

KOMPRESI FILE MENGGUNAKAN ALGORITMA LEMPEL ZIV WELCH (LZW)

Aries Suharso ¹⁾, Jejen Zaelani ²⁾, Didi Juardi ³⁾

¹⁾ Teknik Informatika Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang
Jl. HS. Ronggowaluyo, Telukjambe Timur, Karawang 41361

Abstrak

Di bidang teknologi informasi, komunikasi data terkait erat dengan pengiriman file. Ukuran file terkadang menjadi kendala dalam proses pengiriman. File besar akan membutuhkan waktu pengiriman lebih lama dibandingkan dengan file dengan ukuran lebih kecil. Karena itu, untuk menangani masalah tersebut, salah satunya dengan cara kompresi. Penelitian ini menggunakan metode eksperimen dengan model pengembangan air terjun dengan analisis, desain, pengkodean, dan pengujian. Aplikasi ini menerapkan algoritma Ziv Welch Lempel (LZW). Ziv Welch Lempel Algorithm (LZW) termasuk dalam teknik kompresi lossless, yang merupakan teknik kompresi yang tidak mengubah data asli. Hasil penilaian kompresi menggunakan algoritma Lempel Ziv Welch (LZW) menunjukkan tingkat rata-rata rasio kompresi dan untuk semua jenis file teks sebesar 51,04% dengan rata-rata 2,56 detik, untuk jenis file gambar sebesar 37,26% dengan rata-rata waktu 0,44 detik. Berdasarkan persentase rata-rata rasio kompresi untuk semua jenis file yang diuji menggunakan algoritma LZW (Lempel Ziv Welch) adalah 40,40% dengan waktu rata-rata yang diperlukan adalah 1,81 detik.

Kata kunci: Lempel Ziv Welch (LZW), Rasio Kompresi, Jenis File, Rugi, Diperlukan Waktu.

Abstract

In the field of information technology, data communication is closely related to file delivery. The size of the file is sometimes a constraint in the delivery process. Large files will take longer delivery times compared to files with smaller sizes. Therefore, to handle the problem, one of them by means of compression. This study uses an experimental method with a waterfall development model with analysis, design, coding, and testing. This application applies the Ziv Welch Lempel (LZW) algorithm. The Ziv Welch Lempel Algorithm (LZW) is included in the lossless compression technique, which is a compression technique that does not change the original data. The result of a compression assessment used the Lempel Ziv Welch (LZW) algorithm shows the average rate of compression ratio and for all types of text files by 51.04% with an average of 2.56 seconds, for an image file type of 37.26% with an average time of 0.44 seconds. Based on the average percentage of the compression ratio for all file types tested using the LZW algorithm (Lempel Ziv Welch) is 40.40% with an average time required is 1.81 seconds.

Keywords : Lempel Ziv Welch (LZW), Compression Ratio, Files Type, Lossless, Time Required.

1. Pendahuluan

Perkembangan teknologi informasi yang pesat sangat mempengaruhi dalam penyajian sebuah data atau informasi. Pada saat ini penyajian sebuah data atau informasi tidak hanya dalam bentuk teks, tetapi dapat berupa gambar, audio dan video. Keempat elemen data tersebut merupakan bagian dari objek digital. Pada umumnya representasi objek dalam bentuk digital membutuhkan memori. Semakin besar ukuran objek, maka semakin besar pula memori yang dibutuhkan.

Di samping itu komunikasi data sangat sering dilakukan. Komunikasi data ini berhubungan erat dengan pengiriman file menggunakan sistem transmisi elektronik dari satu terminal komputer ke terminal komputer yang lain. Besarnya ukuran file terkadang menjadi kendala dalam proses pengirimannya. File dengan ukuran besar akan memakan waktu pengiriman yang lebih lama dibandingkan dengan file yang memiliki ukuran lebih kecil, terkadang ada resiko tidak dapat tertampung file pada media penyimpanan, sehingga akan memperkecil kapasitas kosong dalam memori media penyimpanan. Oleh karena itu, manusia selalu berusaha untuk menemukan suatu cara alternatif untuk menangani permasalahan tersebut, salah satunya dengan cara kompresi [25]. Secara spesifik, kompresi file bertujuan untuk mengubah data menjadi sekumpulan kode untuk menghemat tempat penyimpanan dengan atau tanpa mengurangi kualitasnya serta mempercepat waktu transmisi data. Dengan memanfaatkan teknik kompresi ini, maka proses pengiriman file akan menjadi lebih maksimal dan lebih cepat.

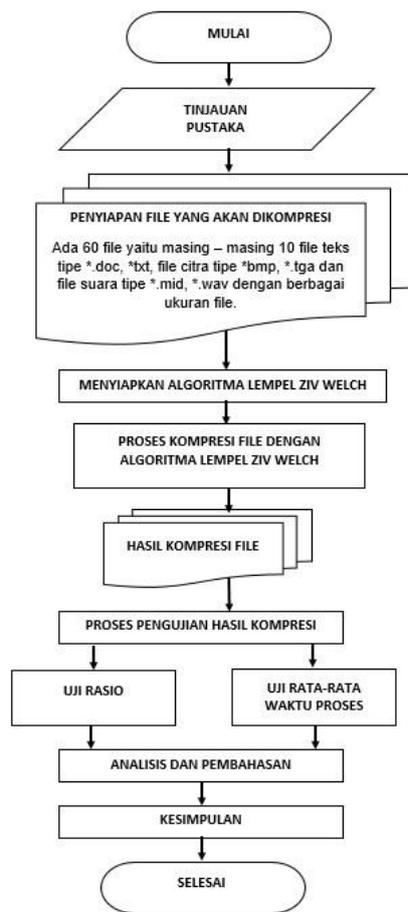
Sejak era 1940-an, mulai dipikirkannya suatu metode untuk memperkecil ukuran data atau kompresi. Hal ini menunjukkan bahwa di masa lalu, para pakar juga telah memprediksi pertumbuhan data yang akan semakin besar. Terdapat banyak metode kompresi, tetapi secara umum mempunyai teknik yang terbagi menjadi dua kategori yaitu *lossy compression* dan *lossless compression*. Masing-masing teknik tersebut mempunyai kekhasan sendiri terhadap data yang menjadi obyek kompresinya. Pada teknik *lossy compression*, lebih tepat digunakan pada obyek data audio, video dan citra digital, bahkan telah digunakan di dunia kedokteran sedangkan yakni untuk kompresi data ECG [2]. Teknik *lossless compression* lebih tepat digunakan pada data dokumen yang banyak mengandung teks daripada data multimedia karena mampu mempertahankan kebutuhan informasi yang dikandung oleh data [4]. Ada banyak teori dan metode untuk kompresi data, diantaranya *metode Huffman*, *Run-Length Encoding (RLE)*, *Lempel Ziv Welch (LZW)*, *Shanon-Fano*, dan beberapa metode lainnya.

Pada penelitian sebelumnya [12] telah didapatkan hasil perbandingan kinerja algoritma kompresi *huffman* [9], LZW dan *dmc* pada berbagai tipe file dengan rata-rata algoritma LZW membutuhkan waktu kompresi yang tersingkat (kecepatan kompresinya = $1139 \text{ KByte/sec} \pm 192,5$). *Algoritma Lempel Ziv Welch (LZW)* termasuk pada teknik *lossless compression*, yaitu teknik kompresi yang tidak mengubah data aslinya. Hal tersebut yang menyebabkan algoritma ini banyak dipakai dalam proses kompresi.

Penelitian ini merupakan penerapan lain dari *Algoritma Lempel Ziv Welch (LZW)* dalam kompresi ukuran file pada beberapa tipe file teks, file citra dan file suara yang mempunyai besar kapasitas berbeda. Kinerja kompresi algoritma *Lempel Ziv Welch (LZW)* dievaluasi dengan cara memperhatikan aspek rasio, *lossless* dan *Time Required* antara file asli sebelum kompresi dengan file setelah kompresi.

2. Metode Penelitian

Metode Penelitian dilakukan menggunakan Metode Eksperimen, berikut tahapan penelitian dijelaskan dalam bentuk digram alir seperti pada Gambar 1.



Gambar 1 Diagram alir tahapan penelitian

Objek penelitian yaitu file teks, file citra dan file suara yang mempunyai besar kapasitas berbeda. Pengujian difokuskan pada pengujian keakuratan kompresi file teks murni, file citra dan file suara. Objek pengujian sebagai Bahan Uji Coba dalam kompresi file ini yaitu 60 file yaitu masing – masing 10 file teks yang mempunyai tipe *.doc, *.txt, file citra yang mempunyai tipe *.bmp, *.tga dan file suara yang mempunyai tipe *.mid, *.wav pengujian dilakukan terhadap berbagai ukuran file.

Data input yang digunakan dalam uji coba ditunjukkan pada Tabel 1

Tabel 1 Daftar File Awal Yang Akan Dikompresi

No	Nama File	Tipe File	Ukuran File (Byte)
1	cp936	*.txt	839.399
2	demo	*.txt	384.407
3	license	*.txt	434.229
4	Line break	*.txt	880.517
5	log trace	*.txt	547.812
6	map	*.txt	142.601
7	news	*.txt	415.575
8	status	*.txt	1.277.661

9	strings	<i>*.txt</i>	359.312
10	table text	<i>*.txt</i>	1.810.352
11	modul	<i>*.doc</i>	9.308.995
12	argumen	<i>*.doc</i>	2.343.316
13	antivirus	<i>*.doc</i>	62.976
14	graph	<i>*.doc</i>	156.672
15	prosem	<i>*.doc</i>	1.169.920
16	queue	<i>*.doc</i>	305.152
17	bilangan	<i>*.doc</i>	61.952
18	peserta didik	<i>*.doc</i>	65.024
19	experien	<i>*.doc</i>	23.040
20	file	<i>*.doc</i>	12.372.776
21	crest	<i>*.bmp</i>	4.319.278
22	edison	<i>*.bmp</i>	65.078
23	gordon	<i>*.bmp</i>	649.034
24	horses	<i>*.bmp</i>	287.518
25	jackass	<i>*.bmp</i>	67.702
26	santa	<i>*.bmp</i>	492.302
27	test	<i>*.bmp</i>	2.260.662
28	dad	<i>*.bmp</i>	271.102
29	wolf	<i>*.bmp</i>	811.062
30	yuerling	<i>*.bmp</i>	24.662
31	bird	<i>*.tga</i>	65.554
32	demo	<i>*.tga</i>	5.116.364
33	ffc	<i>*.tga</i>	750.044
34	flag_b24	<i>*.tga</i>	50.242
35	france	<i>*.tga</i>	333.330
36	frymire	<i>*.tga</i>	3.706.188
37	library	<i>*.tga</i>	163.346
38	flag_b16	<i>*.tga</i>	30.770
39	serrano	<i>*.tga</i>	1.498.296
40	xing_t32	<i>*.tga</i>	157.458
41	3 Diva	<i>*.mid</i>	107.300
42	pemain cinta	<i>*.mid</i>	62.968
43	sayang	<i>*.mid</i>	54.946
44	matahariku	<i>*.mid</i>	68.323
45	alamat palsu	<i>*.mid</i>	92.598
46	marilah kemari	<i>*.mid</i>	79.354
47	hitamku	<i>*.mid</i>	83.327
48	badai pasti berlalu	<i>*.mid</i>	73.928
49	madu & racun	<i>*.mid</i>	26.543
50	rio febrian	<i>*.mid</i>	128.294
51	4ch	<i>*.wav</i>	1.352.316
52	gtr	<i>*.wav</i>	220.972
53	look what	<i>*.wav</i>	524.288
54	mifi	<i>*.wav</i>	376.154
55	stereo	<i>*.wav</i>	234.278
56	goal	<i>*.wav</i>	41.102
57	finish	<i>*.wav</i>	37.740

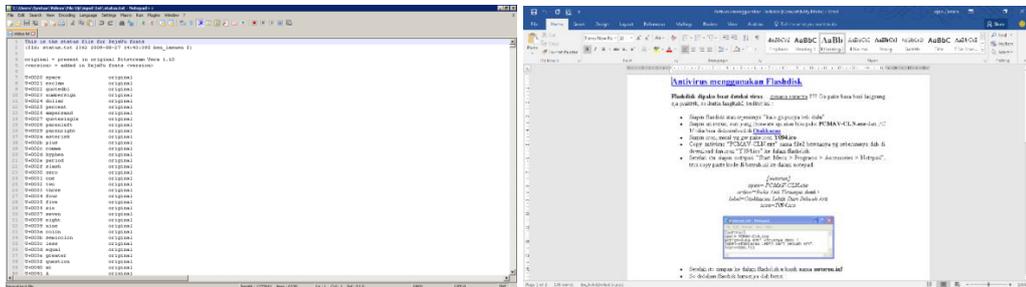
58	speech	*.wav	153.644
59	windows print	*.wav	353.836
60	tada	*.wav	285.228

Proses kompresi akan menjadikan semua tipe file menjadi file *.zen yang selanjutnya dapat dikembalikan melalui prose Dekompresi. Penentuan hasil kinerja Algoritma LZW dihitung dari seberapa besar perubahan kapasitas file yang dikompresi dan seberapa cepat waktu proses kompresi dilakukan. Adapun hasil pengukurannya disajikan dalam bentuk tabel dan grafik yang memperlihatkan hasil percobaan dengan berbagai proporsi antara data contoh dan data uji coba.

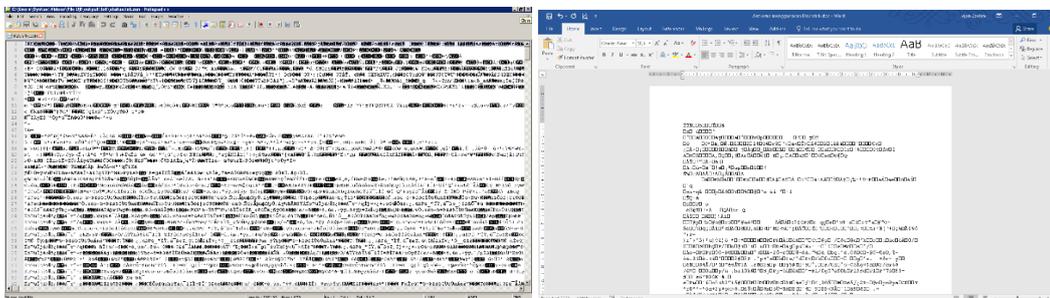
3. Hasil dan Pembahasan

Visualisasi hasil kompresi LZW

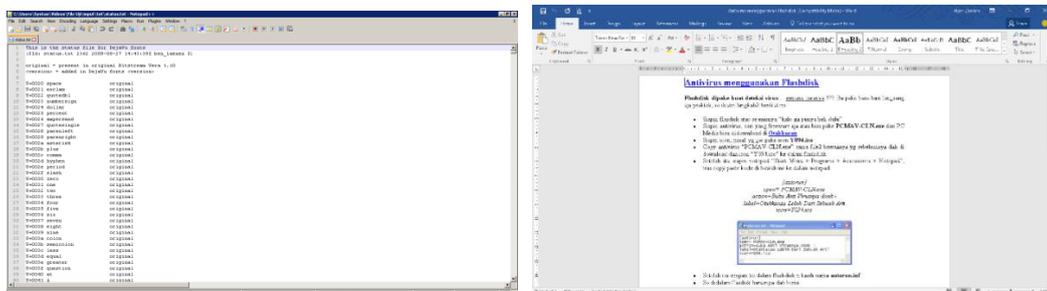
Hasil visual uji coba kompresi LZW untuk tampilan isi file jenis dokumen yaitu file *.txt dan file *.doc sebelum melakukan proses kompresi masih menampilkan teks yang dapat terbaca, berbeda dengan file *.txt dan file *.doc yang sudah melewati proses kompresi karena telah melalui pemampatan file.



Gambar 2 Tampilan isi dari file *.txt (kiri) dan file *.doc (kanan) sebelum proses kompresi LZW



Gambar 3 Tampilan isi dari file *.txt dan file *.doc setelah melalui proses kompresi LZW

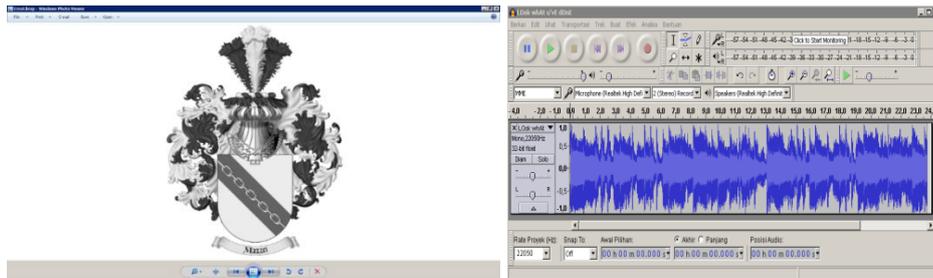


Gambar 4 Tampilan isi dari file *.txt dan file *.doc setelah dilakukan proses Dekompresi

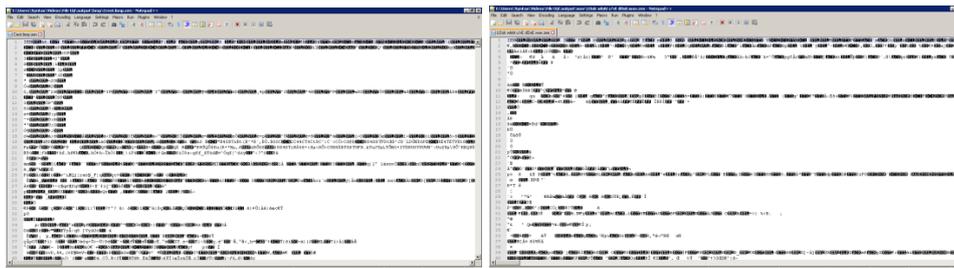


Setelah melalui proses kompresi maka tampilan isi dari file *.txt berubah dan tidak dapat terbaca karena semua karakter yang ada telah dirubah. Tetapi dengan melalui proses dekompresi maka isi dari file *.txt akan kembali seperti semula.

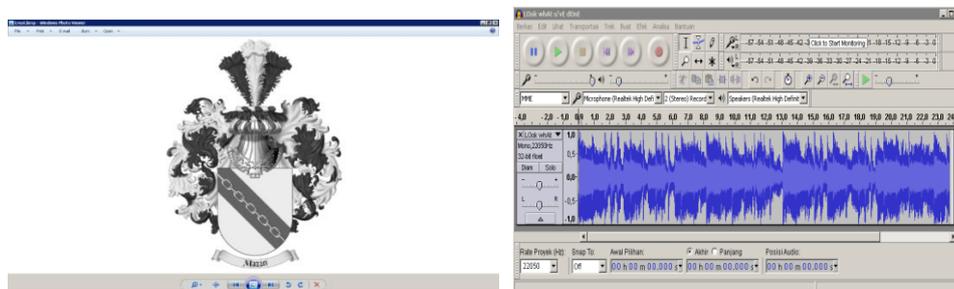
Hasil visual uji coba kompresi LZW untuk tampilan isi file jenis Gambar atau Citra yang diwakili oleh file *.bmp dan isi file jenis Audio atau suara yang diwakili oleh file *.wav sebelum melakukan proses kompresi masih dapat dibuka namun sudah mengalami perubahan yang tidak terbaca sebagaimana mestinya hal ini dikarenakan file sudah melewati proses kompresi sehingga terjadi pemampatan file.



Gambar 5 Tampilan isi dari file *.bmp (kiri) dan file *.wav (kanan) sebelum proses kompresi LZW



Gambar 6 Tampilan isi dari file *.bmp dan file *.wav setelah melalui proses kompresi LZW



Gambar 7 Tampilan isi dari file *.bmp dan file *.wav setelah dilakukan proses Dekompresi

Setelah melalui proses kompresi maka tampilan isi dari file *.bmp dan file *.wav berubah dan tidak dapat terbaca karena semua karakter yang ada telah dirubah. Tetapi dengan melalui proses dekompresi maka isi dari file *.bmp dan file *.wav akan kembali seperti semula.

Evaluasi hasil rata-rata Rasio dan waktu Kompresi LZW

Hasil rata – rata tingkat rasio kompresi LZW dan rata – rata waktu proses kompresi LZW untuk semua tipe file teks, file citra, dan file suara ditunjukkan pada tabel 3.1 Untuk hasil terbaik rasio rata-rata yang dicapai oleh jenis file teks dengan hasilnya sebesar 51,04%. Namun untuk rata – rata waktu terbaik dicapai oleh jenis file suara dengan waktu 0,44 detik hal ini disebabkan karena ukuran untuk file suara relatif kecil sehingga membuat proses kompresi menjadi lebih cepat.

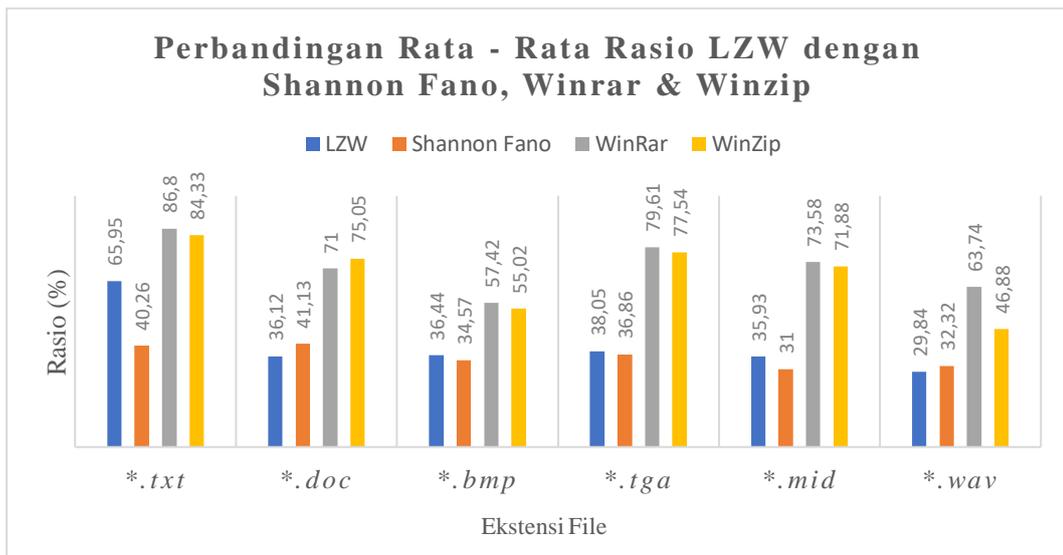


Tabel 2 Hasil Rata-rata Rasio dan Waktu proses kompresi Algoritma Lempel Ziv Welch (LZW)

No.	Jenis File	Tipe File	Rasio (%)	Waktu (detik)	Rata – Rasio Berdasarkan Jenis File (%)	Rata – Waktu Berdasarkan Jenis File (detik)
1	Teks	*.txt	65,95	0,78	51,04	2,56
		*.doc	36,12	4,34		
2	Citra	*.bmp	36,44	2,00	37,26	2,44
		*.tga	38,08	2,88		
3	Suara	*.mid	35,93	0,19	32,89	0,44
		*.wav	29,84	0,68		
Rata – Rata Keseluruhan					40,40	1,81

Evaluasi Kinerja Algoritma LZW dibandingkan dengan Shannonn Fano, WinZip dan WinRAR

Hasil uji coba aplikasi kompresi dengan algoritma LZW dibandingkan dengan perangkat lunak aplikasi kompresi yang menggunakan algoritma *shannon fano* sudah cukup baik dalam memampatkan data. Namun jika dibandingkan dengan aplikasi kompresi lainnya seperti WinZip dan WinRAR, masih kurang baik dalam memampatkan data. Hal ini disebabkan pada perangkat lunak yang sudah populer digunakan pada saat ini menerapkan lebih dari satu algoritma, sehingga memungkinkan hasil kompresi yang didapatkan sangat maksimal. Perbandingan rasio hasil kompresi algoritma LZW dengan perangkat lunak WinZip dan WinRAR.



Gambar 8 Perbandingan Rata-Rata Rasio Hasil Kompresi Algoritma LZW dengan Shannon Fano, WinZip dan WinRAR

Diskusi

Berdasarkan hasil penelitian ini, kami sarankan sebagai referensi untuk penelitian selanjutnya sebagai berikut:

- Perbaiki dalam proses pembacaan karakter yang dilakukan dalam algoritma (Lempel Ziv Welch), sehingga dengan demikian proses kompresi dapat memampatkan semua file dengan lebih baik.
- Pada uji kinerja Algoritma kompresi file ini hanya menggunakan satu inputan saja. Akan lebih baik jika aplikasi mampu menerima banyak inputan (multi) dan membentuk satu data baru hasil kompresi.

- Meningkatkan rasio kompresi dengan cara menggabungkan algoritma kompresi LZW (Lempel Ziv Welch) pada algoritma rasio kompresi yang lain seperti Huffman, Run-Length Encoding (RLE) atau Shanon-Fano.
- Pada pengembangan selanjutnya kompresi yang dibuat diharapkan lebih variatif dan inovatif dari aplikasi yang telah ada sebelumnya.

4. Kesimpulan

Algoritma *Lempel Ziv Welch* telah dapat digunakan untuk memperkecil ukuran file pada file teks, citra dan suara yang mempunyai besar kapasitas berbeda yang disimpan pada memori sehingga dalam pengiriman file tidak membutuhkan waktu yang lama. Hal ini terbukti dari hasil kompresi Algoritma *Lempel Ziv Welch* dapat melakukan proses kompresi untuk memperkecil ukuran file dengan hasil rata – rata rasio kompresi terhadap file teks sebesar 51,04% dan rata – rata waktunya 2,56 detik, file citra sebesar 37,26% dan rata – rata waktunya 2,44 detik dan file suara sebesar 32,89% dan rata – rata waktunya 0,44 detik. Keseluruhan rata – rata tingkat rasio kompresi untuk semua tipe file yang diuji menggunakan algoritma LZW (Lempel Ziv Welch) adalah 40,40% dan rata – rata waktu yang dibutuhkan selama 1,81 detik.

Evaluasi Kinerja Algoritma *Lempel Ziv Welch* dapat ditunjukkan berdasarkan uji coba yang telah dilakukan, dengan membandingkan rasio kompresi terhadap beberapa file teks, file citra dan file suara, algoritma LZW (Lempel Ziv Welch) lebih baik dari aplikasi kompresi yang menggunakan algoritma Shannon Fano dalam memampatkan file terutama pada file yang bertipe *.txt dengan hasil rasio yang didapat 65,95% untuk LZW dan 40,26% untuk Shannon Fano. Namun jika dibandingkan dengan perangkat lunak aplikasi kompresi lainnya seperti WinZip dan WinRar, masih kurang baik dalam memampatkan data. Hal ini disebabkan pada perangkat lunak yang sudah populer digunakan pada saat ini menerapkan lebih dari satu algoritma, sehingga memungkinkan hasil kompresi yang didapatkan sangat maksimal.

References

- [1] Akinwale OB & Kehinde LO. (2017). Data Compression for Remote Laboratories. iJIM – Vol. 11, No. 4
- [2] Al-alak, S.M.K., Alwan, I.H. and Hussein, A.A.A. 2017. Adapted LZW Protocol for ECG Data Compression. Journal Of University Of Babylon for Pure and Applied Sciences. 25, 5 (Nov. 2017), 1618 - 1626.
- [3] Barakbah, A. R, Karlita, T & Ahsan, A. S. (2013). Logika dan Algoritma. Teknik Informatika. Politeknik Elektronika Negeri Surabaya
- [4] Devi A.M.D & Kumar Shiva K.B. (2018). Secure reversible color image data hiding technique using image classifier and Lempel Ziv Welch image compression technique. International Journal of Engineering & Technology
- [5] Dzulhaq, I. M & Andayani, A. A. (2014). Aplikasi Kompresi File dengan Metode Lempel Ziv Welch. Jurnal SISFOTEK Global. Vol 1: hal 1-4.
- [6] Fengan Li, Lingjiao Chen, Arun Kumar, Jeffrey F. Naughton, Jignesh MP, Xi Wu. (2017). When Lempel Ziv Welch Meets Machine Learning: A Case Study of Accelerating Machine Learning using Coding.
- [7] Handayani, D. (2001). Sistem Berkas. Yogyakarta: J&J Learning.
- [8] Hari. (2012). Compression Algorithms: Huffman and Lempel Ziv Welch (LZW). Lecture Notes, Chapter 3. MIT 6.02
- [9] Huffman D (1991). Scientific American, pp. 54-58. <https://www.huffmancoding.com/my-uncle/scientific-american>.
- [10] Kadir, A. & Triwahyuni, T. Ch. (2005). Pengenalan Teknologi Informasi. Yogyakarta: ANDI.
- [11] KBBI. Kamus Besar Bahasa Indonesia (KBBI) : Kamus Versi Online/Daring (dalam jaringan). <https://www.kbbi.web.id/data> (diakses pada tanggal 28 Januari 2018 pukul 19.38 sampai 24 Agustus 2018 pukul 21.40 WIB)

- [12] Linawati & Panggabean, H. P. (2004). Perbandingan Kinerja Algoritma Kompresi Huffman, LZW dan DMC pada Berbagai Tipe File. *Integral*. Vol 9: hal 7-16.
- [13] Narang M. (1998). Lempel Ziv Welch data compression using associative processing as an enabling technology for real time application. RIT Scholar Works, <https://scholarworks.rit.edu/theses> (diakses pada tanggal 30 Januari 2018 pukul 17.28 sampai 30 Agustus 2018 pukul 21.40 WIB)
- [14] Nelson, M & Gailly, J.L. (1996). *The Data Compression*. New York: M&T book.
- [15] Prasetyo, G. B. (2013). *Kompresi File Audio Wave Menggunakan Algoritma Huffman Shift Coding*. Ilmu Komputer. Universitas Brawijaya.
- [16] Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta: ANDI
- [17] Rizky, S. (2011). *Konsep Dasar Perangkat Lunak*. Jakarta: Prestasi Pustaka.
- [18] Salomon, D & Motta, M. (2010). *Handbook of Data Compression (4th edition)*. London: Springer-Verlag.
- [19] Salomon, D. (2007). *Data Compression, The Complete Reference (4th edition)*. London: Springer-Verlag.
- [20] Tambunan, R. A. P. (2014). Perancangan Aplikasi Kompresi File Menggunakan Algoritma Half Byte. *Pelita Informatika Budi Darma*. Vol 8: hal 104-109.
- [21] Wahyudi, B. (2008). *Catatan Manajemen Basis Data*. Fakultas Ekonomi. Universitas Gunadarma.
- [22] Warsita, R, Setiawan, R. A & Yoannita. (2015). Rancang Bangun Aplikasi Kompresi Audio Berbasis Android Menggunakan Algoritma Huffman. *Teknik Informatika*. STMIK Global Informatika MDP.
- [23] Welch, Terry A. (1984). A Technique for High-performance Data Compression. *IEEE Computer* (june 1984), pp-8-19. https://www2.cs.duke.edu/courses/spring03/cps296.5/papers/welch_1984_technique_for.pdf (diakses pada tanggal 30 Januari 2018 pukul 17.28 sampai 30 Agustus 2018 pukul 21.40 WIB)
- [24] Wibowo, A. (2012). Kompresi Data menggunakan Metode Huffman. *Semantik*. Vol 2: hal 1-5.
- [25] Winanti, W. (2006). *Aplikasi Pohon Biner*. Teknik Informatika. Institut Teknologi Bandung.