

# Perancangan Aplikasi Kompresi File Gambar Menggunakan Algoritma Elias Omega Code

Puji Lestari\*, Surya Darma Nasution

Fakultas Ilmu Komputer dan Teknologi Informasi, Teknik Informatika, Universitas Budi Darma, Medan

Jl. Sisingamangaraja No.338, Siti Rejo I, Kec. Medan Kota, Kota Medan, Sumatera Utara, Indonesia

Email: <sup>1,\*</sup>pujilestaari4@gmail.com, <sup>2</sup>surya.darma.nasution1@gmail.com

Email Penulis Korespondensi: pujilestaari4@gmail.com

Submitted: 24/05/2022; Accepted: 14/06/2022; Published: 31/07/2022

**Abstrak**—Kebiasaan untuk mengumpulkan data membuat kapasitas ruang penyimpanan yang besar menjadi lebih kecil sehingga membuat pengguna komputer menjadi sulit untuk menyimpan data yang banyak atau data yang berukuran besar. Salah satu file yang berukuran besar dan biasanya dikumpulkan oleh manusia yaitu file gambar, hal ini memakan banyak ruang penyimpanan. Salah satu solusi untuk mengatasi masalah tersebut yaitu dengan menggunakan algoritma kompresi seperti elias omega code sehingga ukuran file gambar yang besar menjadi lebih kecil. Algoritma Elias Omega Code merupakan algoritma kompresi yang berjenis lossless compression, dimana hasil kompresinya dapat dikembalikan seperti semula tanpa ada kehilangan bit. Tujuan penggunaan algoritma elias omega agar file gambar yang disimpan menjadi file gambar yang berukuran kecil. Hasil yang didapat dari membangun aplikasi kompresi file gambar yaitu ukuran file gambar menjadi lebih kecil, dengan compression ratio sebesar 66,5%.

**Kata Kunci:** Aplikasi; Kompresi; Kompresi File; Elias Omega Code; File Gambar

**Abstract**—The habit of collecting data makes the capacity of large storage space smaller, making it difficult for computer users to store large amounts of data or large data. One of the files that are large and usually collected by humans is image files, this takes up a lot of storage space. One solution to overcome this problem is to use a compression algorithm such as elias omega code so that the size of large image files becomes smaller. Elias Omega Code Algorithm is a compression algorithm with a lossless compression type, where the compression results can be returned as before without any bit loss. The purpose of using the elias omega algorithm is so that the image file that is saved becomes a small image file. The results obtained from building an image file compression application are that the image file size becomes smaller, with a compression ratio of 66.5%.

**Keywords:** Application; Compression; File Compression; Elias Omega Code; Image Files

## 1. PENDAHULUAN

Penggunaan teknologi saat ini sudah menjadi suatu hal yang biasa dalam kehidupan sehari-hari, karena penggunaan teknologi sangat memudahkan untuk saling bertukar informasi. Hal ini berpengaruh terhadap media penyimpanan, karena terbatasnya media penyimpanan (storage) mengakibatkan data yang bisa disimpan menjadi sedikit. Selain masalah kapasitas media penyimpanan, ukuran file yang besar akan membuat pengiriman data menjadi semakin lama. Sehingga diperlukan suatu teknik kompresi data yang dapat merubah ukuran suatu data menjadi lebih kecil.

Terdapat beberapa aplikasi kompresi yang tersedia memiliki hasil kinerja yang berbeda-beda, ada yang hasilnya maksimal yaitu dapat mengubah ukuran data berkisar 50% lebih kecil dari data aslinya dan ada pula aplikasi kompresi yang hasilnya lebih besar dari ukuran data aslinya. Ukuran file gambar atau file citra yang relatif besar dimana biasanya semakin baik kualitas suatu gambar akan berpengaruh terhadap ukurannya sehingga ukuran file gambar tersebut menjadi lebih besar. Salah satu dari format file gambar yang cukup populer dan yang sering digunakan untuk penyimpanan file citra yang format file BMP yang dikembangkan oleh Microsoft. Berbeda dengan format JPG yang telah dikompresi, format BMP merupakan format yang tidak terkompresi dan ukurannya lebih besar.

Ada sejumlah algoritma yang dikenal untuk kompresi data berdasarkan beberapa ide yang cocok untuk tipe data yang berbeda dan menghasilkan hasil yang berbeda, tetapi semuanya didasarkan pada prinsip kompresi data yang sama dengan cara menghilangkan redundansi dari file data sumber, beberapa termasuk algoritma Elias Omega Codes, Run length Encoding, Rice Code, Levenstein Code, Goldbach Code dan lain-lain.

Hasil penelitian terdahulu yang dilakukan oleh Iwan Fitrianto Rahmad dan Helmi Kurniawan disimpulkan bahwa untuk mengompresi suatu file gambar dapat dilakukan dengan cara menyederhanakan isi file dengan karakteristik yang sama, sehingga selama penyimpanan ukuran file dapat diperkecil. Kesimpulannya, semakin banyak konten file yang memiliki karakteristik yang sama, semakin banyak file yang dapat dikompresi hingga 50 dengan ukuran aslinya [1]. Penelitian lainnya juga dilakukan oleh Cut Try Utary dan kesimpulan dari penelitian tersebut yaitu aplikasi kompresi file gambar telah dirancang dapat menghemat ruang penyimpanan, karena aplikasi tersebut dapat mengubah ukuran file gambar awal yang besar sehingga dapat diperkecil [2]. Penelitian lainnya tentang kompresi file gambar juga dilakukan oleh esterina novebri simanjuntak dan hasil dari penelitian tersebut yaitu file gambar berhasil dikompresi menggunakan algoritma prefix code dengan rasio kompresi sebesar 37,5% [3]. Dalam penelitian yang dilakukan oleh agus adi pramadi, dkk disimpulkan bahwa ukuran file gambar dapat berkurang sehingga dapat menghemat ruang alokasi penyimpanan, hasil kompresi sangat baik sehingga dalam melakukan transmisi data dan pemindahan alokasi file gambar lebih mudah dan cepat [4]. Penelitian yang dilakukan

oleh amir maniar silitonga tentang kompresi file gambar, dimana hasil penelitian tersebut yaitu dengan menggunakan algoritma Subexponential Code dapat merubah file gambar yang semulanya memiliki ukuran besar menjadi ukuran yang lebih kecil, dan akan memberikan manfaat ruang memori yang lebih sedikit [5]. Sedangkan penelitian yang dilakukan oleh elvia hariska tentang mengkompresi file gambar menggunakan additive code dimana menghasilkan ratio of compression (rc) sebesar 1,739 , compression ratio (cr) sebesar 57,5 % , dan redundancy sebesar 42,5 % [6].

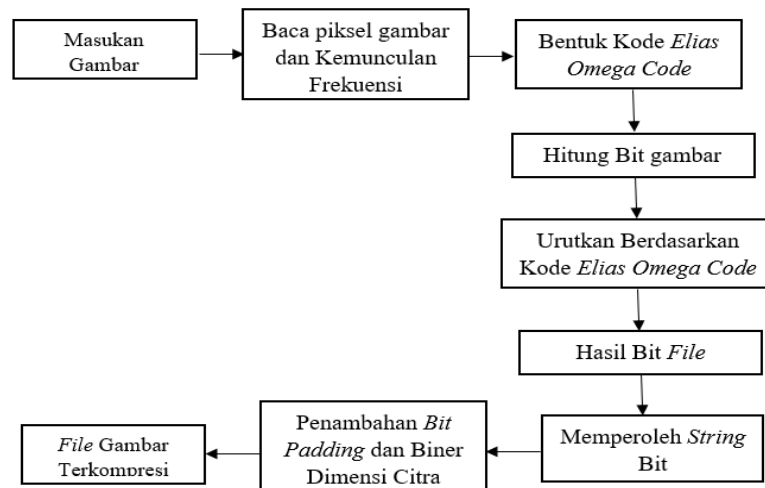
Algoritma yang digunakan dalam penelitian ini adalah algoritma Elias Omega Codes. Algoritma Elias Omega Code, di mana sebagian besar karakter disimpan dalam bit terkecil yang sesuai dengan code\_so\_far ken, digunakan secara rekursif untuk mengkodekan awalan M, sehingga kadang-kadang disebut kode Elias rekursif [7]. Penelitian tentang algoritma elias omega code juga pernah dilakukan oleh sri agustiani, dkk dimana dari hasil penelitian tersebut dapat disimpulkan bahwa hasil rasio kompresi berkisar 53-62% [8]. Penelitian yang dilakukan oleh nur aisyah dan soeb aripin tentang kompresi file audio menggunakan algoritma elias omega code menyimpulkan bahwa hasil dari kompresi file audio tersebut memiliki rasio kompresi sebesar 60% [9]. Penelitian yang dilakukan oleh ibrahim hasan, dkk tentang menganalisis parameter kompresi pada algoritma elias omega code dan fibonacci code, disimpulkan bahwa dari proses kompresi pada string karakter file digital, algoritma Fibonacci Code lebih unggul dalam hal kompresi ratio of compression dengan rata-rata 1,72, compression ratio dengan rata-rata 57% dan space saving dengan rata-rata 42% [10]. Sedangkan penelitian yang dilakukan oleh febika rada kesuma tentang kombinasi algoritma elias omega code dan algoritma golomb rice menyimpulkan bahwa space saving hingga mencapai 55,5 % sedangkan rasio kompresi sebanyak 44,4% [11].

Dalam penelitian ini akan dirancang aplikasi kompresi file gambar menggunakan perangkat lunak Microsoft Visual Studio 2008, yang merupakan sebuah software untuk membuat aplikasi windows atau disebut juga aplikasi berbasis desktop. Aplikasi yang akan dirancang dalam penelitian ini menggunakan algoritma elias omega codes untuk mengompresi file gambar, yaitu aplikasi yang bertujuan dapat memperkecil ukuran suatu file gambar dan meminimalisir pemakaian memori penyimpanan.

## 2. METODOLOGI PENELITIAN

### 2.1 Kompresi

Kompresi adalah konversi data dalam bentuk set karakter ke bentuk yang dikodekan dengan tujuan menghemat kebutuhan penyimpanan dan waktu transmisi data. [12][13]. Kompresi data penting karena mengurangi kebutuhan penyimpanan data, meningkatkan kecepatan transfer data, dan mengurangi kebutuhan bandwidth. Ada banyak algoritma kompresi data, algoritma kompresi ini dibuat dari ide yang berbeda dan cocok untuk berbagai jenis atau tipe data dan menghasilkan output yang berbeda tergantung pada algoritma yang digunakan. Namun, prinsip dasar di balik setiap algoritma kompresi adalah sama, yaitu mengompresi data dengan menghilangkan redundansi atau redundansi dari data asli [14]. Dalam penelitian ini tahapan yang dilakukan untuk mengkompresi file gambar bisa dilihat pada gambar 1.



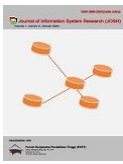
**Gambar 1.** Tahapan Mengkompresi File Gambar

Pada proses kompresi data terdapat beberapa variabel yang bisa digunakan untuk menganalisa kualitas atau kinerja dari suatu algoritma kompresi data yaitu :

1. Ratio of Compression ( $R_C$ )

Ratio of Compression ( $R_C$ ) merupakan nilai perbandingan antara ukuran bit data sebelum dilakukan proses kompresi dengan ukuran bit data setelah dilakukan proses kompresi.

Rumus :



Rc = Jumlah bit sebelum dikompresi / Jumlah bit sesudah dikompresi ..... (1)

2. Compression Ratio (CR)

Compression Ratio(CR) merupakan persentase perbandingan antara data yang sudah dikompresi dengan data sebelum proses kompresi.

Rumus :

Cr = Jumlah bit sesudah dikompresi / Jumlah bit sebelum dikompresi x100% ..... (2)

3. Space Savings (SS)

Space Saving (SS) merupakan selisih antara data sebelum proses kompresi dengan besar data yang sudah dikompresi.

Rumus :

Ss = 100 % - CR ..... (3)

4. Time Compression

Time compression merupakan waktu yang dibutuhkan untuk melakukan proses untuk mengkompresi dan proses untuk mendekompresi data. Semakin cepat waktu yang dibutuhkan untuk proses kompresi maka semakin efisien algoritma kompresi yang digunakan.

2.2 Elias Omega Code

Algoritma Elias Omega Code adalah algoritma kompresi yang diperkenalkan oleh Peter Elias pada tahun 1975. Peter Elias menjelaskan tiga kode awalan yang berguna. Ide utama dari kode ini adalah bahwa awalan bilangan bulat dikodekan menggunakan representasi kriptografi tingkat atas. Penentuan panjang M diselesaikan dengan cara yang berbeda dengan kode Elias Omega yang berbeda. Ada tiga kode Elias, yaitu Kode Elias Gamma, Kode Elias Delta dan Kode Elias Omega. Algoritma Elias Omega, tidak seperti algoritma Elias lainnya, digunakan secara rekursif untuk mengkodekan awalan M, itulah sebabnya mengapa kadang-kadang disebut cipher Elias Recursive. [3].

Pengkodean Encoding pada elias omega code menggunakan bilangan bulat positif n dan dilakukan secara rekursif dalam langkah-langkah berikut :

- 1. Inisialisasikan kode dengan = 0.
2. Jika n = 1, maka stop (proses berhenti) jika tidak, maka tambahkan representasi biner dari n ke kode yang ada dan asumsikan bahwa kita telah menambahkan bit sejumlah L (panjang digit bit).
3. Ulangi langkah ke 2 sampai representasi biner dari L - 1 untuk menggunakan n.

Misalnya jika n = 17 dikodekan dengan cara (1) berikan angka 0, (2) tambahkan dengan representasi biner dari 17, yaitu 10001 (terdiri dari 5 bit), (3) tambahkan dengan nilai 3 bit dari 5-1 = 1002, dan (4) tambahkan nilai 2 bit dari 3-1 = 102. Maka, hasilnya adalah 10|100|10001|0.

Tabel 1. Kode Elias Gamma

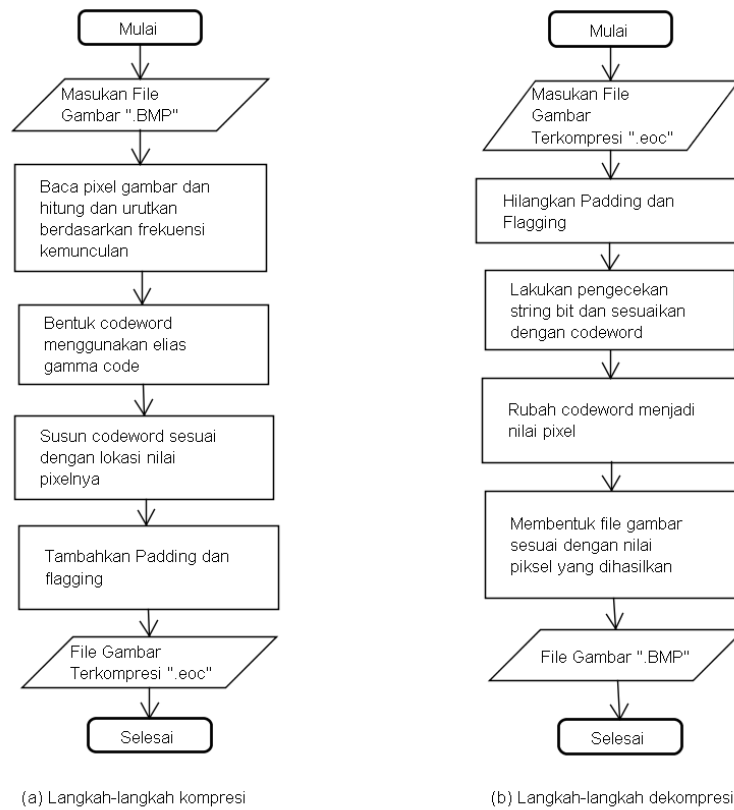
Table with 2 columns: Nilai n and Kode Elias Omega. It lists binary representations for n from 1 to 18.

Sumber : David Solomon, Giovanni Motta, 2010 [3].

3. HASIL DAN PEMBAHASAN

3.1 Analisis Proses Kompresi File Gambar

Dalam mengkompresi file gambar, algoritma yang digunakan adalah algoritma elias omega code. Langkah-langkah yang terdapat pada aplikasi kompresi file gambar yaitu proses mengkompresi file gambar dan proses dekompresi terhadap file gambar yang telah terkompresi. Adapun langkah-langkah kompresi dan dekompresi pada file gambar berformat bmp dapat dilihat pada gambar 2.



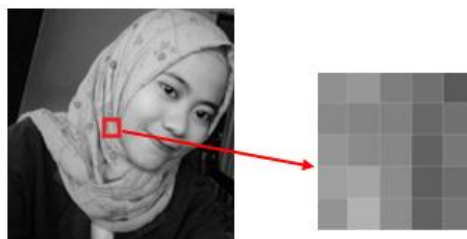
**Gambar 2.** Langkah-langkah kompresi dan dekompresi

Untuk mengkompresi file gambar terlebih dahulu memilih gambar yang akan dikompresi. Sebagai contoh, file gambar yang akan digunakan untuk proses kompresi adalah file gambar berjenis grayscale yang memiliki ukuran 668 KB dengan resolusi 477 x 478 pixels. Tampilan citra grayscale yang digunakan dapat dilihat pada gambar 3.



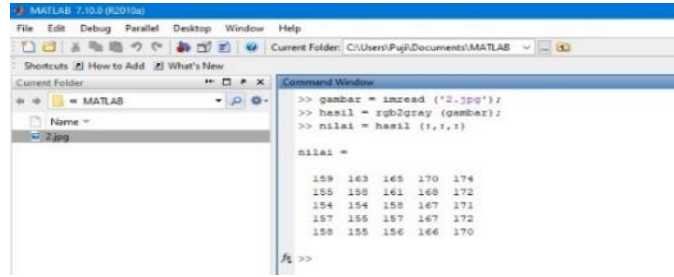
**Gambar 3.** Citra grayscale

Karena ukuran file citra grayscale 477 x 478 pixel, dengan jumlah pixel keseluruhan adalah 228.006 maka perlu diambil sample dari file gambar grayscale dengan resolusi sample 5x5 pixel agar analisa tidak terlalu panjang. Kemudian file gambar yang berukuran 5x5 pixel akan dikompresi menggunakan algoritma Elias Omega. Pengambilan sample dapat dilihat pada gambar 4.



**Gambar 4.** Sample grayscale 5 x 5 pixel

Langkah selanjutnya adalah membaca isi dari file gambar tersebut. Setiap file gambar berisi nilai piksel. Nilai piksel yang akan diolah adalah 5 x 5 sehingga menghasilkan 25 piksel sebagai sampel citra yang akan dianalisis. Nilai 25 piksel diperoleh dengan menggunakan aplikasi MATLABR2010a yang gambarnya dapat dilihat pada Gambar 5.



**Gambar 5.** Nilai 25 Pixel Sampel

Dalam penerapan algoritma Elias Omega Codes terdapat dua proses penyelesaian yaitu proses kompresi dan proses dekompresi.

1. Analisa proses kompresi file gambar dengan menggunakan Elias Omega Codes.

Hasil nilai diatas telah dibuat ke dalam bentuk matriks. Matriks nilai pixels dapat dilihat pada Gambar 6.

$$\begin{pmatrix} 159 & 163 & 165 & 170 & 174 \\ 155 & 158 & 161 & 168 & 172 \\ 154 & 154 & 158 & 167 & 171 \\ 157 & 155 & 157 & 167 & 172 \\ 158 & 155 & 156 & 166 & 170 \end{pmatrix}$$

**Gambar 6.** Matriks Nilai Pixel

Berdasarkan pada gambar 5 di atas maka didapatkan nilai pixel karakter sebagai berikut, 159, 163, 165, 170, 174, 155, 158, 161, 168, 172, 154, 154, 158, 167, 171, 157, 155, 157, 167, 172, 158, 155, 156, 166, 170. Nilai pixel ini dimasukan kedalam tabel untuk dilakukan pembacaan frekuensi. Pembacaan frekuensi dilakukan dengan menghitung jumlah nilai yang sama di setiap nilai pixel yang muncul serta akan dilakukan pengurutan dari frekuensi besar ke frekuensi yang kecil. Adapun pembacaan frekuensi dan pengurutannya dapat dilihat pada tabel 2.

**Tabel 2.** Pengurutan Pixel yang Belum Dikompresi

Nilai Pixel	Binary	Bit	Frek	Bit x Frek
155	10011011	8	3	24
158	10011110	8	3	24
170	10101010	8	2	16
172	10101100	8	2	16
154	10011010	8	2	16
167	10100111	8	2	16
157	10011101	8	2	16
159	10011111	8	1	8
163	10100011	8	1	8
165	10100101	8	1	8
174	10101110	8	1	8
161	10100001	8	1	8
168	10101000	8	1	8
171	10101011	8	1	8
156	10011100	8	1	8
166	10100110	8	1	8
Total Bit				200 bit

Berdasarkan tabel 2, nilai desimal (karakter) sama dengan 8 bit bilangan biner. Jadi, 25 bilangan desimal memiliki nilai biner 200 bit. Untuk mengkonversi dari satuan ke byte, jumlah bit dibagi 8, jadi  $200/8 = 25$  byte. Langkah selanjutnya adalah membangkitkan elias encoding dari kode omega. Aturan pengkodean menggunakan kode omega elias dapat ditemukan di landasan teori bab sebelumnya. Adapun kode elias omega code dapat dilihat pada tabel 3 di bawah ini:

**Tabel 3.** Kode Elias Omega Code

n	Kode Elias Omega	n	Kode Elias Omega
1	0	10	11 1010 0
2	10 0	11	11 1011 0
3	11 0	12	11 1100 0

n	Kode Elias Omega	n	Kode Elias Omega
4	10 100 0	13	11 1101 0
5	10 101 0	14	11 1110 0
6	10 110 0	15	11 1111 0
7	10 111 0	16	10 100 10000 0
8	11 1000 0	17	10 100 10001 0
9	11 1001 0	18	10 100 10010 0

Berdasarkan pengkodean Omega Elias di atas, hanya menampilkan kode sampai dengan nilai karakter 18. Bagaimana proses untuk mendapatkan kode Omega Elias, jika jumlah karakter yang dihasilkan lebih dari 18 karakter. Misalnya nilai = 19. Nilai desimal 19 dikodekan dengan (1) memberikan angka 0, (2) menambahkannya ke representasi biner 19, yaitu 10011 (2) (5 bit), (3) menambahkan Nilai 3-bit 51 = 1002 dan (4) tambahkan nilai 2-bit 31 = 102. Hasilnya adalah 10 | 100 | 10011 | 0. Proses selanjutnya adalah mengompresi nilai piksel sampel dengan nilai kode omega elias yang diperoleh dari Tabel 3 di atas. Contoh kompresi gambar dapat dilihat pada tabel di bawah ini :

**Tabel 4.** Pixel yang Sudah Dikompresi Dengan Elias Omega Code

n	Nilai Des Pixel	Elias Omega Code	Bit	Frek	Bit x Frek
1	155	0	1	3	3
2	158	10 0	3	3	9
3	170	11 0	3	2	6
4	172	10 100 0	6	2	12
5	154	10 101 0	6	2	12
6	167	10 110 0	6	2	12
7	157	10 111 0	6	2	12
8	159	11 1000 0	7	1	7
9	163	11 1001 0	7	1	7
10	165	11 1010 0	7	1	7
11	174	11 1011 0	7	1	7
12	161	11 1100 0	7	1	7
13	168	11 1101 0	7	1	7
14	171	11 1110 0	7	1	7
15	156	11 1111 0	7	1	7
16	166	10 100 10000 0	11	1	11
Total Bit					133 bit

Dari perhitungan tabel diatas setelah dikompresi dengan menggunakan elias omega code adalah 133 bit. Untuk diubah menjadi satuan bytes maka dibagi 8 yaitu  $133/8 = 16,6$  bytes. Dapat dibentuk nilai bit baru hasil kompresi dari susunan nilai desimal pixel citra sampel awal sebelum kompresi yaitu, 159, 163, 165, 170, 174, 155, 158, 161, 168, 172, 154, 154, 158, 167, 171, 157, 155, 157, 167, 172, 158, 155, 156, 166, 170 (tanpa tanda koma dan spasi) menjadi nilai bit biner “1110000111001011101 001101110110010011110001111010101000101010 1010101001011001111001011100101101011001010001000111111010100100000110”.

Kemudian, sebelum mendapatkan hasil keseluruhan dari kompresi akhir, ditambahkan urutan bit itu sendiri, yaitu bit padding dan bit flag. Jika sisa panjang string 8 bit adalah 0, tambahkan 00000001. Nyatakan dengan bit terakhir. Sedangkan jika sisa panjang bitstring di 8 adalah  $n(1,2,3,4,5,6,7)$  maka tambahkan 0 hingga  $7 - n + "1"$  di akhir bitstring. Dinyatakan ke L Kemudian tambahkan bilangan biner dari 9 ke n. dideklarasikan dengan bit terakhir. Karena jumlah bit string adalah 133 yang tidak habis dibagi delapan dan sisanya adalah 5 bit, nyatakan sisanya sebagai n. kemudian tambahkan dari 0 ke 0 hingga  $7 - n + "1"$  ke akhir string bit. Dinyatakan ke L Kemudian tambahkan bilangan biner dari 9 ke n. Deklarasikan dengan bit terakhir.

$$7 - n + "1"$$

$$7 - 5 + "1" = 001$$

Bit Akhir 9 - n

$$\text{Bit Akhir} = 9 - 5 = 4 = 00000100$$

**Gambar 7.** Perhitungan penambahan bit

```
111000011100101110100110111011001001111000111101010100
010101010101010010110011111001011100101110101100101000
100011111101010010000011000100000100
```

**Gambar 8.** String Bit yang telah dilakukan penambahan



Panjang total dalam bit setelah menambahkan bit adalah  $133+3+8 = 144$  bit. Kemudian dilanjutkan dengan memisahkan bit-bit tersebut menjadi beberapa kelompok. Setiap grup terdiri dari 8 bit seperti gambar 8 berikut:

11100001	11001011	10100110	11101100	10011110
00111101	01010001	01010101	01010010	11001111
10010111	00101110	10110010	10001000	11111101
01001000	00110001	<b>00000100</b>		

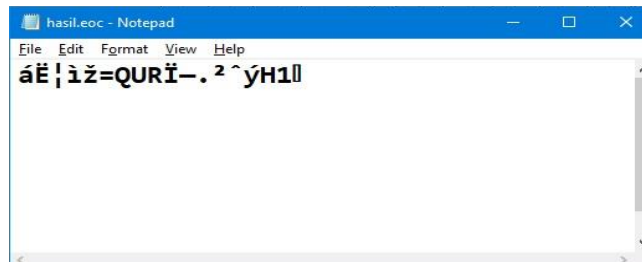
**Gambar 9.** Pembagian String Bit

Berdasarkan pembagian kelompok nilai biner, 18 nilai biner baru telah dikompresi dengan nilai biner tambahan bitwise. Setelah pembagian selesai, piksel split akan diubah menjadi karakter dengan terlebih dahulu mencari nilai desimal dari bit string menggunakan kode ASCII untuk mengetahui nilai piksel yang dikompresi. Nilai piksel yang dikompresi dapat dilihat pada tabel di bawah ini:

**Tabel 5.** Nilai Desimal Pixel Terkompresi

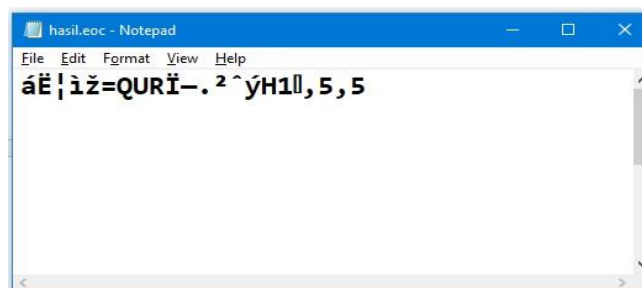
Urutan Pixel Terkompresi	Nilai Desimal Terkompresi	Urutan Pixel Terkompresi	Nilai Desimal Terkompresi
1	225	10	207
2	203	11	151
3	166	12	46
4	236	13	178
5	158	14	136
6	61	15	253
7	81	16	72
8	85	17	49
9	82	18	4

Setelah mengetahui nilai desimal, maka akan mengubah nilai desimal menjadi karakter. Karakter yang diperoleh dari hasil kompresi disimpan dalam file dengan ekstensi ".eoc", dan jika file dibuka dengan aplikasi notepad, karakter akan muncul seperti gambar di bawah ini:



**Gambar 10.** Hasil Karakter Kompresi

Jadi selama dekompresi dapat dilihat bahwa ukuran gambar asli adalah 5x5 piksel, string ",5,5" akan ditambahkan ke karakter yang dikompresi. Karakter koma pertama (,) digunakan untuk memisahkan karakter terkompresi dari karakter dimensi. Karakter pertama "5" digunakan untuk menentukan nilai sumbu X pada gambar asli. Karakter koma kedua (,) digunakan untuk memisahkan nilai sumbu X dari nilai sumbu Y pada gambar asli. Karakter kedua "5" digunakan untuk menentukan nilai dari sumbu y pada gambar asli. Jadi hasil kompresi gambar setelah menambahkan string ukuran gambar ".5.5" (tanpa tanda kutip) adalah sebagai berikut:



**Gambar 11.** Hasil Kompresi Dengan Penambahan Informasi Dimensi Citra Awal

Dari hasil kompresi tersebut dapat diukur kinerja dari algoritma Elias Omega Codes sebagai berikut :

- 1) Ratio of Compression ( $R_C$ )

$$R_C = \frac{200}{133} = 1,5037$$

2) Compression Ratio ( $C_R$ )

$$R_C = \frac{133}{200} \times 100\% = 66,5 \%$$

3) Space Savings ( $S_S$ )

$$S_S = 100\% - C_R$$

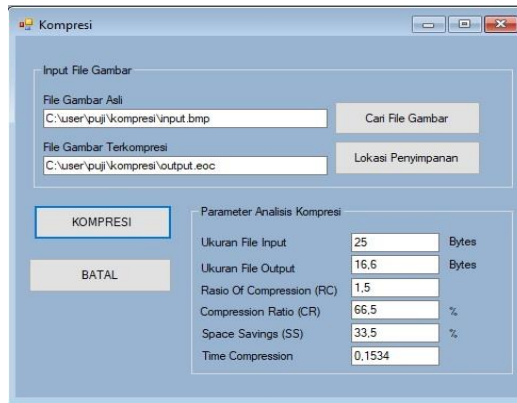
$$S_S = 100\% - 66,5\% = 33,5\%$$

**3.2 Pengujian**

Hasil pengujian dari perangkat lunak yang dibangun yaitu :

1) Proses Kompresi

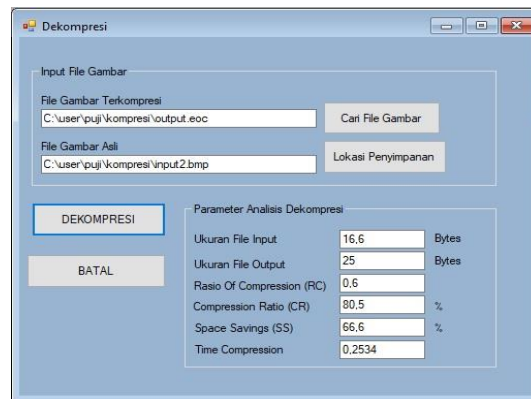
Selama proses kompresi, hal pertama yang harus dilakukan adalah memilih file gambar yang akan dikompres dan memilih tempat penyimpanan untuk dikompres. Selanjutnya aplikasi akan melakukan proses kompresi. Tampilan form kompresi dapat dilihat pada gambar 12.



**Gambar 12.** Proses Kompresi

2) Proses Dekompresi

Selama proses dekomposisi, hal pertama yang harus dilakukan adalah memilih file gambar yang akan didekompresi dan memilih tempat menyimpannya untuk dikompres atau didekompresi. Selanjutnya adalah kompresi file. Tampilan form dekomposisi dapat dilihat pada Gambar 13.



**Gambar 13.** Proses Dekompresi

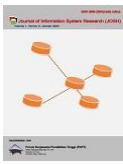
**4. KESIMPULAN**

Dari perancangan aplikasi kompresi file gambar dapat disimpulkan bahwa aplikasi kompresi file gambar yang dibangun dapat berjalan dengan baik untuk mengkompresi file gambar dengan compression ratio mencapai 66,5 %, dan karena algoritma yang digunakan adalah jenis lossless compression, maka file gambar hasil kompresi tidak dapat ditampilkan gambarnya dan jika ingin menampilkan gambarnya maka harus dilakukan proses dekomposisi. Dalam pengembangan penelitian selanjutnya bisa mencoba untuk mengkompresi file lainnya seperti file video, dokumen ataupun file audio.

**REFERENCES**

[1] I. Fitrianto and H. Kurnaiwan, “Kompresi file citra bitmap menggunakan algoritma rle dan lz78,” *CSRID J.*, vol. 3, no. 2, pp. 81–92, 2011.  
 [2] C. T. Utari, P. Studi, M. Teknik, U. S. Utara, and K. Citra, “IMPLEMENTASI ALGORITMA RUN LENGTH





- ENCODING UNTUK PERANCANGAN APLIKASI KOMPRESI DAN DEKOMPRESI,” vol. V, no. 2, pp. 24–31, 2016.
- [3] E. N. Simanjuntak, “Penerapan Algoritma Prefix Code Pada Kompresi File Gambar,” *BEES Bull. Electr. Electron. Eng.*, vol. 1, no. 3, pp. 96–100, 2021.
- [4] A. A. Pramadi, S. D. Nasution, B. Purba, A. Event, and R. Code, “PENERAPAN ALGORITMA EVEN-RODEH PADA APLIKASI KOMPRESI FILE,” *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, pp. 73–84, 2019, doi: 10.30865/komik.v3i1.1570.
- [5] A. M. Silitonga, S. D. Nasution, and P. Ramadhani, “IMPLEMENTASI ALGORITMA SUBEXPONENTIAL CODE UNTUK KOMPRESI FILE,” *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, pp. 340–348, 2019, doi: 10.30865/komik.v3i1.1611.
- [6] E. Hariska, “Perancangan Aplikasi Kompresi File Gambar Menggunakan Algoritma Additive Code,” vol. 5, pp. 193–202, 2021, doi: 10.30865/komik.v5i1.3671.
- [7] D. Salomon and G. Motta, *HandBook Of Data Compression*. Springer, 2010.
- [8] S. A. B. Siburian, R. K. Hondro, and B. Purba, “Implementasi Algoritma Elias Omega Coding Untuk Kompresi File Teks Pada Aplikasi Media Pembelajaran Buku Sekolah Digital Untuk Pelajar,” ... *Teknol. Inf. dan ...*, vol. 4, no. May, pp. 178–185, 2020, doi: 10.30865/komik.v4i1.2677.
- [9] N. Aisyah and S. Aripin, “Penerapan Algoritma Elias Omega Code Pada Kompresi File Audio Aplikasi Murottal Muzzamil Hasbalah,” *Pelita Inform.*, vol. 9, pp. 113–119, 2020.
- [10] I. Hasan, Tommy, and N. Irsa Syahputri, “Analisis Parameter Kompresi Algoritma Elias Omega Code dan Fibonacci Code Pada File Digital,” *Algorith. J. Ilmu Komput. dan Inform.*, vol. 6341, no. April, p. 1, 2021.
- [11] F. R. Kesuma, “Implementasi Kombinasi Algoritma Elias Omega Codes Dan Golomb Rice Dengan Teknik High Compress Pada File Teks,” *Inf. dan Teknol. Ilm.*, vol. 8, no. 2, pp. 76–81, 2021.
- [12] A. Jalaluddin and Y. Melita, “Implementasi Metode Run Length Encoding (RLE) untuk Kompresi Citra,” *J. Tek.*, vol. 3, no. 2, pp. 249–254, 2012, [Online]. Available: [http://journal.unisla.ac.id/pdf/11322011/Jilid 4 Jalal.pdf](http://journal.unisla.ac.id/pdf/11322011/Jilid%204%20Jalal.pdf).
- [13] S. D. Nasution, “Perancangan Aplikasi Kompresi File Teks Dengan Menerapkan Algoritma Goldbach Codes,” *Jurnal Infotek STIEKOM*, vol. 1, no. Februari, pp. 2008–2010, 2016.
- [14] D. Salomon, *Data Compression The Complete Reference Fourth Edition*, vol. 53, no. 9. Springer, 2007.
- [15] D. Salomon, *Handbook of Data Compression*. London: Springers, 2007.