

Perbandingan Algoritma *Elias Delta Code* dengan *Levenstein* Untuk Kompresi *File Teks*

Ledi Varia Simanjuntak

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia

Email: ¹dledy7@gmail.com

Abstrak—File teks adalah kumpulan dari karakter – karakter atau string yang menjadi satu kesatuan. Teks yang memuat banyak karakter didalamnya selalu menimbulkan masalah pada media penyimpanan dan kecepatan waktu pada saat transmisi data. File teks merupakan file yang berisi informasi-informasi dalam bentuk teks. Data yang berasal dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam basis data merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca. Melihat kebutuhan akan pemampatan data maka perlu dilakukan kompresi atau pemampatan data yang mampu memperkecil data, sehingga memperoleh keuntungan mengurangi pemakaian ruang media penyimpanan eksternal, mempercepat proses transfer data antar media penyimpanan, memampatkan data dengan tingkat rasio lebih tinggi dari data aslinya. Proses kompresi dilakukan dengan membandingkan beberapa algoritma untuk mengatasi masalah dalam media penyimpanan, dimana metode untuk kompresi tersebut menggunakan Algoritma *Elias Delta Code* dan Algoritma *Levenstein*.

Kata Kunci: File Teks, Kompresi, Algoritma *Elias Delta Code*, Algoritma *Levenstein*

Abstract—A text file is a collection of characters or strings that become a single unit. Text that contains many characters in it always causes problems in the storage media and the speed of time when transmitting data. Text file is a file that contains information in the form of text. Data derived from word processing documents, numbers used in calculations, names and addresses in the database are examples of text data input consisting of characters, numbers and punctuation marks. Seeing the need for data compression it is necessary to compress or compress data that can reduce data, thus gaining the advantage of reducing the use of external storage media space, accelerating the process of transferring data between storage media, compressing data with a higher ratio than the original data. The compression process is done by comparing several algorithms to overcome problems in storage media, where the method for compression using the *Elias Delta Code Algorithm* and the *Levenstein Algorithm*.

Keywords: Text File, Compression, *Elias Delta Code Algorithm*, *Levenstein Algorithm*

1. PENDAHULUAN

Di dalam komunikasi digital pemakaian teknologi kompresi data sangat penting, hal ini disebabkan karena pertumbuhan penggunaan data secara elektronik lebih pesat dibandingkan dengan pertumbuhan besarnya tempat penyimpanan (storage) data dan kanal komunikasi yang tersedia. Dalam dunia internet pengguna jaringan internet meningkat lebih cepat dari pada kanal (bandwidth) yang tersedia, sehingga teknologi kompresi digunakan untuk efisiensi penggunaan kanal dan mempercepat proses pengiriman informasi.

File teks adalah file komputer yang hanya berisi teks dan tidak memiliki format khusus seperti teks tebal, teks miring, gambar, dll. File teks digunakan untuk menyimpan data atau informasi tekstual standar dan terstruktur yang dapat dibaca manusia. Ini didefinisikan dalam beberapa format yang berbeda, termasuk ASCII (American Standard Code for Information Interchange) paling populer untuk penggunaan lintas platform, dan ANSI (American National Standards Institute) untuk platform operasi berbasis Windows. Dalam sistem operasi Windows (OS), file teks dibuat menggunakan editor teks, seperti Notepad atau Word. Ini memiliki ekstensi file .txt. Kegiatan-kegiatan yang berkaitan dengan dokumen tidak lagi menggunakan kertas. Segala bentuk dokumen seperti formulir pendaftaran, laporan, bukti-bukti transaksi, dan sebagainya didistribusikan dan dikelola dalam bentuk *file*. Format txt. seringkali digunakan karena populer, serta memiliki tampilan yang tidak berubah ketika dibuka di perangkat apapun. Namun, ada sebuah masalah yang kerap dihadapi pada saat berurusan dengan dokumen berbentuk txt., yaitu masalah ukuran *file* yang relatif besar. Kompresi bertujuan untuk mengurangi ukuran data tersebut menjadi sekecil mungkin.

Kompresi bertujuan untuk meminimalkan jumlah bit yang diperlukan untuk merepresentasikan suatu file. Kompresi adalah waktu pengiriman data pada saluran komunikasi lebih singkat dan membutuhkan ruang memori dalam storage yang lebih sedikit. Kompresi dikembangkan untuk memudahkan penyimpanan dan pengiriman file. Teknik kompresi yang ada sekarang memungkinkan data dikompresi sehingga ukurannya menjadi jauh lebih kecil dari pada ukuran asli. Secara umum, metode kompresi data dapat dibagi ke dalam dua kelompok yaitu lossy dan lossless. Lossy teknik kompresi data dimana hasil kompresi dari data yang terkompresi tidak sama dengan data aslinya karena ada informasi yang hilang, namun sudah cukup untuk digunakan. Lossless Teknik kompresi dimana data hasil kompresi dapat dikompres lagi dengan hasil yang sama dari data sebelumnya. Ada beberapa algoritma yang digunakan dalam penelitian ini yaitu *Elias Delta Code* dan *Levenstein*.

Elias Delta code adalah kode universal yang mengkodekan bilangan bulat positif yang dikembangkan oleh Peter Elias. Tekniknya adalah mencari pangkat nilai binernya. Algoritma *Elias Delta Code* adalah teknik pengkompresian data yang digunakan untuk mengkompresi file yang akan merubah file menjadi bentuk biner. Begitu juga dengan Metode *Levenstein* juga digunakan untuk mengkompresi file berupa biner. Kedua algoritma

tersebut dipilih karena merupakan algoritma yang populer dan merupakan algoritma yang menjadi dasar dari beberapa algoritma yang lainnya.

Dalam kutipan “jurnal Leni Marlina, dkk [1]”. Dengan judul “Data Compression Using Elias Delta Code”. Menyatakan hasil bahwa kompresi yang dilakukan oleh Elias Delta Code sangat bagus menghemat kapasitas penyimpanan. Metode Elias Delta Code dapat memampatkan semua tipe data. Data diubah menjadi deretan set karakter dan kemudian diurutkan berdasarkan frekuensi penampilan karakter. Dan pada penelitian sebelumnya, dengan “jurnal Antoni, dkk [2]”. Dengan judul “Result Analysis Of Text Data Compression On Elias Gamma Code , Elias Delta Code, and Levenstein Code”. Menyatakan hasil Hasil pengujian dekompresi data dengan metode Elias Delta Code dan Levenstein menunjukkan bahwa Levenstein memerlukan waktu yang lebih sedikit untuk mengembalikan data hasil kompresi ke data semula .

2. METODOLOGI PENELITIAN

2.1 Kompresi

Kompresi adalah proses pengkodean informasi menggunakan bit atau informasi-*bearing unit* yang lain yang lebih rendah dari pada representasi data yang tidak terkodekan dengan suatu sistem *encoding* tertentu [3]. Proses kompresi merupakan proses mereduksi suatu data untuk menghasilkan representasi digital yang padat atau mampat (*Compact*) namun tetap dapat mewakili kuantitas informasi yang terkandung pada data tersebut. Pada citra, video, dan audio, kompresi mengarah pada minimisasi jumlah bit rate untuk representasi digital. Pada beberapa literatur, istilah kompresi sering disebut juga *source coding*, *data compression*, *bandwith compression*, dan *signal compression*.

Berdasarkan kesimpulan, kompresi dengan arti kata lain yaitu memadatkan, yang berarti mengecilkan suatu data dan proses pengubahan sekumpulan data menjadi suatu bentuk kode untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data. Kompresi menjadi sangat penting karena memperkecil kebutuhan penyimpanan data dan memperkecil kebutuhan *bandwidth*. Kompresi data bisa juga dilakukan berupa teks, gambar, audio, dan video.

2.2 Algoritma Elias Delta Code

Elias Delta Code adalah sebuah algoritma kompresi yang dibuat oleh Peter Elias menggunakan kode yang telah dia buat sebelumnya, yaitu *Elias Gamma Code*, sebagai *building block*. Dalam kode *Gamma*, Elias menambah panjang kode dalam unary (α). Dalam kode berikutnya, δ (delta), ditambahkan pada panjang kode dalam biner (β). Dengan demikian, *Elias Delta Code*, yang juga untuk bilangan bulat positif, sedikit lebih kompleks untuk dibangun [7]. *Encode* dalam bilangan bulat positif n dilakukan dalam langkah-langkah berikut:

1. Tulis n dalam biner. Bit yang paling kiri (paling signifikan) akan menjadi 1.
2. Hitung bit, hapus bit paling kiri dari n , dan tambahkan hitungan, di dalam biner, apa yang tersisa dari bit n yang paling kiri setelah telah dihapus.
3. Kurangi 1 dari hitungan di langkah 2 ,dan tambahkan jumlah nol sebanyak hasil hitungan tersebut ke awal kode.

2.3 Algoritma Levenstein

Kode yang sedikit diketahui untuk bilangan bulat non-negatif ini digagas pada tahun 1968 oleh Vladimir Levenshtein. Algoritma Levenstein digunakan secara luas dalam berbagai bidang, misalnya mesin pencari, pengecek ejaan (spell checking), pengenalan pembicaraan (speech recognition), pengucapan dialek, analisis DNA, pendeteksi pemalsuan, dan lain-lain. Algoritma ini menghitung jumlah operasi string paling sedikit yang diperlukan untuk mentransformasikan suatu string menjadi string yang lain.

Algoritma Levenstein bekerja dengan menghitung jumlah minimum pentransformasian suatu string menjadi string lain yang meliputi penghapusan, penyisipan, dan penukaran. Baik encoding dan decoding adalah proses multi langkah.

1. Set angka pertama dari C dengan 1. Letakkan kode-sejauh-ini pada string kosong.
2. Ambil nilai biner dari n tanpa angka 1 di awal dan tambahkan pada kode-sejauh-ini.
3. Nyatakan M sebagai jumlah bit yang ditambahkan pada tahap 2.
4. Jika $M \neq 0$, tambahkan C dengan 1 dan lakukan langkah 2 kembali, tetapi dengan nilai M, bukan n .
5. Jika $M = 0$, tambahkan 1 diikuti dengan 0 pada C ke kode-sejauh-ini dan berhenti.

2.4 File Teks

Teks adalah kumpulan dari karakter – karakter atau string yang menjadi satu kesatuan. Teks yang memuat banyak karakter didalamnya selalu menimbulkan masalah pada media penyimpanan dan kecepatan waktu pada saat transmisi data. *File* teks merupakan *file* yang berisi informasi-informasi dalam bentuk teks. Data yang berasal dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam basis data merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca.

3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

Analisis masalah merupakan proses mengidentifikasi sebab dan akibat dibangunnya sebuah sistem agar sistem yang akan dibangun tersebut dapat berjalan sebagaimana mestinya sesuai dengan tujuan dari sistem itu. Permasalahan yang diangkat dari penelitian ini adalah untuk mendapatkan algoritma kompresi yang lebih baik diantara algoritma *Elias Delta Code* dan *Levenstein* yang akan digunakan untuk memampatkan ukuran *file* teks. Kompresi data menggunakan algoritma *Elias Delta Code* dengan *Levenstein* dipengaruhi oleh jumlah karakter yang berada dalam *file* yang akan dimampatkan. Pada penelitian ini *file* teks yang digunakan yaitu *file* teks berekstensi *.txt.

3.2 Penerapan Algoritma Elias Delta Code

Pada tahap ini, melakukan kompresi file dengan memasukkan file teks *.txt. Isifile dokumen *.txt adalah "LEDI VARIA SIMANJUNTAK".

Langkah selanjutnya adalah dengan melakukan pembacaan isi file (scanning). Pada tahap ini dilakukan pembentukan karakter set karakter-karakter yang terdapat pada isi file. Setelah karakter set diperoleh maka frekuensi kemunculan untuk tiap-tiap karakter akan dihitung. Berikut hasil scanning terhadap file dokumen *.txt :

Data string = LEDI VARIA SIMANJUNTAK
 Karakter set = $\Sigma = \{ L, E, D, I, SP, V, A, R, S, M, N, J, U, T, K \}$
 Jumlah bit = 176 bit
 Frekuensi karakter =
 A = 4 E = 1 M = 1
 I = 3 D = 1 J = 1
 N = 2 V = 1 U = 1
 SP = 2 R = 1 T = 1
 L = 1 S = 1 K = 1

Setelah karakter set dan frekuensi kemunculan tiap karakter diketahui maka selanjutnya karakter set akan diurutkan berdasarkan frekuensi kemunculannya mulai dari karakter dengan frekuensi kemunculan terbesar dengan karakter frekuensi kemunculan terkecil. Jika terdapat lebih dari satu karakter dengan frekuensi kemunculan yang sama maka diurutkan berdasarkan abjad. Untuk hasil pengurutan karakter set dan frekuensi karakter dapat dilihat pada tabel 1. dibawah ini.

Tabel 1. String yang belum Dikompresi

No	Char	ASCI	Binary	Freg	Bits
1	A	65	01000001	4	32
2	I	69	01001001	3	24
3	N	78	01001110	2	16
4	Space	32	01001100	2	16
5	L	76	01001100	1	8
6	E	69	01000101	1	8
7	D	68	01000100	1	8
8	V	86	01010110	1	8
9	R	82	01010010	1	8
10	S	83	01010011	1	8
11	M	77	01001101	1	8
12	J	74	01001010	1	8
13	U	85	01010101	1	8
14	T	84	01010100	1	8
15	K	75	01001011	1	8
Total Bit					176

Tahap selanjutnya adalah pembentukan set karakter. dengan membentuk tabel elias delta code. Setelah itu karakter yang terdapat pada file dokumen yang akan dikompresi diganti dengan kode yang terdapat pada tabel diatas. Setelah diganti, hitung jumlah bit untuk tiap karakter pada file dokumen dengan elias delta code dapat dilihat pada tabel 2. dibawah ini :

Tabel 2. String yang sudah Dikompresi dengan Elias Delta Code

No	Char	Freg	Elias Delta Code	Bits	Bits x Freg
1	A	4	1	1	4
2	I	3	0100	4	12
3	N	2	0101	4	8
4	Space	2	01100	5	10

$$= 32,9$$

d. Space saving (SS) = (1 – Compression Rasio) x 100%
 = 1-67x 100%
 = 66x 100%
 = 6.6%

3.4 Penerapan Metode Eksponensial

Dalam menghitung dan membandingkan proses kompresi dari kedua algoritma tersebut adalah sebagai berikut :

1. Menentukan alternatif
 Untuk menganalisa perbandingan kecepatan antara algoritma Elias Delta Code dan algoritma levenstain dalam melakukan kompresi maka perlu dilakukan penentuan algoritma mana yang akan digunakan sebagai algoritma kompresi.
2. Menentukan Kriteria
 Untuk dapat membandingkan kedua algoritma tersebut, maka selanjutnya perlu dilakukan penentuan kriteria dalam menganalisa proses dan cara kerjanya. Untuk kriteria dapat dilihat pada tabel berikut :

Tabel 4. Penentuan Kriteria

Kriteria	Keterangan
Ratio Of Compression (RC)	Nilai perbandingan antara ukuran bit data sebelum dikompresi dengan ukuran bit data yang telah dikompresi.
Compression Ratio (CR)	Persentase perbandingan antara data yang sudah dikompresi dengan data yang belum dikompresi.
Redundancy (RD)	Hasil penilaian nilai rasio dengan hasil nilai rasio kompresi
Space Saving (SS)	Selisih antara data yang belum dikompresi dengan besar data yangdikompresi.

3. Pemberian nilai pada setiap kriteria
 Pada kriteria yang telah dibentuk harus diberikan nilai. Nilai tersebut dapat dilihat pada contoh dibawah ini yang dimana nilainya diambil berdasarkan analisa algoritma Elias Delta Code dan algoritma levenstain sebelumnya.

Tabel 5. Pemberian Nilai Pada Setiap Kriteria

Alternative	Contoh	Criteria				SS
		RC		CR		
		SB	SD	SD	SB	
Algoritma Elias Delta Code	LEDI VARIA	176 (bit)	121 (bit)	121 (bit)	176 (bit)	1-CR
Algoritma Levenstain	SIMANJUNTAK	176 (bit)	118 (bit)	118 (bit)	176 (bit)	1-CR

Keterangan:

- RC : Ratio Of Compression
- CR : Compression Ratio
- SS : Space Saving
- SB : Sebelum Dikompresi
- SD : Sesudah Dikompresi

4. Menentukan hasil atau prioritas keputusan
 Setelah diperoleh nilai akhir atau total nilai dari masing-masing alternatif, maka tahapan selanjutnya yang perlu dilakukan adalah menentukan prioritas keputusan berdasarkan nilai dari masing-masing alternatif. Hasil prioritas keputusan dapat dilihat pada tabel dibawah ini :

Tabel 6. Prioritas keputusan

Alternatif	RC	CR	SS	Rangking
Algoritma Elias Delta Code	1,4	68,8%	6,75%	1
Algoritma Levenstain	1,5	67%	6,6%	2

Pada tabel diatas menjelaskan bahwa alternatif yang memiliki total nilai tertinggi (6,75%) maka akan diposisikan rangking pertama, hal ini dikarenakan semakin besar (SS) maka kompresinya semakin bagus dan lebih cepat. Berdasarkan analisa tersebut, maka algoritma Elias Delta Code yang menjadi algoritma terbaik dalam melakukan proses kompresi.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan penulis pada kompresi file menggunakan teknik pengolahan citra yang dibangun maka dapat disimpulkan :

1. Dari uji coba yang telah dilakukan dengan menggunakan algoritma Elias Delta Code dan Levenstein, program hanya bisa mengkompresi file yang berekstensi *.txt.
2. Program kompresi file yang diterapkan kedalam bahasa pemrograman Visual Basic .Net 2008 cukup bagus dalam proses kompresi maupun dekompresi file.
3. Nilai rasio kompresi dipengaruhi oleh isi dari file yang dikompresi. Semakin banyak pengulangan karakter pada suatu file yang dikompresi maka semakin tinggi pula rasio kompresinya.

REFERENCES

- [1] E. B. Nababan dan M. Zarlis, "Results Analysis of Text Data Compression On Elias Gamma Code, Elias Delta Code and Levenstein Code," *International Journal of Science and Advanced Technology*, vol. 4, no. 9, pp. 17-24, 2014.
- [2] L. Marlina, A. P. U. Siahaan, H. Kurniawan dan I. Sulistianingsih, "Data Compression Using Elias Delta Code," *Internasional Journal Of Recent Trends in Engineering & Research*, vol. 3, no. 8, pp. 210-217, 2017.
- [3] D. Putra, *Pengolahan Citra Digital*, Yogyakarta: ANDI, 2010.
- [4] I. D. Irawati, MT, L. V. Yovita, M.T dan T. a. Wibowo, M.T, *Jaringan Komputer Dan Data Lanjut*, 2015: DEEPUBLISH, Yogyakarta.
- [5] E. S. Mulyanta, *Pengolahan Digital Image Dengan Photoshop CS2*, Yogyakarta: ANDI, 2006. Gorontalo", *Jurnal Administrasi*, vol.I, pp. 54-65. 2010