

## Perbandingan Algoritma Stout Code Dan Algoritma Elias Delta Code Dalam Kompresi File Video

Lili Rahmawati M.Yus

Fakultas Ilmu Komputer dan Teknologi Informasi, Teknik Informatika, Universitas Budi Darma, Medan, Indonesia  
Email: lilirahmawati209@gmail.com

**Abstrak**-Teknologi kompresi data adalah proses pengubahan sekumpulan data menjadi bentuk kode yang lebih rendah dari representasi data yang tidak terkodekan dengan suatu encoding tertentu. Kompresi data dilakukan dengan mengkodekan setiap karakter menjadi pesan kode yang lebih pendek, dengan tujuan untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk pengiriman data. Pengompresi digunakan untuk mempercepat proses pengiriman dari karakter yang tidak terkodekan menjadi karakter terkodekan tertentu. Dengan melakukan kompresi, data yang besar akan berkurang ukurannya sehingga dapat menghemat kapasitas penyimpanan. Dalam penelitian ini menggunakan Algoritma Stout Code, dengan menggunakan metode tersebut hasil kompresi sangat baik dalam melakukan transmisi data dan pemindahan alokasi file video lebih mudah dan cepat. Dengan melakukan kompresi, data yang besar akan berkurang ukurannya sehingga dapat menghemat ruang penyimpanan. Ada banyak jenis metode ataupun algoritma dalam kompresi file video, tetapi dalam algoritma ini penulis hanya ingin mengetahui proses kompresi menggunakan algoritma Stout Code dan Elias Delta Code dimana penulis telah berhasil mengompres suatu file video dengan algoritma Elias Delta Code, maka algoritma Elias Delta Code yang menjadi algoritma terbaik dalam proses pengkompresian.

**Kata Kunci** : Kompresi file video; Algoritma Stout Code; Elias Delta Code.

**Abstract**-Data compression technology is the process of converting a set of data into a lower coded form of data representation that is not encoded with a certain encoding. Data compression is done by encoding each character into a shorter code message, with the aim of saving the need for storage space and time for data transmission. Compressor is used to speed up the process of sending from non-encoded characters to certain encoded characters. By compressing, large data will be reduced in size so that it can save storage capacity. In this study using the Stout Code Algorithm, by using this method the compression results are very good in transmitting data and transferring video file allocations more easily and quickly. By compressing, large data will be reduced in size so that it can save storage space. There are many types of methods or algorithms in video file compression, but in this algorithm the author only wants to know the compression process using the Stout Code and Elias Delta Code algorithms where the author has succeeded in compressing a video file with the Elias Delta Code algorithm, then the Elias Delta Code algorithm is the the best algorithm in the compression process.

**Keywords** : Video file compression; Stout Code Algorithm; Elias Delta Code.

### 1. PENDAHULUAN

Zaman saat ini kemajuan teknologi sangatlah berkembang dan memudahkan untuk bertukar informasi secara meluas sehingga perkembangan computer semakin hari semakin meningkat karena hampir semua aktivitas dan pekerjaan menggunakan kecanggihan teknologi yang dimiliki oleh komputer, saat ini banyak manusia yang berminat untuk menonton video melalui sosmed atau *gadget* seperti aplikasi *youtube, instagram, tiktok, ome tv, telegram* dan lain-lain. Sehingga membuat ruang penyimpanan sangat penuh dan membuat ruang penyimpanan yang begitu besar. Oleh karena itu kita perlu melakukan proses kompres untuk memampatkan *file* video menjadi lebih kecil. Kompresi data *file* semakin banyak diterapkan pada berbagai jenis data, teks, gambar, *audio*, dan *video*[1]. Ada banyak algoritma kompresi, salah satu adalah algoritma *Stout Code*, dimana kode panjang *variable* untuk bilangan bulat, mirip dengan *Elias Omega*. Pada pengguna algoritma *Stout Code* dengan menggunakan aplikasi *HxD*, dalam penelitian ini akan membandingkan kedua algoritma karena kedua algoritma tersebut termasuk kedalam jenis kompresi *lossless*, dengan langkah-langkah awal pengkompresian memiliki kesamaan yang dimana sebelum melakukan pengkompresian harus terlebih dahulu diambil nilai *hexadecimal* dengan menggunakan *HxD*, setelah mendapatkan nilai *hexadecimal* dari *file* video, maka dilakukan penyusunan berdasarkan banyaknya nilai frekuensi dari setiap *sample* yang diambil, kemudian langkah selanjutnya sesuai dengan ketentuan algoritma masing-masing[2][3][4]. Maka dengan adanya kesamaan maka kedua algoritma cocok untuk dibandingkan dalam pengkompresian *file* video. agar mendapatkan hasil yang optimal[5][6][7]. *Video* memiliki 3 dimensi yaitu berupa 2 dimensi *horizontal* dan *vertical*, dan 1 dimensi waktu. Di dalam *video* terdapat 2 hal yang bisa dikompresi yaitu *frame (still image)* dan *audionya*.

Kompresi adalah sebuah proses dimana *file* (video, gambar, audio dan teks) dapat ditransformasikan ke *file* lain(terkompresi), sehingga *file* yang akan di kompresi dapat sepenuhnya kembali seperti *file* aslinya tanpa kehilangan informasi yang sebenarnya. Dalam kompresi bit data kompres untuk memperkecil ukuran data yang berulang. Dengan adanya kompresi, akan mudah untuk menyimpan lebih banyak data bila dibandingkan dengan data yang tidak dikompres serta data yang dikompres dapat ditransfer lebih cepat karena ada jumlah bit yang lebih sedikit dari aslinya. Proses ini akan sangat berguna jika kita ingin menghemat ruang penyimpanan. Untuk mengetahui algoritma yang sesuai maka dilakukan perbandingan kinerja antara algoritma *Stout Code* dan algoritma *Elias Delta Code* karena kedua algoritma tersebut kedalam jenis kompresi *lossless*. Maka penulis mencoba untuk membandingkan kedua algoritma *Stout Code* dan *Elias Delta Code* untuk menhemat data penyimpanan *file* video saat mengirim *file* video[8].

## 2. METODOLOGI PENELITIAN

### 2.1 Kompresi

Kompresi data adalah proses memperkecil ukuran data untuk menghasilkan tampilan digital yang padat atau mampat (*Compact*) namun mampu mempresentasikan jumlah informasi yang terkandung pada data tersebut. Kompresi data adalah teknik komputer untuk mengkompresi data sehingga hanya memakan sedikit ruang penyimpanan lebih lebih efisien untuk mengurangi waktu pertukaran data tersebut. Teknik kompresi dimana data hasil kompresi dapat dikompresi lagi dan hasilnya tetap sama seperti data sebelum. Kompresi data juga dapat mengurangi kelambatan penyimpanan data, yaitu ketika ukuran data cukup besar dan data didalam nya begitu mengandung banyak karakter berulang. Pada *file* video, kompresi pada minimasi jumlah *bit rate* untuk mengrepresentasi digita, ada dua cara melakukan kompresi yaitu lossless compression dan lossy compression [2][3][4].

### 2.2 File Video

Video adalah suatu bentuk teknologi untuk merekam, menangkap, memproses dan mentransmisikan dan mengatur ulang gambar yang bergerak. Video dapat disimpan menggunakan flim, video, televise, video atau media non computer lainnya. Awalnya, informasi video ini disimpan dalam analog sebagai perubahan isi gelombang, yang mewakili perubahan warna dan kecerahan (*brightness*) gambar yang direkam, disisi lain, komputer digital hanya dapat menyimpan dan memproses data biner. Oleh karena itu, industry komputer didefinisikan warna dalam besaran 24-bit yang dapat digunakan untuk menyimpan sekitar 16,7 juta kemungkinan warna. Dengan demikian, data video dapat disimpan secara digital sebagai titik-titik, masing-masing mewakili warna tberbeda, dan titik-titik tersebut disusun sebagai satu kesatuan akan membentuk suatu gambar yang lengkap [5][9].

### 2.3 Algoritma Stout Code

Dalam artikel singkatnya, *Queintin Stout* memperkenalkan dua keluarga rekursi, RI dan SI, dari simbol integer panjang variable yang mirip dan lebih umum dari pada simbol dari *Elias Omega* dan *Even-Rodeh*. *Code* yang kuat secara universal dan asimtotik optimal sebelum membaca. Sebelum itu pembaca membaca bilangan bulat  $n$  yang diberikan oleh  $L=1 \log_2 n$  (persamaan).

Kedua kode tergantung pada pemilihan parameter integer 1 setelah nilai (lebih besar dari atau sama dengan 2) telah dipilih untuk 1 kata sandi di grup pertama yang terdiri dari awalan RI( $n$ ) dan akhiran 0 $n$ . Sufiksnya adalah nilai biner  $n$  dalam  $L$  bit, didahului dengan biner 0. Awalan RI( $n$ ) terdiri dari satu set panjang mulai dari panjang  $L$  dari  $n$ , yaitu.  $H$ . sebelum  $L_1$ , dan  $L$  kemudian panjang  $L_2$  dari  $L_1$ , dan seterusnya, sampai panjang  $L_1$  cukup pendek untuk masuk ke dalam grup 1-bit. Perhatikan bahwa grup panjang (kecuali mungkin yang paling kiri) dimulai dengan 1. Oleh karena itu, 0 menunjukkan akhir dari kelompok panjang [10].

Misalnya, kami memilih  $l=2$  dan mengkodekan bilangan bulat 985-bit yang nilai pastinya tidak menjadi masalah. Baris sufiks 986 - bit 0 $n$  dan awalan dengan grup  $L = 985 = 11110110012$  panjang  $L_1$  grup ini 1010 = 10102 panjang  $L_2$  grup kedua = 1002 dan panjang  $L_3$  start 3 = 112 selesai, jadi kata sandinya adalah 11|100|1010 |111001|0 $n$  Perhatikan berapa lama grup dimulai dengan 1, ini berarti bahwa setiap grup diikuti oleh 1, kecuali grup panjang terakhir, yang diikuti oleh 1. 0. Analisis sederhana. Decoder mulai membaca bit pertama. Ini adalah panjang grup berikutnya. grup yang lebih panjang dibaca sampai grup yang diikuti oleh 0 ditemukan. Ini menunjukkan bahwa kelompok  $n$  berikutnya adalah  $n$  itu sendiri. Defenisi rekursif dari awalan R1 mudah dibaca dan dipahami. Penulis menggunakan symbol  $L=1? \log_2 n?$  dan  $B(n,1)$  adalah representasi biner 1-biner (Kode Beta) dari bilangan bulat  $n$ . Jadi,  $B(12,5) = 01100$ . Menggunakan lebih banyak biner tanda. Penggunaan algoritma *Stout Code* dalam program kompresi *file* video adalah untuk mengubah ukuran *file* asli menjadi *file* yang lebih kecil. Proses mengkompresi *file* video dimulai dengan memilih *file* video yang akan dikompresi, kemudian *file* tersebut akan dikompresi dengan algoritma *Stout Code* sehingga menghasilkan ukuran *file* terkompresi yang lebih kecil [11].

### 2.4 Algoritma Elias Delta Code

*Elias delta code* adalah kode universal yang mengkodekan bilangan bulat pusat, yang dikembangkan oleh *Peter Elias*. *Elias Delta Code* adalah salah satu dari tiga *Elias Code* yang dipopuleri oleh *Peter Elias*. Didalam kode ini, setiap karakter diakili dengan memetakan kode sumber ke sejumlah variabel bit [8].

Aturan pengkodekan suatu bilangan dengan kode *Elias Delta Code* adalah sebagai berikut:

#### 1. Encoding

Berikut langkah-langkah *Encoding* pada algoritma *Elias Delta Code*:

- Tuliskan bilangan dalam bentuk biner
- Hapus bit paling kiri dari bilangan biner
- Hitung sisa bit pada langkah 2 dan tambahkan 1, kemudian tuliskan hasilnya dalam bentuk biner sebelum angka pada langkah 2
- Kemudian hitung jumlah bit pada langkah 3 dan kurangi 1, tulis angka dalam biner 0, lalu masukan awalan angka pada langkah 3.

Tabel 1. *Elias Delta Code*

<i>Input</i>	<i>Code Elias DeltaCode</i>
0	1
1	0100
1	0101
2	01100
2	01101
2	01110
2	01111
2	00100000
3	00100001
3	00100010
3	00100011
3	00100100
3	00100101
3	00100110
3	00100111
4	001010000
4	00101001

### 2.4 Metode Perbandingan *Exponential*

Metode perbandingan *Exponential* adalah metode penentuan urutan alternatif dengan beberapa kriteria yang lebih dari satu. Rumus dari metode *Exponential* adalah [12]:

$$\text{Total Nilai (TN}_i\text{)} = \sum_{j=1}^m (V_{ij})B_j \dots\dots\dots (1)$$

Keterangan :

- TN<sub>i</sub> = Total nilai alternatif ke-i.
- RK<sub>ij</sub> = Derajat kepentingan relatif kriteria ke-j pada keputusan ke-i, yang dapat dinyatakan dengan skala ordinal (1,2,3,4,5).
- TKK<sub>j</sub> = Derajat kepentingan kriteria keputusan ke-j; TKK<sub>j</sub> > 0;
- M = Jumlah kriteria keputusan.
- n = Jumlah pilihan keputusan.
- J = 1,2,3,... m; m = jumlah kriteria.
- I = 1,2,3.....n; n = jumlah pilihan alternatif[6].

## 3. HASIL DAN PEMBAHASAN

Tahap analisa merupakan tahap yang berpengaruh dan menentukan terhadap tahap selanjutnya. Analisa pada penelitian ini adalah menghitung kompresi *file* video dengan format *MP4*, *file* video mempunyai ukuran yang cukup besar, semakin lama waktu *file* video maka makin besar pula penyimpanan yang diperlukan, dan proses pengiriman *file* tergolong cukup lama. Untuk mengetahui proses kompresi, dekompresi dan perbandingan dapat menggunakan Algoritma *Stout Code* dan *Elias Delta Code*.

Agar mengetahui Algoritma kompresi yang sesuai, maka dilakukan perbandingan kinerja antara algoritma *Stout Code* dan *Elias Delta Code* karena kedua algoritma tersebut termasuk kedalam jenis kompresi *lossless*, maka penulis mencoba untuk membandingkan kedua algoritma *Stout Code* dan *Elias Delta Code* untuk menghemat penyimpanan *file* video saat mengirim *file* video. Dan *file* video terkompresi dan dapat dikirim dengan mudah dan cepat, karena ukuran data yang lebih kecil.

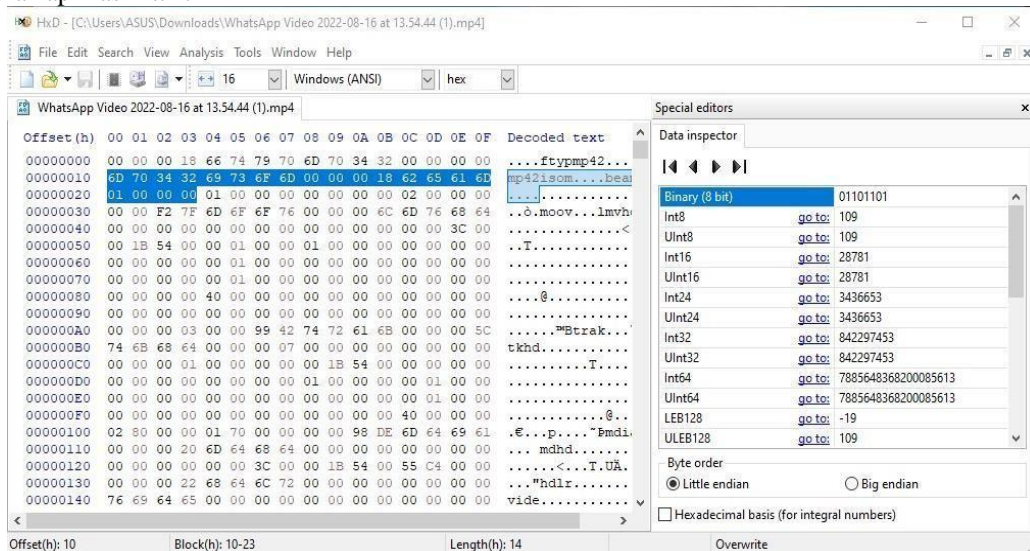
### 3.1 Penerapan Algoritma *Stout Code*

Input *file* yang digunakan adalah *file* video. Berikut merupakan sample video yang akan dilakukan proses kompresi seperti pada table 2 berikut ini:

Tabel 2. Informasi *File* Video

Keterangan	
Format	MP4
Nama <i>file</i>	Pulau Jemur
Ukuran	12,42 MB
Durasi	1.56 Menit

Berdasarkan Tabel diatas, maka file vidio tersebut dapat diubah menjadi nilai *Hexadecimal* dengan menggunakan aplikasi *HxD*.



Gambar 1. Nilai *Hexadecimal* Sample File Vidio.

Tabel 3. Nilai *Hexadecimal* Yang Belum Dikompresi

N	<i>Hexadecimal</i>	<i>Biner</i>	<i>Frekuensi</i>	<i>Bit</i>	<i>Frekuensi x Bit</i>
1	00	00000000	6	8	48
2	6D	01101101	3	8	24
3	70	01110000	1	8	8
4	34	00110100	1	8	8
5	32	00110010	1	8	8
6	69	01101001	1	8	8
7	73	01110011	1	8	8
8	6F	00110111	1	8	8
9	18	11000000	1	8	8
10	62	11000100	1	8	8
11	65	11001010	1	8	8
12	61	11000010	1	8	8
Total <i>Bit</i>					152 <i>Bit</i>

Setelah bilangan *Hexadecimal* diurutkan berdasarkan frekuensi kemunculan dan di dapatkan nilai *biner*, langkah selanjutnya adalah menghitung *bit* dan pengurutan kode Algoritma *Stout Code* dan mendapatkan *file bit* terkompres. Proses kompresi *file* vidio dapat dilihat pada tabel dibawah ini.

Tabel 4. *Hexadecimal* Yang Sudah Dikompresi Pada Algoritma *Stout Code*.

No	<i>Hexadecimal</i>	<i>Codeword</i>	<i>Frekuensi</i>	<i>Bit</i>	<i>Frekuensi x Bit</i>
1	00	01	6	2	12
2	6D	10	3	2	6
3	70	11	1	2	2
4	34	00100	1	5	5
5	32	00101	1	5	5
6	69	00110	1	5	5
7	73	00111	1	5	5
8	6F	011000	1	6	6
9	18	18	18	18	18
10	62	62	62	62	62
11	65	65	65	65	65
12	61	61	61	61	61
Total <i>Bit</i>					70 <i>Bit</i>

Berdasarkan data pada table diatas dapat dibentuk dalam nilai kompresi lokasi nilai *Hexadecimal* dalam sample keluaran sebelum kompresi yaitu, **6D,70,34,32,69,73,6F,6D,00,00,00,18,62,65,61,6D,00,00,00** menjadi nilai biner. "01010101011010101100100001010011000111011000011001011010011011011100"

Kemudian sebelum didapatkan nilai hasil akhir kompresi adalah penambahan urutan *bit*, yaitu *padding* dan *flagging*.

Penambahan *Padding*:

Bit Akhir  $7 - n + "1"$  Bit Akhir =  $7 - 6 + "1" = 01$

Penambahan *Flagging*:

Bit Akhir  $9 - n$

Bit Akhir =  $9 - 6 = 3 = 00110011$

Setelah ditambahkan semua menjadi:

"01010101010110101011001000010100110001110110000110010110100110110111000100110011"

### 3.1.1 Penerapan Metode *Elias Delta Code*

Pada penelitian ini akan membahas tentang pengkompresian *file* video dengan menerapkan algoritma *Elias Delta Code* yang merupakan salah satu teknik kompresi *lossless*.

Nilai *Hexadecimal* = 6D, 70, 34, 32, 69, 73, 6F, 6D, 00, 00, 00, 18, 62, 65, 61, 6D, 00, 00, 00.

**Tabel 5.** String Sebelum Dikompresi Dengan Algoritma *Elias Delta Code*

No	Hexadecimal	Frekuensi	ASCII Code(Decimal)	ASCII Code(Biner)	Bit	Freq x Bit
1	00	6	00	00000000	8	48
2	6D	3	109	01101101	8	24
3	34	1	52	00110100	8	8
4	32	1	50	00110010	8	8
5	69	1	105	01101001	8	8
6	73	1	115	01110011	8	8
7	6F	1	111	01101111	8	8
8	70	1	112	01110000	8	8
9	18	1	24	00011000	8	8
10	62	1	98	01100010	8	8
11	65	1	101	01100101	8	8
12	61	1	97	01100001	8	8
Total Bit						152

Setelah diurutkan secara berurutan dan kemudian memasukan *file* kompresi, dapat dilihat dibawah ini:

**Tabel 6.** String Setelah Dikompresi Dengan Algoritma *Elias Delta Code*.

No	Hexadecimal	<i>Elias Delta Code</i>	Frekuensi	Bit	Frekuensi x Bit
1	00	1	6	1	6
2	6D	0100	3	4	12
3	34	0101	1	4	4
4	32	01100	1	5	5
5	69	01101	1	5	5
6	73	01110	1	5	5
7	6F	01111	1	5	5
8	70	00100000	1	8	8
9	18	00100001	1	8	8
10	62	00100010	1	8	8
11	65	00100011	1	8	8
12	61	00100100	1	8	8
Total Bit					82

Hexadecimal yang telah dibaca adalah "6D, 70, 34, 32, 69, 73, 6F, 6D, 00, 00, 00, 18, 62, 65, 61, 6D, 00, 00, 00", maka hasil string bit yang telah dikompresi adalah: "0100001000000101011000110101110011110100111001000010010001000100011001001000100111"

Penambahan *Flagging*:

$(7 - n + 1)$

$(7 - 2 + "1" = 000001)$

Penambahan *Flagging*:

Bit Akhir  $9 - n$

Bit Akhir =  $9 - 2 = 7 = 00110111$



### 3.1.2 Penerapan Metode *Exponential*

Metode perbandingan *Exponential* merupakan suatu metode yang untuk menentukan urutan prioritas alternatif keputusan dengan kriteria jamak. Adapun rumus dari metode *exponential* ini adalah sebagai berikut:

$$\text{Total Nilai (TN}_i\text{)} = \sum_{j=1}^m (V_{ij})B_j \dots\dots\dots (1)$$

a. *Ratio Of Compression (Rc)*

$$\text{Stout Code} = \frac{152}{70} = 2,17$$

$$\text{Elias Delta Code} = \frac{152}{82} = 1,85$$

b. *Compression Rasio (Cr)*

$$\text{Stout Code} = \frac{70}{152} * 70\% = 0,46\%$$

$$\text{Elias Delta Code} = \frac{152}{82} * 70\% = 1,85\%$$

c. *Space Saving (Ss)*

$$\text{Stout Code} = 100\% - 0,46\% = 0,9954\%$$

$$\text{Elias Delta Code} = 100\% - 1,85\% = 0,9815\%$$

d. *Redudancy*

$$\text{Stout Code} = \frac{152-70}{152} * 100\% = 0,53\%$$

$$\text{Stout Code} = \frac{152-82}{152} * 100\% = 0,46\%$$

**Tabel 7.** Pemberian Nilai Pada Setiap Parameter.

Alternatif	Parameter						SS
	CR		RC		RD		
	SB	SD	SB	SD	SB	SD	
Algoritma Stout Code	152 (bit)	70 (bit)	152 (bit)	70 (bit)	152 (bit)	70 (bit)	1 - CR
Algoritma Elias Delta Code	152 (bit)	82 (bit)	152 (bit)	82 (bit)	152 (bit)	82 (bit)	1 - CR

Keterangan:

- CR : *Compresion Rasio*
- RC : *Ratio Of Compresion*
- SS : *Space Saving*
- SB : Sebelum Dikompresi
- SD : Sesudah Dikompresi

**Tabel 8.** Perbandingan *Stout Code* Dan *Elias Delta Code*

Alternatif	RC	CR	SS	RD	Exponential	Rangking
Algoritma Stout Code	2,17	0,46%	0,9954 %	0,53%	7,866	1
Algoritma Elias Delta Code	1,85	1,85%	0,9815 %	0,46%	6,242	2

## 4. KESIMPULAN

Berdasarkan analisa dan hasil yang telah dilakukan dalam perbandingan algoritma *Stout Code* dan algoritma *Elias Delta Code* untuk kompresi *file* video, maka didapatkan kesimpulan bahwa prosedur untuk mengkompresi *file* video dimulai dari memilih *file* vidieo yang akan dikompresi, lalu *file* tersebut akan dikompresi dengan menggunakan *Stout Code* dan algoritma *Elias Delta Code* untuk mengetahui algoritma mana yang paling terbaik

dalam pengkompresian *file* video. Pada perbandingan algoritma *Stout Code* dan algoritma *Elias Delta Code* untuk melakukan kompresi pada *file* video dengan mengambil nilai *Ratio Of Compression*, *Compression Rasio*, *Space Saving*, dan *Redudancy*, dari proses kedua kompresi algoritma, kemudian hasil akhir dari algoritma *Stout Code* dan algoritma *Elias Delta Code* dibandingkan dengan metode *Exponential*. Dari hasil percobaan algoritma *Stout Code* merupakan algoritma yang memiliki rasio terbaik atau tertinggi dalam pengkompresian dibandingkan dengan algoritma *Elias Delta Code* dimana pengkompresian dengan algoritma *Stout Code* diperoleh *Ratio Of Compression* 2,17, *Compression Rasio* 0,46%, *Spacing Saving* 0,9954, sedangkan dengan algoritma *Elias Delta Code* diperoleh *Ratio Of Compression* 1,85, *Compression Rasio* 0,53%, *Spacing Saving* 0,9947%. Dengan menerapkan metode *Exponential* maka diketahui bahwa algoritma *Elias Delta Code* kurang efektif untuk pengkompresian *file* video dibandingkan dengan algoritma *Stout Code* dimana diperoleh hasil dari algoritma *Stout Code*.

## REFERENCES

- [1] D. Rachmawati, M. A. Budiman, and M. A. Subada, "Comparison study of Fibonacci code algorithm and Even-Rodeh algorithm for data compression," *J. Phys. Conf. Ser.*, vol. 1321, no. 3, 2019, doi: 10.1088/1742-6596/1321/3/032015.
- [2] "Kompresi video - Wikipedia bahasa Indonesia, ensiklopedia bebas."
- [3] T. P. Sari, S. D. Nasution, and R. K. Hondro, "Penerapan Algoritma Levenstein Pada Aplikasi Kompresi File Mp3," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 2, no. 1, pp. 439–443, 2018, doi: 10.30865/komik.v2i1.946.
- [4] R. Handayani, "Perancangan Aplikasi Kompresi File Audio Menerapkan Algoritma Universal Codes," vol. 5, pp. 324–330, 2021, doi: 10.30865/komik.v5i1.3735.
- [5] P. Bahasan and T. Belajar, "Kompresi Audio/Video".
- [6] Y. Devianto and S. Dwiasnati, "Aplikasi Pengambilan Keputusan Indeks Kepuasan Masyarakat Dengan Metode Perbandingan Eksponensial (MPE) Pada Unit Pelayanan Masyarakat Dengan Alat Microcontroller Sebagai Alat Bantu Survey," *J. Ilm. FIFO*, vol. 10, no. 1, p. 13, 2018, doi: 10.22441/fifo.v10i1.2946.
- [7] Lamsah and D. P. Utomo, "Penerapan Algoritma Stout Codes Untuk Kompresi Record Pada Databade Di Aplikasi Kumpulan Novel," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 4, no. 1, pp. 311–314, 2020, doi: 10.30865/komik.v4i1.2710.
- [8] M. Simangunsong, "Perbandingan Algoritma Elias Delta Code Dan Unary Coding Dalam Kompresi Citra Forensik," *Resolusi Rekayasa Tek. Inform. dan ...*, vol. 1, no. 1, pp. 18–26, 2020, [Online]. Available: <http://djournal.com/resolusi/article/view/9>
- [9] E. Sihotang, "Perbandingan Kinerja Algoritma Elias Delta Code Dan Algoritma Reverse Unary Codes Dalam Kompresi Citra," p. 95, 2019.
- [10] R. Manalu, "Implementasi Algoritma Massey-Omura dan Algoritma Elias Delcta Code Pada Pengamanan dan Kompresi File Dokumen," *J. Sist. Komput. dan Inform.*, vol. 1, no. 3, p. 239, 2020, doi: 10.30865/json.v1i3.2152.
- [11] Apijuddin, "Perancangan Aplikasi Kompresi File Teks Menggunakan Algoritma Stout Codes," *Pelita Inform. Inf. dan Inform.*, vol. 9, no. 3, pp.183–188, 2021.
- [12] L. V Simanjuntak, "Perbandingan Algoritma Elias Delta Code dengan Levenstein Untuk Kompresi File Teks," *J. Comput. Syst. ...*, vol. 1, no. 3, pp. 184–190, 2020, [Online]. Available: <https://ejurnal.seminar-id.com/index.php/josyc/article/view/168>