dalam mengompresi file gambar dibandingkan dengan algoritma Lempel-Ziv-Welch (LZW).

KESIMPULAN

Berdasarkan analisis yang dilakukan, penulis menarik kesimpulan sebagai berikut:

1. Kompresi file audio WAV telah berhasil dilakukan menggunakan metode kompresi perbandingan antara algoritma Goldbach Codes dan LZW. Hal ini menunjukkan bahwa aplikasi berfungsi sesuai dengan teknik kompresi yang digunakan.
2. Penelitian ini menunjukkan bahwa dengan menggunakan algoritma Goldbach Codes dan LZW, file audio WAV yang awalnya berukuran besar dapat dikompresi menjadi ukuran yang lebih kecil, sehingga mengurangi kebutuhan ruang penyimpanan.
3. Pengujian menunjukkan bahwa algoritma Goldbach Codes mencapai efisiensi kompresi sebesar 55%.

UCAPAN TERIMKASIH

Penulis mengucapkan terimakasih ke pembimbing yang telah banyak memberikan masukan dalam penyelesaian penelitian ini. Ucapan terima kasih juga penulis ucapkan ke program studi yang telah memberikan kesempatan untuk melakukan penelitian ini.

DAFTAR PUSTAKA

- [1] Muhammad Abdul Rivai Hi Wahab, 2017, "Implementasi Mekanisme Keamanan Data Dalam Bentuk Steganografi Dengan Metode Least Significant Bit (Lsb) Pada File Audio Wav", Juristek, Vol. 5, No. 2, Januari 2017, ISSN 2301-704X, Hal. 202-211, Magister Ilmu Komputer, Sistem Informasi, Universitas Budi Luhur Jakarta
- [2] Nadia.FR dkk, 2020. "Penerapan Algoritma Elias Delta Codes Dalam Kompresi File Teks". Building of Informatics, Technology and Science (BITS) Volume 2, No 2, December 2020 Page 109-114, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia.
- [3] Jamiatul Sisca, 2021. "Penerapan Algoritma Elias Delta Code Untuk Kompresi File Video Pada Aplikasi Video Downloader". Rekayasa Teknik Informatika dan Informasi ISSN 2745-7966 (Media Online) Vol 1, No 4, Maret 2021 Hal 254-

- 264, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia.
- [4] Ledi Varia Simanjuntak, 2020. "Perbandingan Algoritma Elias Delta Code dengan Levenstein Untuk Kompresi File Teks". *Journal of Computer System and Informatics (JoSYC)* Volume 1, No. 3, May 2020, Page 184-190, Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia.
- [5] Simatupang, Julianto, 2019. "Perancangan Sistem Informasi Pemesan Tiket Bus Pada PO. Handoyo Berbasis Online". Volume 3, No. 2 Oktober 2019 ISSN. 2549-0222 *Jurnal Intra-Tech*.
- [6] Supiyandi dan Okta Frida, "kompresi data huffman audio," vol. 2, no. 1, pp. 71-80, 2020.
- [7] UlfaLu'luilmaknun, Nilza Humaira Salsabila, "Penggunaan Metode Run Length Encoding Untuk Kompresi Data," *J. Tek. Inform. Unika St. Thomas*, vol. 03, no. 1, pp. 45-51, 2020, [Online]. Available: <http://ejournal.ust.ac.id/index.php/JTIUST/article/view/245/263>
- [8] F. Masruri and S. Teknik, "Kompresi Citra Digital Menggunakan Kode Huffman," 2020.
- [9] M. Yogie, "Penerapan Algoritma Goldbach Codes pada Kompresi File Gambar Terenkripsi Vigenere Cihper," *Pelita Inform. Budi Darma*, vol. 17, no. April, pp. 316–320, 2020.
- [10] H. Yuni, A. Sinaga, and L. Sitorus, "Pengamanan File Citra Digital Dengan Menggunakan Metode Least Significant Bit Dan End Of File," vol. 02, pp. 33-41, 2017.
- [11] W. F. Mahmudy and J. Matematika, "Steganografi Pada File Citra Bitmap 24 Bit Untuk Pengamanan Data Menggunakan Metode Least Significant Bit (LSB) Insertion," no. 2, pp. 38-44.
- [12] I. F. Interpretasi, "Format Citra dan Struktur Data untuk Citra," 2021.