

Implementasi Metode Lempel-Ziv-Welch pada Kompresi File Teks

Muhammad Charis Setiawan¹, Gusti Eka Yuliasuti², Andy Rachman³

Institut Teknologi Adhi Tama Surabaya

e-mail: arizsetyawan94@gmail.com

ABSTRACT

The increasing use of digital data in this era goes along with the increasing need to store data on storage media. As a result, data security becomes important when using the internet, especially websites, because they also have security threats such as attacks on URLs that can penetrate databases. In response to problems of increasing data and URL security, compression is necessary for data and encryption is beneficial for URL security. This study employed the half-byte compression method and cipher block chaining encryption. The data used for compression comprised txt, docx, and doc files, while the encrypted URL used the ID of the file. The results of the average value of compression testing obtained the highest compression ratios for.txt files (1.150), docx files (120.469), and doc files (47.622). Meanwhile, the results of encryption testing obtained an average encryption execution time of 0.0000466347 seconds. The attempt to penetrate the database through SQL injection failed. In conclusion, the best average occurred in a.docx file because docx/doc files only compress text without images, making the half-byte method applied by researchers only suitable for files containing text. Encryption was able to protect against SQL injection attacks, and the execution time was quite good, under 1 second.

Kata kunci: Compression, Data Compression, Decompression, File Text, , Lempel-Ziv-Welch

ABSTRAK

Perkembangan teknologi informasi dan komunikasi saat ini berkembang sangat pesat. Salah satu perkembangan di bidang teknologi saat ini adalah file teks dan media penyimpanan. Pada saat ini format file teks terus berkembang dalam berbagai macam. Tentunya masalah yang sering dihadapi saat ini yakni penggunaan media penyimpanan data. Dengan adanya permasalahan tersebut maka dibutuhkan suatu metode kompresi file teks, salah satunya menggunakan metode Lempel-Ziv-Welch untuk mengurangi ukuran data dengan melakukan kompresi pada suatu file teks. Data yang akan diuji kompresi pada penelitian ini sebanyak 15 data file teks berekstensi *.txt yang dibagi dalam 3 kategori, yakni 200 KB, 100 KB dan dibawah 100 KB. Pada penelitian ini hasil kompresi metode Lempel-Ziv-Welch mendapatkan hasil kompresi dengan rata – rata selisih ukuran sebesar 86.1048 bytes. Setelah diuji kompresi, tiap file akan diuji dengan pengukuran berdasarkan Ratio of Compress (RC), Compression Ratio (CR), Redudancy dan Space Saving. Berdasarkan Dari 4 pengujian yang dilakukan terhadap 15 data yang diuji, metode Lempel-Ziv-Welch memiliki rata-rata Ratio of Compress (RC) sebesar 4,37, Compression Ratio (CR) sebesar 22.99%, Redudancy sebesar 76.99% dan Space Saving sebesar 21.99%.

Kata kunci: Data Kompresi, Dekompresi, File Teks, Kompresi, Lempel-Ziv-Welch

PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi saat ini berkembang sangat pesat [2]. Penggunaan teknologi informasi dan komunikasi (TIK) saat ini cukup meluas dan mempengaruhi seluruh aspek didalam kehidupan masyarakat [3]. Salah satu perkembangan dibidang teknologi saat ini adalah file teks dan media penyimpanan, dimana media penyimpanan merupakan faktor yang paling utama, karena akan digunakan tempat dimana data akan disimpan [4]. Pada saat ini format file teks berkembang dalam berbagai macam, seperti .pdf .docx, .rtf, html dan lain sebagainya.

Tentunya masalah yang sering dihadapi saat ini yakni penggunaan media penyimpanan data, sehingga hal ini membutuhkan media penyimpanan yang cukup besar [5]. Dengan adanya permasalahan tersebut maka dibutuhkan suatu metode kompresi untuk mengurangi ukuran data dengan melakukan kompresi pada suatu file teks. Kompresi merupakan suatu proses untuk mengubah sebuah ukuran file yang besar menjadi file yang berukuran lebih kecil, dengan menghilangkan beberapa informasi sehingga file tersebut dapat disimpan secara efisien [6]. Secara umum tujuan dari kompresi adalah untuk memperkecil dan mengurangi redundancy (kelebihan data) pada suatu data, sehingga menghemat penggunaan alokasi penyimpanan pada suatu memori [7].

Pada penelitian ini metode yang akan diterapkan yaitu menggunakan metode Lempel-Ziv-Welch. Algoritma Lempel-Ziv-Welch merupakan algoritma kompresi Universal Lossless berbasis dictionary yang diciptakan oleh Abraham Lempel, Jacob Ziv, dan Terry Welch [8]. Metode Lempel-Ziv-Welch merupakan metode yang bersifat Lossless Compression. Lossless Compression merupakan suatu metode kompresi data dengan tidak menghilangkan atau mengurangi jumlah dari sebagian “informasi” selama proses kompresinya. Sehingga jumlah total bit atau informasi yang dikembalikan dari proses dekompresi akan sama dengan data aslinya [9]. Prinsip kerja algoritma Lempel-Ziv-Welch adalah memeriksa setiap karakter yang muncul, kemudian menggabungkan dengan karakter berikutnya menjadi suatu string baru yang diindekskan kedalam dictionary [10].

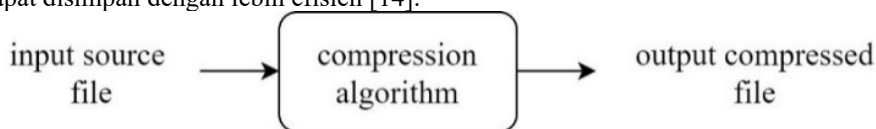
TINJAUAN PUSTAKA

File Teks

Teks merupakan kumpulan dari suatu karakter atau string yang menjadi satu kesatuan [11]. File teks merupakan file yang berisi informasi dalam bentuk teks. Data input yang ada didalam file teks berasal dari dokumen pengolah kata yang terdiri dari huruf, angka, dan tanda baca. Seperti angka yang digunakan dalam perhitungan, nama dan alamat dalam database merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca [12].

Kompresi Data

Pada awalnya, teori informasi memunculkan bidang ilmu komputer yang dikenal sebagai kompresi data [13]. Proses kompresi merupakan suatu proses untuk membuat representasi digital dari ukuran suatu data menjadi lebih kecil dan ringkas, tanpa mengurangi makna dan nilai dari informasi yang terkandung pada data tersebut. Sistem kerja dari kompresi data adalah dengan melakukan proses mengkonversi input data stream (sumber stream, atau data mentah asli) kedalam bentuk stream lain (output stream, atau stream yang sudah terkompresi) yang memiliki ukuran relatif lebih kecil. Menurut pendapat lain, kompresi data juga diartikan sebagai teknik kompresi yang digunakan pada data untuk menghasilkan ukuran data yang lebih kecil dari ukuran aslinya agar dapat disimpan dengan lebih efisien [14].



Gambar 1 Diagram Konteks Kompresi Secara Umum
(Sumber : [1])

Pada gambar 1 menunjukkan tahap proses kompresi diawali dengan inputan berupa file asli kemudian file dikompresi menggunakan metode atau algoritma kompresi tertentu yang akan menghasilkan file terkompresi.

Pada perkembangannya kompresi data dibagi menjadi dua bagian, yaitu:

a. **Lossless Compression**

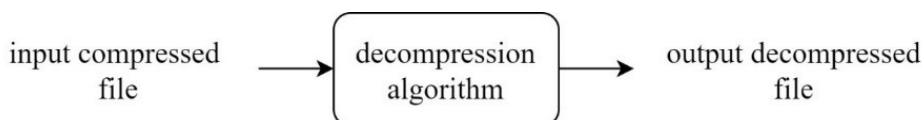
Metode Lossless Compression merupakan teknik kompresi dimana data yang dikompresi dapat disusun kembali dari data aslinya [15].

b. **Lossy Compression**

Lossy Compression adalah teknik kompresi data dimana beberapa informasi dari data asli dihapus selama proses kompresinya. Tetapi, kompresi lossy tidak sepenuhnya menghapus semua informasi yang ada pada suatu data [9].

Dekompresi

Dekompresi merupakan kebalikan dari proses kompresi. Dekompresi adalah proses pengembalian data dari proses kompresi, ke data awal atau kedalam bentuk file semula [16]. Setelah dilakukan proses kompresi terhadap sebuah gambar atau file, maka data atau file akan dapat dikembalikan ke dalam bentuk semula yaitu dengan melakukan dekompresi [17].



Gambar 2 Diagram Konteks Dekompresi Secara Umum
 (Sumber : [1])

Pada gambar 2 proses dekompresi diawali dengan menginputkan file yang sudah terkompresi kemudian didekompresi menggunakan algoritma yang sama dengan proses kompresinya untuk mengembalikan file aslinya.

Lempel-Ziv-Welch

Algoritma Lempel-Ziv-Welch merupakan algoritma kompresi data lossless yang ditemukan oleh Terry Welch, dimana algoritma Lempel-Ziv-Welch merupakan peningkatan versi dari algoritma Lempel Ziv 77 (LZ77) dan Lempel Ziv 78 (LZ78) yang dikembangkan oleh Abraham Lempel dan Jacob Ziv pada tahun 1977 dan 1978 [12]. Algoritma Lempel-Ziv-Welch merupakan algoritma yang bersifat adaptif dan berbasis dictionary, sehingga tidak ada kemungkinan kehilangan data saat proses kompresinya [8]. Algoritma Lempel-Ziv-Welch ini umumnya digunakan untuk kompresi file teks atau semacamnya [18]. Prinsip umum kerja algoritma Lempel-Ziv-Welch adalah mengecek setiap karakter yang muncul kemudian menggabungkan dengan karakter selanjutnya menjadi sebuah string baru. Kemudian string baru tersebut akan diindekkan kedalam suatu dictionary [10].

Rasio Kompresi

Rasio Kompresi merupakan hasil perbandingan antara ukuran data setelah proses kompresi dengan ukuran data asli [19]. Dalam menentukan hasil rasio kompresi dapat diukur dengan parameter perhitungan rumus sebagai berikut :

$$\begin{aligned}
 \text{Ratio of Compression (RC)} &= \frac{\text{Ukuran data sebelum dikompresi}}{\text{Ukuran data sesudah dikompresi}} \\
 \text{Compression Ratio (CR)} &= \frac{\text{Ukuran data sesudah dikompresi}}{\text{Ukuran data sebelum dikompresi}} \times 100 \\
 \text{Redundancy (RD)} &= \frac{\text{Sebelum dikompresi} - \text{Sesudah dikompresi}}{\text{Ukuran data sebelum dikompresi}} \times 100
 \end{aligned}$$

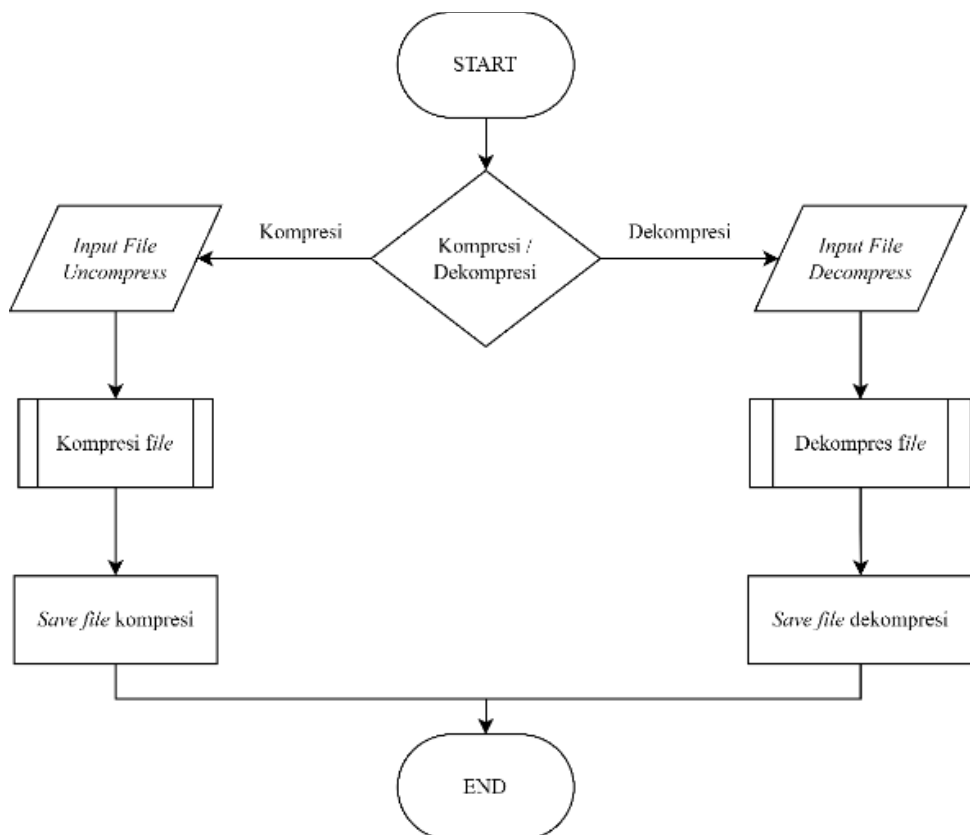
$$\text{Space Saving (SS)} = (1 - \text{Compression ratio (CR)}) \times 100\%$$

Semakin tinggi rasio yang dihasilkan pada suatu kompresi data maka semakin efektif teknik kompresi tersebut

METODE

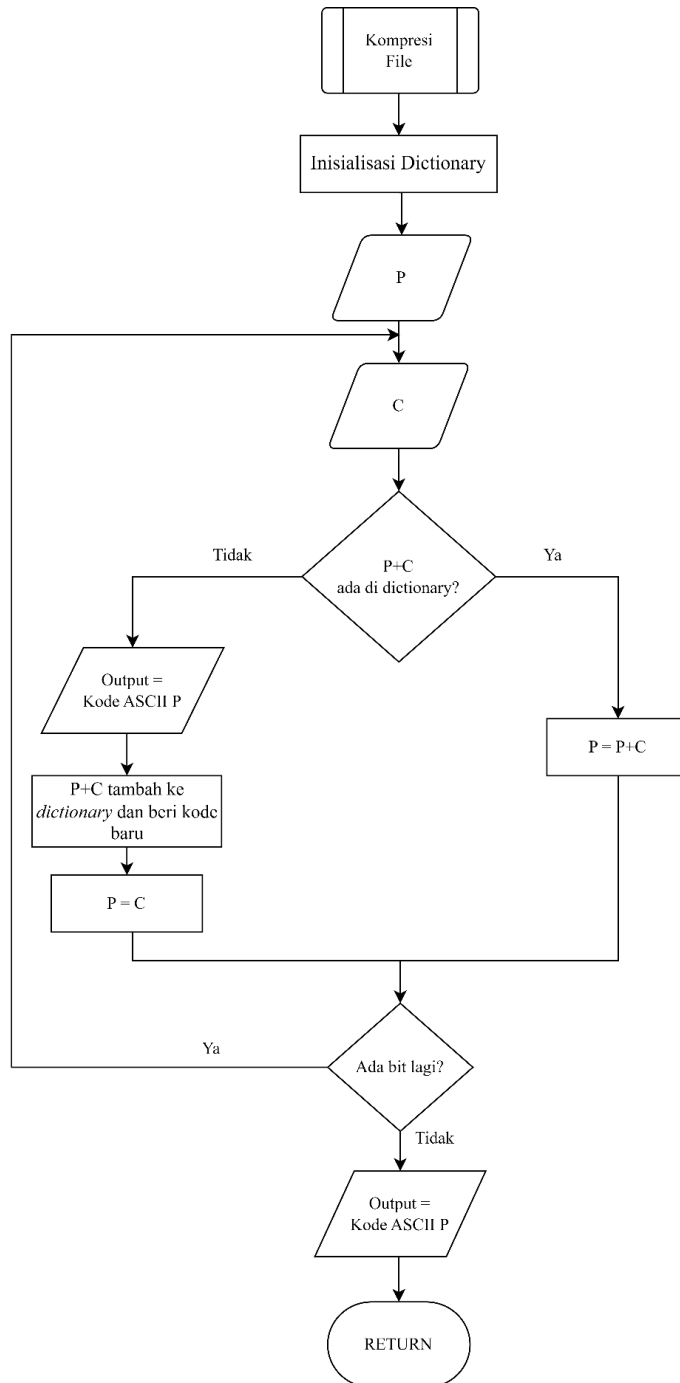
Pada rancangan sistem yang nantinya akan dibuat terdapat 2 menu utama, yakni menu kompresi dan dekompresi. Kompresi algoritma Lempel-Ziv-Welch menggunakan teknik dictionary dalam kompresinya. Dimana string karakter digantikan oleh kode tabel yang dibuat setiap ada string yang masuk. Tabel dibuat untuk referensi masukan string selanjutnya. Ukuran tabel dictionary pada algoritma Lempel-Ziv-Welch asli adalah 4096 sampel, dimana 256 sampel pertama digunakan untuk tabel karakter single (Extended ASCII), dan sisanya digunakan untuk pasangan karakter atau string dalam data input.

Adapun alur atau rancangan dari program kompresi file teks dengan metode Lempel-Ziv-Welch dapat dilihat pada gambar



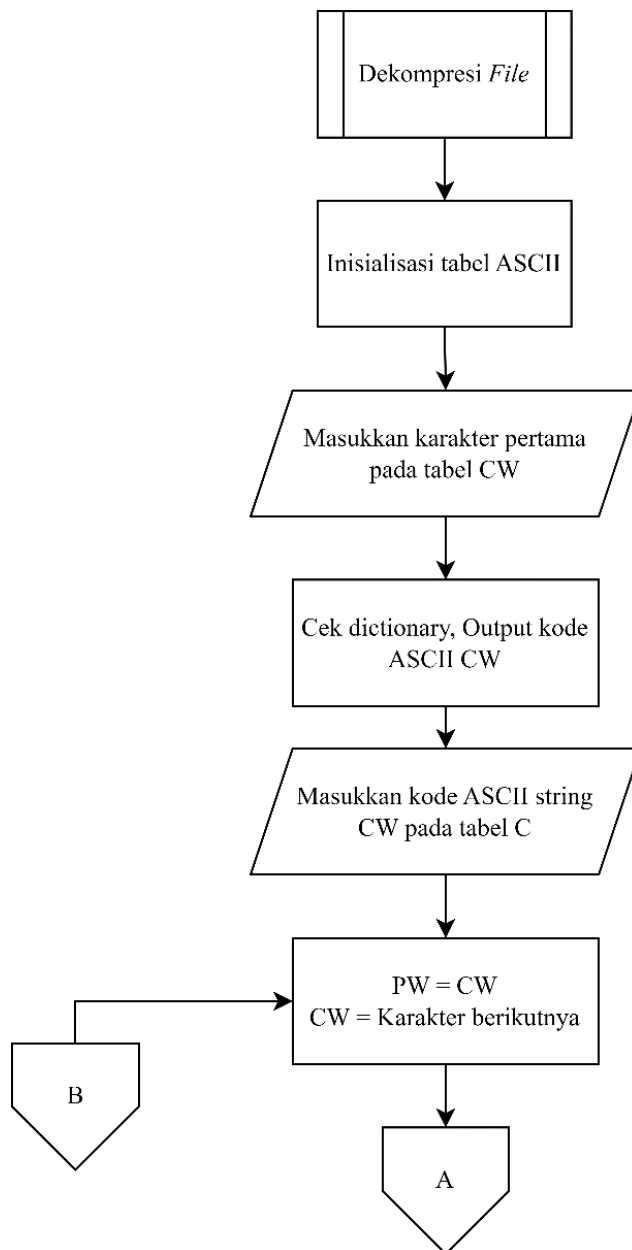
Gambar 3 Rancangan Program Kompresi dan Dekompresi

Alur Kompresi Lempel-Ziv-Welch

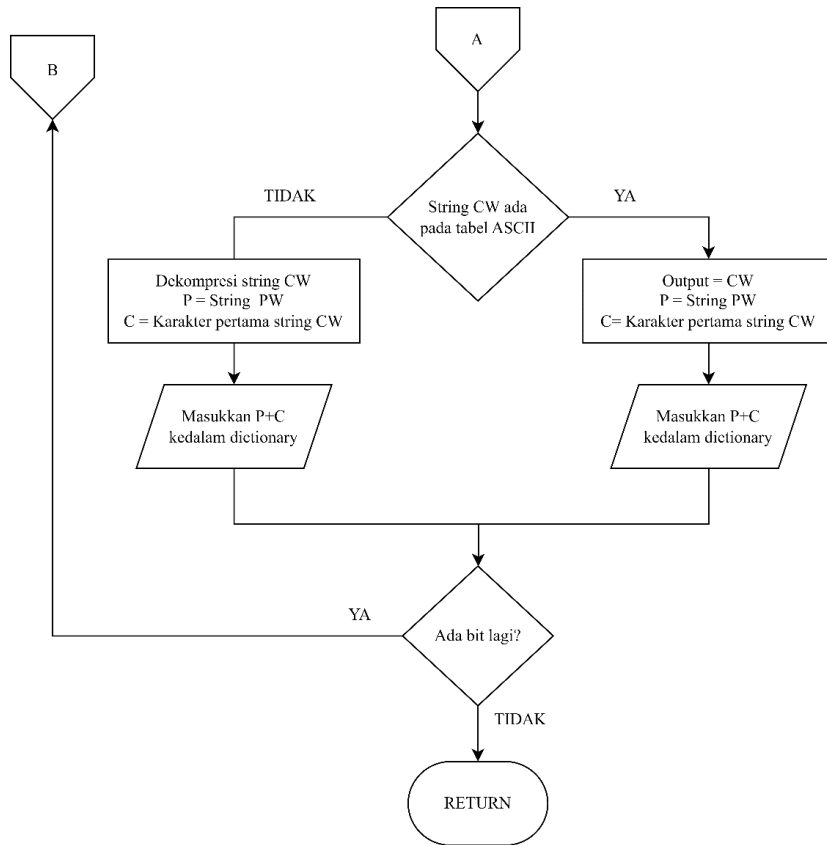


Gambar 4 Flowchart Algoritma Lempel-Ziv-Welch

Alur Dekompresi Lempel-Ziv-Welch



Gambar 5 Flowchart Dekompresi Lempel-Ziv-Welch

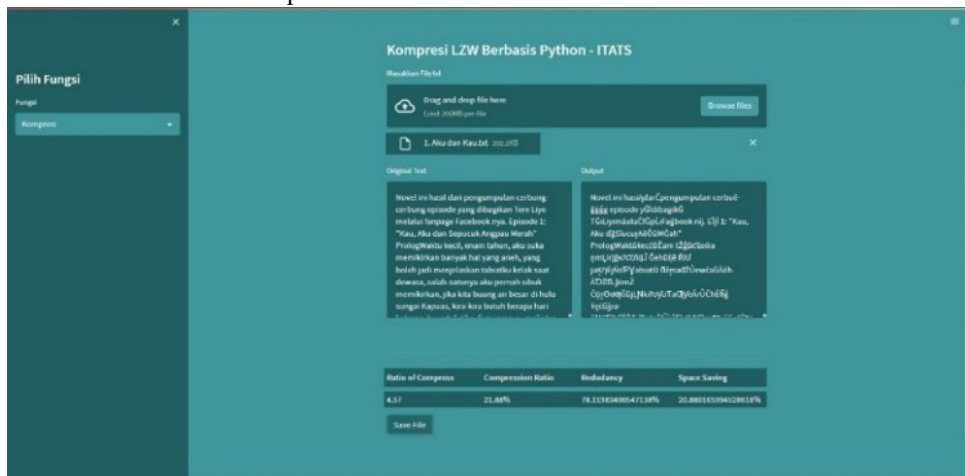


Gambar 6 Lanjutan Flowchart Dekompresi Lempel-Ziv-Welch

HASIL DAN PEMBAHASAN

Tampilan Halaman Sistem

a. Halaman Menu Kompresi File



Gambar 7 Halaman Menu Kompresi File

Pada gambar 3 merupakan tampilan dari sistem kompresi file teks yang terdiri dari menu kompresi. Dimana user dapat mengkompresi file teks yang akan dikompresi pada halaman ini.

b. Halaman Menu Dekompresi File



Gambar 8 Halaman Menu Kompresi File

Pada gambar 4 merupakan tampilan dari sistem dekomposisi file teks yang terdiri dari menu dekomposisi. Dimana user dapat mendekompresi file teks yang akan telah dikompresi sebelumnya pada halaman ini.

Pengujian Kompresi

Data yang digunakan pada penelitian ini adalah bertipe data file teks berformat *.txt, yang berisikan cerita novel dan E-book yang diperoleh dari beberapa sumber diinternet. Data yang akan diuji sebanyak 15 yang dibagi menjadi 3 kategori ukuran, yakni ukuran 200 KB, 100 KB dan dibawah 100 KB. Berikut untuk hasil pengujiannya :

a. File Ukuran 200 KB

Tabel 1 Hasil Pengujian file Teks Ukuran 200 KB

No	Nama File	Ukuran Sebelum Kompresi	Ukuran Sesudah Kompresi	Selisih Ukuran
1	Cinta Sabar.txt	227.118 bytes	94.042 bytes	227.118 bytes
2	Aku dan Kau.txt	207.063 bytes	89.738 bytes	117.325 bytes
3	Laskar Pelangi.txt	248.958 bytes	108.362 bytes	140.596 bytes
4	Waktu Yang Fana.txt	218.616 bytes	92.168 bytes	126.448 bytes
5	Negri Bedebah 1.txt	213.943 bytes	92.276 bytes	121.667 bytes
Rata - rata				146.6308 bytes

Berdasarkan hasil pengujian kompresi file teks ukuran 200 KB pada tabel 1 mendapatkan rata-rata selisih ukuran sebesar 146.6308 bytes.

b. File Ukuran 100 KB

Tabel 2 Hasil Pengujian file Teks Ukuran 100 KB

No	Nama <i>File</i>	Ukuran Sebelum Kompresi	Ukuran Sesudah Kompresi	Selisih Ukuran
1	Negri Bedebah 2.txt	152.265 <i>bytes</i>	68.642 <i>bytes</i>	83.623 <i>bytes</i>
2	Pahlawanku.txt	144.771 <i>bytes</i>	66.592 <i>bytes</i>	78.179 <i>bytes</i>
3	Ketika Cinta Bertasbih.txt	130.590 <i>bytes</i>	66.592 <i>bytes</i>	63.998 <i>bytes</i>
4	Informatika.txt	119.689 <i>bytes</i>	55.540 <i>bytes</i>	64.149 <i>bytes</i>
5	Ips.txt	121.509 <i>bytes</i>	49.748 <i>bytes</i>	71.761 <i>bytes</i>
Rata - rata				72.342 <i>bytes</i>

Berdasarkan hasil pengujian kompresi file teks ukuran 100 KB pada tabel 1 mendapatkan rata-rata selisih ukuran sebesar 146.6308 bytes.

c. File Ukuran dibawah 100 KB

Tabel 3 Hasil Pengujian file Teks Ukuran dibawah 100 KB

No	Nama <i>File</i>	Ukuran Sebelum Kompresi	Ukuran Sesudah Kompresi	Selisih Ukuran
1	Negri Bedebah 3.txt	99.052 <i>bytes</i>	48.436 <i>bytes</i>	50.616 <i>bytes</i>
2	IndoPPkn.txt	77.802 <i>bytes</i>	33.704 <i>bytes</i>	44.098 <i>bytes</i>
3	Digital.txt	75.893 <i>bytes</i>	37.704 <i>bytes</i>	38.189 <i>bytes</i>
4	Inggris.txt	68.802 <i>bytes</i>	37.758 <i>bytes</i>	31.044 <i>bytes</i>
5	Sosiologi.txt	63.109 <i>bytes</i>	30.348 <i>bytes</i>	32.761 <i>bytes</i>
Rata - rata				39.3416 <i>bytes</i>

Berdasarkan hasil pengujian kompresi file teks ukuran dibawah 100 KB pada tabel 1 mendapatkan rata-rata selisih ukuran sebesar 39.3416 *bytes*. Berdasarkan 15 data yang diuji pada tabel 1, 2 dan 3 mendapatkan persentase total rata-rata selisih ukuran sebesar 86.1048 bytes.

Pengujian Rasio Kompresi

a. File Ukuran 200 KB

Tabel 4 Hasil Pengujian Rasio File Ukuran 200 KB

No	Nama <i>File</i>	<i>Ratio of Compress</i>	<i>Compression Ratio (CR)</i>	<i>Redudancy</i>	<i>Space Saving</i>
1	Cinta Sabar.txt	4.79	20,90%	79.10%	19.89%
2	Aku dan Kau.txt	4.57	21,88%	78.12%	20.88%
3	Laskar Pelangi.txt	4.56	21,95%	78.05%	20.95%
4	Waktu Yang Fana.txt	4.72	21,17%	78.83%	20.17%
5	Negri Bedebah 1.txt	4.63	21,62%	78.38%	20.62%

Rata - rata	4.65	21.5%	78.49%	20.50%
-------------	------	-------	--------	--------

Berdasarkan hasil pengujian rasio kompresi file teks ukuran 200 KB pada tabel 4 mendapatkan rata- rata Ratio of Compress sebesar 4.65, Compression Ratio (CR) sebesar 21.5%, Redudancy sebesar 78.49% dan Space Saving sebesar 20.50%.

b. File Ukuran 100 KB

Tabel 5 Hasil Pengujian Rasio File Ukuran 100 KB

No	Nama File	Ratio of Compress	Compression Ratio (CR)	Redudancy	Space Saving
1	Negri Bedebah 2.txt	4.38	22.84%	77.16%	21.83%
2	Pahlawanku.txt	4.35	23.01%	76.99%	22,00%
3	Ketika Cinta Bertasbih.txt	4.10	24.37%	75.63%	23.36%
4	Informatika.txt	4.27	23.4%	76.49%	22.40%
5	Ips.txt	4.84	20.67%	79.33%	19.66%
Rata - rata		4.38	22.85%	77.12%	21.85%

Berdasarkan hasil pengujian rasio kompresi file teks ukuran 100 KB pada tabel 5 mendapatkan rata- rata Ratio of Compress sebesar 4.38, Compression Ratio (CR) sebesar 22.85%, Redudancy sebesar 77.12% dan Space Saving sebesar 21.85%.

c. File Ukuran dibawah 100 KB

Tabel 6 Hasil Pengujian Rasio File Ukuran dibawah 100 KB

No	Nama File	Ratio of Compress	Compression Ratio (CR)	Redudancy	Space Saving
1	Negri Bedebah 3.txt	4.07	24.55%	75.45%	23.54%
2	IndoPPkn.txt	4.58	21.85%	78.14%	20.85%
3	Digital.txt	3.99	25.04%	74.95%	24.04%
4	Inggris.txt	3.63	27.53%	72.47%	26.52%
5	Sosiologi.txt	4.13	24.2%	75.79%	23.20%
Rata - rata		4.08	24.63%	75.36%	23.63%

Berdasarkan hasil pengujian rasio kompresi file teks ukuran dibawah 100 KB pada tabel 6 mendapatkan rata- rata Ratio of Compress sebesar 4.08, Compression Ratio (CR) sebesar 24.63%, Redudancy sebesar 75.36% dan Space Saving sebesar 23.63%.

KESIMPULAN

Berdasarkan penelitian ini, dapat disimpulkan bahwa penulis berhasil mengembangkan aplikasi kompresi file teks menggunakan metode Lempel-Ziv-Welch dengan Bahasa pemrograman Python berbasis web. Namun, perlu diperhatikan bahwa terdapat perbedaan hasil antara pengujian dalam aplikasi dan perhitungan manual, yang disebabkan oleh proses encoding saat penyimpanan data. Selain itu, metode Lempel-Ziv-Welch mampu mencapai rata-rata penurunan ukuran sebesar 86.1048 bytes dari 15 data yang diuji. Dalam 4 pengujian yang dilakukan terhadap 15 data, metode ini memiliki rata-rata Ratio of Compress (RC) sebesar 4.37, Compression Ratio (CR) sebesar 22.99%, Redundansi sebesar 76.99%, dan Space Saving sebesar 21.99%.

DAFTAR PUSTAKA

- [1] H. Hidayat, W. Zarman, And T. Pamungkas, "Implementasi Algoritma Kompresi Lzw Pada Database Server," *Komputa J. Ilm. Komput. Dan Inform.*, Vol. 2, No. 1, Pp. 7–14, 2013, Doi: 10.34010/Komputa.V2i1.74.
- [2] F. Muttaqin, H. Maulana, And G. E. Yulastuti, "Sistem Informasi Penjadwalan Latihan Dan Kunjungan Penyuluh Pertanian (Sijalutani)," *Integer J. Inf. Technol.*, Vol. 7, No. 2, Pp. 116–121, 2022, Doi: 10.31284/J.Integer.2022.V7i2.3402.
- [3] M. A. Choiri, A. Rachman, A. Purwadi, And A. K. Salim, "Rancang Bangun Sistem Informasi Perpustakaan Sekolah Berbasis Web Di Smk Islam Al-Futuhiyyah Menggunakan Model Waterfall," *Snestik (Seminar Nas. Tek. Elektro, Sist. Informasi, Dan Tek. Inform.*, Pp. 197–206, 2021.
- [4] S. R. Saragih And D. P. Utomo, "Penerapan Algoritma Prefix Code Dalam Kompresi Data Teks," *Komik (Konferensi Nas. Teknol. Inf. Dan Komputer)*, Vol. 4, Pp. 249–252, 2020, Doi: 10.30865/Komik.V4i1.2691.
- [5] I. M. Halawa And M. Syahrizal, "Analisa Perbandingan Algoritma Prediction By Partial Matching (Ppm) Dengan Lempelziv Welch (Lzw) Pada Kompresi File Teks Menggunakan Metode Eksponensial," *J. Pelita Inform.*, Vol. 6, No. 4, Pp. 464–468, 2018.
- [6] D. Riyansyah, "Perancangan Aplikasi Kompresi File Video Menggunakan Algoritma Interpolative Coding," *Komik (Konferensi Nas. Teknol. Inf. Dan Komputer)*, Vol. 3, No. 1, Pp. 392–397, 2019, Doi: 10.30865/Komik.V3i1.1618.
- [7] Y. Baskoro, "Kompresi Dengan Metode Discrete Cosine Transform (Dct) Untuk Upload Gambar Pada Aplikasi Chatroom," *J. Ris. Inov. Bid. Inform. Dan Pendidik. Inform.*, Vol. 1, No. 2, 2020.
- [8] S. Akhter And M. B. Chowdhury, "Bangla And English Text Cryptography Based On Modified Blowfish And Lempel-Ziv-Welch Algorithm To Minimize Execution Time," *Ist Int. Conf. Robot. Electr. Signal Process. Tech. Icrest 2019*, Pp. 96–101, 2019, Doi: 10.1109/Icrest.2019.8644450.
- [9] W. E. Pangesti, G. Widagdo, D. Riana, And S. Hadiani, "Implementasi Kompresi Citra Digital Dengan Membandingkan Metode Lossy Dan Lossless Compression Menggunakan Matlab," *J. Khatulistiwa Inform.*, Vol. 8, No. 1, Pp. 53–58, 2020, Doi: 10.31294/Jki.V8i1.7759.
- [10] E. S. Panggabean, "Analisa Perbandingan Algoritma Lempel Ziv Welch Dan Algoritma Deflate Pada File Teks Dengan Metode Independent Sample T-Test," *J. Pelita Inform.*, Vol. 6, No. 3, Pp. 333–336, 2018.
- [11] S. Ramadanu, "Perancangan Aplikasi Pencarian File Teks Dengan Menggunakan Algoritma Colussi Dan Algoritma Simon," *J. Ris. Komput.*, Vol. 6, No. 1, Pp. 24–29, 2019.
- [12] N. F. Rizky, S. D. Nasution, And F. Fadlina, "Penerapan Algoritma Elias Delta Codes Dalam Kompresi File Teks," *Build. Informatics, Technol. Sci.*, Vol. 2, No. 2, Pp. 109–114, 2020, Doi: 10.47065/Bits.V2i2.138.

- [13] R. A. Purba And L. Sitorus, “Analisis Perbandingan Algoritma Arithmetic Coding Dengan Algoritma Lempel Ziv Welch (Lzw) Dalam Kompresi Teks,” *J. Tek. Inform. Unika St. Thomas*, Vol. 03, Pp. 158–165, 2018.
- [14] A. D. Pradana And I. Saputra, “Penerapan Algoritma Interpolative Coding Pada Aplikasi Kompresi File Gambar,” *Komik (Konferensi Nas. Teknol. Inf. Dan Komputer)*, Vol. 3, No. 1, Pp. 220–227, 2019, Doi: 10.30865/Komik.V3i1.1592.
- [15] J. L. Phandany, A. M. Sambul, And A. S. M. Lumenta, “Comparative Study Of Digital Image Optimal Compression Algorithm Using Python,” Vol. 11, No. 1, Pp. 23–34, 2022.
- [16] A. M. Ruslida, Sapri, And D. Sartika, “Implementasi Algoritma Byte Pair Encoding Untuk Kompresi File,” Vol. 18, No. 2, Pp. 253–260, 2022.
- [17] D. Cahayati, A. M. H. Pardede, And H. Khair, “Implementasi Algoritma Elias Gamma Kompresi Pada File Teks,” Vol. 6341, Pp. 159–166, 2022.
- [18] M. B. Ibrahim And K. A. Gbolagade, “Performance Comparison Of Huffman Coding And Lempel-Ziv-Welch Text Compression Algorithms With Chinese Remainder Theorem”.
- [19] D. Oktaviani And I. M. Suartana, “Implementasi Kompresi Data Dengan Modifikasi Algoritma Lempel-Ziv-Welch (Lzw) Untuk File Dokumen,” *J. Informatics Comput. Sci.*, Vol. 1, No. 03, Pp. 128–137, 2020, Doi: 10.26740/Jinacs.V1n03.P128-137.