

# Implementasi Algoritma Elias Gamma Code Untuk Kompresi File Teks Pada Aplikasi Watpad

Elfrida Siahaan

<sup>1</sup>Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia  
Email: [elfridasiahaan3@gmail.com](mailto:elfridasiahaan3@gmail.com)

**Abstrak**– Dalam era teknologi yang semakin canggih seperti saat ini, Kebutuhan terhadap kapasitas penyimpanan yang semakin besar merupakan penyebab munculnya berbagai teknik kompresi. Semakin banyak data yang tersimpan maka semakin besar pula kapasitas yang dibutuhkan. Ukuran data yang besar mengakibatkan pemborosan memori dan lambatnya proses pemindahan data. Penyimpanan data yang semakin lama akan mengakibatkan menumpuknya data – data yang semakin banyak pula. Seiring perkembangan teknologi saat ini maka semakin berkembangnya kemampuan dalam mengolah data yaitu dengan memanfaatkan metode kompresi yang diharapkan dapat menjawab masalah tersebut. Berdasarkan permasalahan tersebut maka perlu untuk meminimalisir penyimpanan pada *database* dengan menggunakan algoritma *Elias Gamma Code* digunakan untuk mengkompresi *record database* pada aplikasi Wattpad agar memperkecil ukuran *bit* dari suatu karakter. Sehingga dengan menerapkan algoritma *Elias Gamma Code* untuk mengkompresi *record database* dapat mengurangi kapasitas penyimpanan secara optimal pada *smartphone (handphone)*.

**Kata Kunci:** Kompresi, *Record Database*, *Unified Modelling Language*, Wattpad, *Elias Gamma Code*

**Abstract**– In this era of increasingly sophisticated technology, the need for greater storage capacity is the cause of the emergence of various compression techniques. The more data stored, the greater the capacity required. Large data sizes result in wasteful memory and the slow process of transferring data. Data storage that takes longer will result in the accumulation of more and more data. Along with current technological developments, the ability to process data is increasingly developing, namely by utilizing compression methods which are expected to answer this problem. Based on these problems it is necessary to minimize storage in the database using the *Elias Gamma Code* algorithm used to compress database records on the Wattpad application in order to minimize the bit size of a character. So that by applying the *Elias Gamma Code* algorithm to compress the database record can reduce the optimal storage capacity on smartphones (cellphones).

**Keywords:** *Record Database*, *Wattpad*, *Unified Modeling Language*, *Compression*, *Elias Gamma Code*.

## 1. PENDAHULUAN

Kompresi data adalah proses mengubah aliran data masukan menjadi aliran data baru yang lebih kecil. Aliran bermasalah adalah file atau buffer di memori. Ada banyak metode kompresi data. Kompresi lossy dan lossless adalah pengelompokan metode kompresi berdasarkan integritas data. Kompresi teks dapat dilakukan dengan mengompresi konten (konten berukuran besar membuat ukurannya lebih kecil tanpa kehilangan keaslian datanya) [1].

Berdasarkan pengertian secara umum, Wattpad merupakan salah satu platform aplikasi dan situs komunitas online untuk para penulis dan pembaca. Artinya, melalui Wattpad ini, pengguna dapat menulis atau membaca konten informasi dalam lingkup cerita sastra cyber (dunia sastra). Konten informasi di Wattpad mempunyai banyak genre kategori seperti cerita pendek, klasik, *faction*, *adventure*, anime, *romance*, *fantasy*, humor, *spiritual*, misteri, horor, puisi, fiksi ilmiah, fiksi sejarah, fiksi remaja, fiksi umum, fiksi penggemar, *chikLit*, klasik dan non fiksi [2].

Permasalahan yang timbul adalah kapasitas penyimpanan (*storage*). Penyimpanan selalu penuh apabila banyak mengunduh aplikasi, video, audio, teks, dan juga file-file lain nya di dalam *smartphone*. Di dalam wattpad terdapat gambar dan teks atau Cerita-cerita yang tersimpan dalam sebuah *database*, sehingga pada saat pengguna mengakses informasi akan muncul sesuai dengan yang diinginkan. Semakin banyak data atau file yang tersimpan di *smartphone* maka semakin besar kapasitas penyimpanan yang dibutuhkan. Hal ini bisa menyebabkan pemborosan media penyimpanan pada *smartphone* karena sebagian *smartphone* memiliki kapasitas penyimpanan yang kecil, meskipun saat ini telah tersedia kapasitas *smartphone* yang besar.

Salah satu teknik sederhana yang saat ini dibutuhkan untuk masalah diatas adalah kompresi. Kompresi adalah proses mereduksi suatu data dengan meminimalkan ukuran dari data tersebut dengan tujuan memampatkan ukuran data tanpa menghilangkan keaslian dari data tersebut. Penyimpanan file teks yang memiliki ukuran lebih besar dapat dikompres menjadi ukuran yang lebih kecil dan mengurangi redundansi dari data-data yang terdapat dalam suatu file sehingga dapat disimpan atau ditransmisikan secara efisien. Salah satu metode yang digunakan dalam melakukan kompresi atau pemampatan data yaitu algoritma *Elias Gamma Code* untuk mengompresi file teks tersebut. Diharapkan dengan memperkecil ukuran dari file yang ada dapat mengefisiensikan penyimpanan yang ada.

Berdasarkan penelitian terdahulu yang dilakukan oleh Sukiman dan Tintin Chandra dengan judul “Aplikasi Kompresi File dengan Algoritma *Elias Gamma*” melakukan penelitian terhadap algoritma *Elias Gamma Code*. Menyimpulkan bahwa *Elias Gamma Code* mempunyai rentang 60% – 80%. Pengujian baik untuk proses kompresi dan dekompresi menunjukkan kecepatan proses rata-rata cukup cepat karena *Elias Gamma Code* melakukan proses *encoding* dalam proses kompresinya. Waktu yang dibutuhkan untuk kompresi tidak bergantung pada besar ukuran file dan begitu pula waktu dekompresi tidak tergantung pada waktu kompresi. Secara rata-rata, kompresi lebih membutuhkan banyak waktu dari pada dekompresi. Besarnya rasio kompresi tergantung kepada jumlah bit per sample dan sample rate. Proses

kompresi dengan menggunakan *Elias Gamma Code* mempunyai keterbatasan karena hanya dapat merepresentasikan 7 karakter pertama dengan jumlah bit di bawah 8 bit [3].

Algoritma *Elias Gamma* memiliki proses kompresi dan dekompresi cukup cepat karena algoritma elias gamma melakukan proses encoding pada saat proses kompresi berlangsung. Waktu kompresi tidak bergantung pada besarnya ukuran file dan waktu dekompresi tidak tergantung pada waktu kompresi. Waktu kompresi rata-rata membutuhkan waktu lebih dari pada waktu dekompresi *Elias* menambah panjang dalam unary ( $u$ ) untuk membentuk tabel kode elias gamma. Dalam kode berikutnya,  $E\gamma$  ditambahkan pada panjang kode ( $M$ ) dalam biner ( $\beta$ ). Dengan demikian elias gamma code, yang juga untuk bilangan bulat positif, sedikit lebih kompleks untuk dibangun[4]

## 2. METODOLOGI PENELITIAN

### 2.1 Kompresi

Kompresi data dalam bidang ilmu komputer, ilmu pengetahuan dan seni adalah sebuah penyajian informasi ke dalam bentuk yang lebih sederhana. Kompresi data atau source coding dalam ilmu komputer dan teori informasi adalah proses mengencode informasi dengan menggunakan lebih sedikit bit dari suatu sumber yang belum di-encode melalui penggunaan skema pengkodean yang spesifik. Kompresi data dapat diartikan juga sebagai proses yang dapat mengubah sebuah aliran data masukan (sumber atau data asli) ke dalam aliran data yang lain (keluaran atau data yang dimampatkan) yang memiliki ukuran yang lebih kecil. Kompresi data sudah ada sejak puluhan tahun yg lalu. Kompresi data menjadi salah satu cabang Teori Informasi karena kompresi data berkecimpung dengan masalah redundancy dalam suatu informasi [5].

### 2.2 Algoritma Elias Gamma Code

Algoritma elias gamma code dibuat oleh Peter Elias. Tabel kode elias gamma, elias dibuat dengan menambah panjang kode dalam Unary ( $u$ ). Kode berikutnya,  $E\gamma$  ditambahkan pada panjang kode ( $M$ ) dalam biner ( $\beta$ ). Dengan demikian, elias gamma code, yang juga untuk bilangan bulat positif, sedikit lebih kompleks untuk dibangun. Aturan untuk mengkodekan sebuah bilangan dengan menggunakan *Elias Gamma* adalah sebagai berikut [7]:

1. Rubah bilangan kode dalam bentuk biner.
2. Kurangkan 1 dari jumlah bit yang dipilih pada langkah pertama dan tambahkan sesuai dengan banyaknya bilangan nol. Proses yang ekuivalen untuk menyatakan proses yang pada point nomor dua adalah sebagai berikut:
  - a. Pisahkan integer menjadi pangkat 2 tertinggi ( $2^N$ ) yang dapat dan ditampungnya sisakan digit biner  $N$  dari integer tersebut.
  - b. Kodekan  $N$  dalam bentuk unary, jika  $N$  adalah nol maka diikuti oleh satu.
  - c. Tambahkan sisa digit biner  $N$  yang telah dihasilkan.

Proses kompresi/encoding suatu integer dengan berdasarkan elias gamma dilakukan dengan cara:

1. Tentukan nilai  $N$  untuk pangkat yang paling mendekati nilai  $n$  yang dituliskan sebagai  $\beta(n)$ . Nilai ini disebut sebagai unary code, dimana jumlah nilai  $N$  ditulis menjadi angka 0 dan diakhiri dengan angka 1.
2. Dapatkan nilai  $L$  dengan mengurangi nilai  $n$  dengan nilai  $2^N$ , nilai yang didapati diubah menjadi bilangan biner.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Analisa Masalah

Analisa dan perancangan perangkat lunak yang dilakukan untuk mengkompresi *file* teks pada aplikasi watsapp menggunakan algoritma elias gamma code. Aplikasi Watsapp merupakan aplikasi yang sangat populer di kalangan remaja saat ini untuk membaca berbagai macam cerita dan juga dapat menerbitkan cerita online.

Banyak cerita dalam aplikasi watsapp berbentuk teks. File teks pada aplikasi watsapp tersimpan di dalam *database*, sehingga semakin banyak file teks yang tersimpan, maka semakin besar kapasitas yang digunakan. Algoritma *Elias Gamma Code* merupakan salah satu teknik *lossless compression* yang memperkecil suatu data berdasarkan dengan frekuensi karakter pada objek yang akan dilakukan proses kompresi. Proses kompresi algoritma *elias gamma code* dilakukan berdasarkan karakter yang sering muncul dan akan memiliki jumlah bit terkecil berdasarkan kode *elias gamma code* sedangkan karakter paling sedikit muncul akan memiliki jumlah bit terpanjang.

Tujuan dari analisa terhadap sistem yang akan dibangun yaitu untuk mengetahui dan merumuskan kebutuhan dari sistem serta membantu meminimalkan ukuran data.

#### 3.1.1 Penerapan *Elias Gamma Code*

Berdasarkan analisa, aplikasi watsapp membutuhkan kapasitas penyimpanan yang besar apabila melakukan instalasi pada *smartphone(handphone)*. Dengan melakukan kompresi file teks *record database*, aplikasi watsapp di instal pada *smartphone* yang memerlukan kapasitas penyimpanan memori yang besar akan dikompresi sehingga mengurangi jumlah penyimpanan memori dan dapat melakukan instalasi di memori yang kecil. Dalam menganalisa aplikasi watsapp harus dilakukan pengambilan *sample* yaitu nama judul cerita watsapp untuk mendapatkan nilai dari karakternya. Berikut adalah *record database* watsapp.

**Tabel 1.** Record Database Judul cerita Wattpad

No	Judul cerita Wattpad
1	ACCIDENTALLY IN LOVE
2	Let Me Love You
3	Me And Mr Perfect
4	Somewhere Only We Know
5	Waiting For Happiness
6	Love Is Not Simple
7	My Destiny
8	Promise
9	My Life My Love My Friend
10	You Are My Destiny

### 1. Kompresi Elias Gamma Code

Sebelum menggunakan algoritma kode Elias Gamma untuk kompresi, terlebih dahulu menentukan *string* yang digunakan sebagai contoh *record database* yang berisi *string* "ACCIDENTALLY IN LOVE". Sebelum dilakukan proses kompresi harus diketahui terlebih dahulu bahwa *string* "ACCIDENTALLY IN LOVE" terdiri atas 20 karakter, dimana setiap 1 karakter tersebut dihitung 1 *byte* dan 1 *byte* itu sama dengan 8 *bit* maka total *string* sebelum dikompresi adalah 20 *byte* atau sama dengan 160 *bit*.

Data *String* = ACCIDENTALLY IN LOVE

Karakter =  $\sum = \{A, C, C, I, D, E, N, T, A, L, L, Y, SP, I, N, SP, L, O, V, E\}$

Frekuensi Karakter =

**Tabel 2.** Frekuensi Data

Karakter	Frekuensi
A	2
C	2
I	2
D	1
E	2
N	2
T	1
L	3
Y	1
SP	2
O	1
V	1
Total	20

Setelah mengetahui karakter dan frekuensinya, maka langkah selanjutnya adalah melakukan pengurutan karakter berdasarkan kemunculan karakter mulai dari yang terbesar ke frekuensi terkecil. Untuk lebih jelasnya lihat tabel 3.3 di bawah ini.

**Tabel 3.** Pembentukan Dikompresi

NO	Karakter	Frekuensi	BINER	BIT	Frekuensi x Bit
1	L	3	01001100	8	24
2	A	2	01000001	8	16
3	C	2	01000011	8	16
4	I	2	01001001	8	16
5	E	2	01000101	8	16
6	N	2	01001110	8	16
7	SP	2	00100000	8	16
8	D	1	01000100	8	8
9	T	1	01010100	8	8
10	Y	1	01011001	8	8
11	O	1	01001111	8	8
12	V	1	01010110	8	8
Total Bit					160

Menurut tabel di atas, nilai 1 karakter adalah bilangan 8 bit. Jadi, nilai dari 20 karakter adalah 160 bit. Untuk mengubah satuan menjadi byte maka jumlah keseluruhan bit dibagi 8. Maka dihasilkan  $160/8 = 20$  byte.

Langkah selanjutnya adalah membentuk tabel kode kode Elias Gamma. Kode elias gamma dapat dilihat pada Tabel 3.4 Dibawah ini:

**Tabel 4.** Kode *Elias Gamma*

N	Kode Elias Gamma
1	1
2	01 0
3	01 1
4	001 00
5	001 01
6	001 10
7	001 11
8	0001 000
9	0001 001
10	0001 010
11	0001 011
12	0001 100
13	0001 101
14	0001 110
15	001 111

Setelah itu karakter-karakter yang akan dikompresi diganti dengan kode yang terdapat pada tabel kode *Elias Gamma Code*. Setelah diganti, hitung jumlah *bit* untuk tiap karakter. Untuk hasil pergantian karakter dapat dilihat pada tabel 3.5 dibawah ini.

**Tabel 5.** Total Bit Pergantian Karakter

No	Karakter	Frekuensi Karakter	Kode Elias Gamma Code	Bit	Bit x Frekuensi
1	L	3	1	1	3
2	A	2	01 0	3	6
3	C	2	01 1	3	6
4	I	2	001 00	5	10
5	E	2	001 01	5	10
6	N	2	001 10	5	10
7	SP	2	001 11	5	10
8	D	1	0001 000	7	7
9	T	1	0001 001	7	7
10	Y	1	0001 010	7	7
11	O	1	0001 011	7	7
12	V	1	0001 100	7	7
Total bit					90 Bit

Setelah dikompresi dengan menggunakan *elias gamma code*, maka total bit yang dihasilkan adalah 90 bit. Untuk diubah menjadi satuan byte maka dibagi 8 yaitu  $90/8 = 11.25$  byte.

Tahap selanjutnya adalah menyusun kembali kode-kode yang telah dibuat pada tabel diatas sesuai dengan posisi karakter pada *string*. *String* yang telah dibaca adalah **ACCIDENTALLY IN LOVE**

<u>01 0</u>	<u>01 1</u>	<u>01 1</u>	<u>001 00</u>	<u>0001 000</u>	<u>001 01</u>	<u>001 10</u>	<u>0001 001</u>	<u>01 0</u>
A	C	C	I	D	E	N	T	A
<u>1</u>	<u>1</u>	<u>0001 010</u>	<u>001 11</u>	<u>001 00</u>	<u>001 10</u>	<u>001 11</u>	<u>1</u>	
L	L	Y	SP	I	N	SP	L	
<u>0001 011</u>	<u>0001 100</u>	<u>001 01</u>						
O	V	E						

Sehingga diperoleh *string bit* berikut:

010011011001000001000001010011000010010101100010100011100100001100011110001011000110000101.

Langkah selanjutnya setelah *string bit* diperoleh maka dilakukan penambahan *padding* dan *flagging*. Jika sisa bagi panjang *string bit* terhadap 8 adalah 0 maka penambahan 00000001. Nyatakan dengan bit akhir. Sedangkan jika sisa bagi *string bit* terhadap 8 adalah n (1,2,3,4,5,6,7) maka tambahkan 0 sebanyak  $7 - n + "1"$  di akhir *string bit*. Nyatakan dengan L. Lalu tambahkan bilangan biner dari  $9 - n$ . Nyatakan dengan bit akhir.

Karena jumlah *string bit* yang dikompresi yaitu 90 bit dan 90 tidak habis dibagi 8 dan sisanya 2 bit atau dengan kata lain  $90 \bmod 8 = 2$ , nyatakan sisa bagi tersebut dengan nilai n, maka tambahan 0 sebanyak  $7 - n + "1"$  diakhir *string*

bit. Nyatakan dengan L. Lalu tambahkan bilangan biner dari  $7 - n$  Lalu tambahkan bilangan biner dari  $9 - n$ . Nyatakan dengan bit akhir.

$$\begin{aligned}
 &7 - n + "1" \\
 7 - 2 + "1" &= 5 = "00001" \\
 &\text{Bit akhir} \\
 &9 - n \\
 9 - 2 = 7 &= "0000111"
 \end{aligned}$$

**Gambar 1.** Perhitungan Penambahan Bit

```

0100110110010000010000010100110000100101011000101000
1110010000110001111000101100011000010100000100000111
    
```

**Gambar 2.** Sting Bit Yang dilakukan Penambahan

Total 6 bit yang dihasilkan dengan cara ini adalah  $90 + 6 + 8$  hingga 104 bit. Selanjutnya bagi bit menjadi beberapa kelompok. Setiap grup terdiri dari 8 bit, seperti yang ditunjukkan gambar di bawah ini:

```

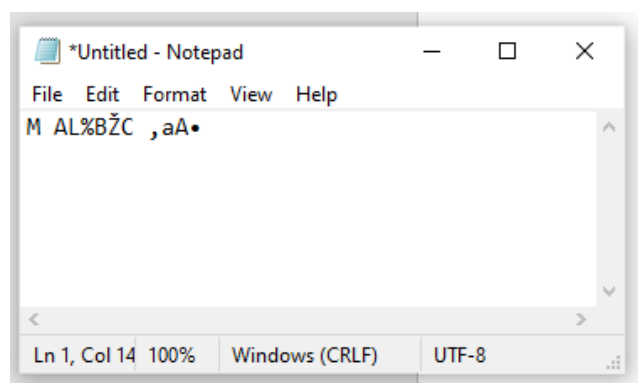
01001101    10010000    01000001    01001100
00100101    01100010    10001110    01000011
00011110    00101100    01100001    01000001
00000111
    
```

**Gambar 3.** Pembagian Sting Bit

Berdasarkan pembagian ini, ada 13 bit nilai biner terkompresi baru (13 byte) dan nilai biner bit yang ditambahkan. Setelah pembagian selesai, pertama-tama gunakan kode ASCII untuk mencari nilai desimal dari string bit untuk menemukan nilai yang telah diekspresikan bersama, lalu ubah nilai yang dibagi ke karakter lain. Adapun nilai yang sudah terkompresi dapat dilihat pada tabel di bawah ini:

**Tabel 6.** Hasil Karakter Elias Gamma

No	Biner	Desimal	Karakter
1	01001101	77	M
2	10010000	144	
3	01000001	65	A
4	01001100	76	L
5	00100101	37	%
6	01100010	98	B
7	10001110	142	Ž
8	01000011	67	C
9	00011110	30	
10	00101100	44	,
11	01100001	97	A
12	01000001	65	A
13	00000111	7	.



**Gambar 4.** Hasil Karakter Kompresi Kode Elias Gamma

Berdasarkan hasil kompresi bit total, ukuran sebelum kompresi adalah 160 bit dan ukuran setelah kompresi adalah 104 bit. Hasil kompresi berdasarkan algoritma kode gamma elias adalah:

1. Ratio of Compression ( $R_c$ )

$$R_c = \frac{\text{Jumlah Bit sebelum dikompresi}}{\text{Jumlah Bit sesudah dikompresi}}$$

$$R_c = \frac{160}{104}$$

$R_c = 1.53$  Hasil dari pembagian data sebelum dan sesudah dikompresi

2. Compression Ratio ( $C_R$ )

$$C_R = \frac{\text{Jumlah Bit sesudah dikompresi}}{\text{Jumlah Bit sebelum dikompresi}} \times 100\%$$

$$C_R = \frac{104}{160} \times 100\%$$

$C_R = 0.65\%$  → Hasil dari pembagian data sesudah dan sebelum dikompresi lalu di kali 100 %

3. Redundancy ( $R_d$ )

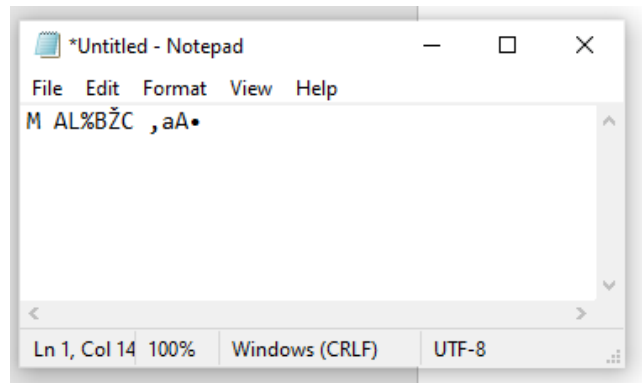
$$R_d = 100\% - C_R$$

$$R_d = 100\% - 0.65\%$$

$R_d = 0.35\%$  → Hasil kelebihan atau pemborosan dari hasil  $C_R$  diatas

2. Dekompresi Elias Gamma Code

Proses selanjutnya melakukan dekompresi. Untuk melakukan dekompresi dari *string bit* yang telah dikompresi atau *string* akhir, maka didapatkan karakter dibawah ini:



Gambar 5. Karakter Kompresi Kode Elias Gamma

Selanjutnya mengubah hasil karakter kompresi karakter elias gamma ke biner seperti pada gambar dibawah ini.

01001101	10010000	01000001	01001100
00100101	01100010	10001110	01000011
00011110	00101100	01100001	01000001
00000111			

Gambar 6. Biner Kompresi Kode Elias Gamma

dan menghasilkan *string bit* sebagai berikut:

01001101100100000100000101001100001001010110001010001110010000110001111000101100011000010100000100000111

Langkah selanjutnya menghilangkan *padding* dan *flagging* dengan cara melakukan pembacaan pada 8 bit terakhir, kemudian mengubah ke dalam bilangan desimal 0100000111 bilangan desimal angka 7 dan nyatakan dengan n, lalu menggunakan rumus  $7 + n = 7 + 7 = 14$ , jadi hilangkan 14 bit terakhir dari string bit yang telah dihasilkan. Adapun *string bit* yang telah dihilangkan *padding* dan *flagging* yaitu:

010011011001000001000001010011000010010101100010100011100100001100011110001011000110000101

Langkah selanjutnya setelah *string bit* semula diperoleh maka melakukan pengecekan dari bit pertama sampai akhir berdasarkan tabel *elias gamma code*. Seperti pada tabel dibawah ini:

Tabel 7. Pengecekan Tabel

Indeks	Nilai	Keterangan	Karakter
1	0	Tidak Ada	
2	01	Tidak Ada	
3	010	Ada Pada Tabel	A

Indeks	Nilai	Keterangan	Karakter
4	0	Tidak Ada	
5	01	Tidak Ada	
6	011	Ada Pada Tabel	C
7	0	Tidak Ada	
8	01	Tidak Ada	
9	011	Ada Pada Tabel	C
10	0	Tidak Ada	
11	00	Tidak Ada	
12	001	Tidak Ada	
13	0010	Tidak Ada	
14	00100	Ada Pada Tabel	I
15	0	Tidak Ada	
16	00	Tidak Ada	
17	000	Tidak Ada	
18	0001	Tidak Ada	
19	00010	Tidak Ada	
20	000100	Tidak Ada	
21	0001000	Ada Pada Tabel	D
22	0	Tidak Ada	
23	00	Tidak Ada	
24	001	Tidak Ada	
25	0010	Tidak Ada	
26	00101	Ada Pada Tabel	E
27	0	Tidak Ada	
28	00	Tidak Ada	
29	001	Tidak Ada	
30	0011	Tidak Ada	
31	00110	Ada Pada Tabel	N
32	0	Tidak Ada	
33	00	Tidak Ada	
34	000	Tidak Ada	
35	0001	Tidak Ada	
36	00010	Tidak Ada	
37	000100	Tidak Ada	
38	0001001	Ada Pada Tabel	T
39	0	Tidak Ada	
40	01	Tidak Ada	
41	010	Ada Pada Tabel	A
42	1	Ada Pada Tabel	L
43	1	Ada Pada Tabel	L
44	0	Tidak Ada	
45	00	Tidak Ada	
46	000	Tidak Ada	
47	0001	Tidak Ada	
48	00010	Tidak Ada	
49	000101	Tidak Ada	
50	0001010	Ada Pada Tabel	Y
51	0	Tidak Ada	
52	00	Tidak Ada	
53	001	Tidak Ada	
54	0011	Tidak Ada	
55	00111	Ada Pada Tabel	SP
56	0	Tidak Ada	
57	00	Tidak Ada	
58	001	Tidak Ada	
59	0010	Tidak Ada	
60	00100	Ada Pada Tabel	I
61	0	Tidak Ada	
62	00	Tidak Ada	
63	001	Tidak Ada	
64	0011	Tidak Ada	

Indeks	Nilai	Keterangan	Karakter
65	00110	Ada Pada Tabel	N
66	0	Tidak Ada	
67	00	Tidak Ada	
68	001	Tidak Ada	
69	0011	Tidak Ada	
70	00110	Ada Pada Tabel	SP
71	1	Ada Pada Tabel	L
72	0	Tidak Ada	
73	00	Tidak Ada	
74	000	Tidak Ada	
75	0001	Tidak Ada	
76	00010	Tidak Ada	
77	000101	Tidak Ada	
78	0001011	Ada Pada Tabel	0
79	0	Tidak Ada	
80	00	Tidak Ada	
81	000	Tidak Ada	
82	0001	Tidak Ada	
82	00011	Tidak Ada	
84	000110	Tidak Ada	
85	0001100	Ada Pada Tabel	V
86	0	Tidak Ada	
87	00	Tidak Ada	
88	001	Tidak Ada	
89	0010	Tidak Ada	
90	00101	Ada Pada Tabel	E

Proses pembacaan *string bit* dilakukan dari awal hingga ketemu 1. Catatan posisi pertama atau urutan angka 1 dan nyatakan sebagai p sehingga  $p = 2$ , apabila ada angka 0 sebelum angka 1 nyatakan jumlah 0 dengan n sehingga  $n = 1$ , lanjutkan pembacaan *string bit* setelah angka 1 sebanyak  $n = 1$ , sehingga *string bit* ketemu yaitu 010, ganti kode hasil pembacaan dengan karakter yang terdapat pada tabel, maka mendapatkan karakter A, selanjutnya lakukan cara yang sama dengan menghitung bit yang telah diproses sebelumnya dan dihasilkan *string* akhir menjadi **ACCIDENTALLY IN LOVE**, sehingga kembali ke teks sebelum dikompresi.

#### 4. KESIMPULAN

Berdasarkan penelitian yang dilakukan, dapat diambil beberapa kesimpulan dari pembahasan sebelumnya. kesimpulannya Berdasarkan proses kompresi menggunakan algoritma *elias gamma code* telah berhasil melakukan proses kompresi file teks *record database* pada aplikasi wattpad sehingga proses kompresi berjalan sesuai dengan teknik kompresi. Penerapan *algoritma elias gamma code* telah membuktikan bahwa file teks *record database* pada wattpad memiliki kapasitas penyimpanan yang besar dapat dikompresi menjadi kapasitas penyimpanan yang lebih kecil. Pada aplikasi wattpad dirancang dengan menggunakan Eclipse Juno dengan menerapkan algoritma *elias gamma code*, sehingga diharapkan memudahkan penulis dalam mengkompresi

#### REFERENCES

- [1] R. D. Pratiwi, S. D. Nasution, and F. Fadlina, "Perancangan Aplikasi Kompresi File Teks Dengan Menerapkan Algoritma Fixed Length Binary Encoding (Flbe)," *J. Media Inform. Budidarma*, vol. 2, no. 1, pp. 10–14, 2018, doi: 10.30865/mib.v2i1.813.
- [2] C. Evain, "PEMANFAATAN WATTPAD SEBAGAI APLIKASI SELF-PUBLISHING."
- [3] D. T. Sukiman, Chandra, "Aplikasi Kompresi File dengan algoritma Elias Gamma," no. 18, 2013.
- [4] D. Pratiwi, T. Zebua, and R. Parapat, "ANALISIS PERBANDINGAN KINERJA ALGORITMA FIXED LENGTH BINARY ENCODING DAN ALGORITMA ELIAS GAMMA CODE DALAM," vol. 3, pp. 424–430, 2019, doi: 10.30865/komik.v3i1.1623.
- [5] P. B. S. Ari Anwar Gucci, Muhammad Syahrizal, "Analisa Dan Implementasi Kompresi Citra Ct Scan Menggunakan Metode Delta Modulation," *Pelita Inform.*, vol. 17, no. vol.17 no.2, p. 3, 2018.
- [6] S. D. Nasution, "Perancangan Aplikasi Kompresi File Teks Dengan Menerapkan Algoritma Goldbach Codes," *Jurnal Infotek STIEKOM*, vol. 1, no. Februari, pp. 2008–2010, 2016.
- [7] H. Sartika and T. Zebua, "Perancangan Dan Implementasi Algoritma Elias Gamma Code Untuk Mengkompresi Record Database Pada Aplikasi Rangkuman Pengetahuan Umum Lengkap," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 259–265, 2019, doi: 10.30865/komik.v3i1.1600.
- [8] G. G. Maulana, "Pembelajaran Dasar Algoritma Dan Pemrograman Menggunakan El-Goritma Berbasis Web," *J. Tek. Mesin*, vol. 6, no. 2, p. 8, 2017, doi: 10.22441/jtm.v6i2.1183.
- [9] D. Salomon and G. Motta, *Handbook of data compression*. 2010.
- [10] O. Frida et al., "Analisis Perbandingan Pemampatan Data Teks Dengan Menggunakan Metode Huffman Dan Half – Byte," *Algoritma. J. Ilmu Komput. dan Inform.*, vol. 6341, no. April, pp. 1–6, 2018.
- [11] S. Suryadi, "Perancangan Aplikasi Pencarian File Dengan," vol. 2, no. 2, pp. 24–29, 2014.

- [12] Y. Sugiyani, V. Rosalina, and I. Yunan, "Perancangan Aplikasi Edukatif Berbasis Multimedia," *J. Prosisko*, vol. 1, no. September, pp. 55–59, 2014, [Online]. Available: <http://e-jurnal.lppmunsera.org/index.php/PROSISKO/article/download/89/150>.
- [13] D. P. Sari and A. Yau, "Pengembangan Menulis Kreatif Melalui Wattpad," pp. 1–4, 2017.
- [14] Q. Noer and S. El, "Motivasi Penggemar K-Pop sebagai Citizen Author Menulis Fan Fiction Motivasi Penggemar K-Pop sebagai Citizen Author Menulis Fan," vol. 12, no. 1, 2020.
- [15] Sukanto dan and Shalahuddin(2015:29), *Shalahuddin, M. Rosa A.S 2015. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Edisi REVISI*. 2015.
- [16] K. K. Budaya and T. Samosir, "APLIKASI EDUKASI BUDAYA TOBA SAMOSIR BERBASIS ANDROID," vol. 9, no. 1, pp. 9–18, 2016.
- [17] Sri Sulistiyani, *Step by Step Menjadi Programmer android*. Yogyakarta: Andi, Wahana KOMputer, 2013.
- [18] S. K. Alfa Satyaputra, M.Sc, Eva Maulina Aritonang, *JAVA for Beginners with eclipse 4.2 Juno*. Jakarta: PT Elex Media Komputindo Kelompok Gramedia, Anggota IKAPI, 2012.
- [19] Ihsan and D. P. Utomo, "Analisis Perbandingan Algoritma Even-Rodeh Code Dan Algoritma Subexponential Code Untuk Kompresi File Teks," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 4, no. 1, 2020.
- [20] S. R. Saragih and D. P. Utomo, "Penerapan Algoritma Prefix Code Dalam Kompresi Data Teks," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 4, no. 1, 2020.
- [21] 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, 2020.