

# **Penerapan Algoritma Elias Delta Code Untuk Kompresi File Video Pada Aplikasi Video Downloader**

**Jamiatul Sisca<sup>1\*</sup>**

<sup>1</sup>Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia  
Email: <sup>1\*</sup>jamisis@gmail.com

**Abstrak**—Perkembangan media komunikasi di era sekarang ini sangat pesat perkembangannya terutama bagian media komunikasi dalam bentuk aplikasi yang berfungsi untuk mengunduh suatu file. salah satunya adalah aplikasi Video Downloader yang merupakan software aplikasi yang berfungsi untuk mengunduh suatu file video ataupun file audio. Namun ada beberapa permasalahan yang ditemukan saat selesai mengunduh terutama hasil unduhan dari file video. Dimana, file video yang dibutuhkan harus memiliki resolusi yang tepat dan memiliki kualitas yang bagus. Namun, permasalahannya adalah kapasitas untuk resolusi yang dibutuhkan memiliki kapasitas yang cukup besar untuk memori penyimpanan yang telah disediakan. Maka dari itu diperlukan adanya kompresi file, dimana kompresi ini harus dilakukan dengan menggunakan algoritma yang tepat seperti menggunakan algoritma Elias Delta Code. Dimana algoritma Elias Delta Code membantu proses kerja dengan mengkompresi ukuran dari file video tersebut tanpa menurangi kualitas dari video tersebut. Aplikasi kompresi ini dibangun dengan menggunakan Eclipse Juno.

**Kata Kunci:** Kompresi File Video Algoritma Elias Delta Code.

**Abstract**— The development of communication media in today's era is very rapid, especially the part of communication media in the form of an application that functions to download a file. one of them is the Video Downloader application which is an application software that functions to download a video file or an audio file. However, there were several problems that were found when the download was finished, especially the download results from the video file. Where, the required video file must have the right resolution and have good quality. However, the problem is the capacity for the required resolution has a large enough capacity for the storage memory that has been provided. Therefore, file compression is required, where this compression must be done using an appropriate algorithm such as using the Elias Delta Code algorithm, where the Elias Delta Code algorithm helps the work process by compressing the size of the video file without reducing the quality of the video. by using Eclipse Juno.

**Keywords:** Elias Delta Code Algorithm Video File Compression.

## **1. PENDAHULUAN**

Kemajuan teknologi dalam bidang media komunikasi dari zaman ke zaman telah berkembang sangat pesat. Terutama perkembangan media komunikasi dalam bidang aplikasi berbasis android yang berfungsi untuk mengunduh suatu file. Salah satu media komunikasi yang sangat pesat perkembangannya adalah software aplikasi Video Downloader dimana software aplikasi ini berfungsi untuk mengunduh suatu file video ataupun file audio.

Namun ada beberapa permasalahan yang ditemukan saat selesai mengunduh terutama hasil unduhan dari file video. Dimana, file video yang dibutuhkan harus memiliki resolusi yang tepat dan memiliki kualitas yang bagus. Namun, permasalahannya adalah kapasitas untuk resolusi yang dibutuhkan memiliki kapasitas yang cukup besar untuk memori penyimpanan yang telah disediakan maka dari itu diperlukan adanya kompresi file.

Kompresi file adalah file-file yang digabungkan menjadi satu dengan tujuan untuk memperoleh ukuran file yang lebih kecil untuk dibandingkan dengan file aslinya. File yang dikompresi memungkinkan file lebih cepat ketika di-download dan lebih banyak file data yang tersimpan dalam media penyimpanan eksternal[1]. Meskipun kapasitas file tersebut sudah diperkecil dari ukuran sebelumnya, namun tidak sedikitpun mengurangi kualitas dari isi file video tersebut dan kompresi semacam ini bersifat lossless. Loseless adalah kompresi file data dimana hasil kompresi dan dekompresinya sama dan yang mengalami perubahan hanyalah ukuran file data saja.

Untuk melakukan proses kompresi file ini memerlukan algoritma yang tepat digunakan untuk mereduksi ukuran dari file video ini. Salah satunya adalah dengan menggunakan algoritma Elias Delta Code. Dimana algoritma

Elias Delta Code membantu proses kerja dengan mengkompresi ukuran dari file video tersebut tanpa menurangi kualitas dari video tersebut.

## **2. METODOLOGI PENELITIAN**

### **2.1 Kompresi**

Kompresi file adalah proses mengkodekan informasi menggunakan bit atau information-bearing unit yang lain yang lebih rendah dari pada representasi file yang tidak terkodekan dengan suatu sistem encoding tertentu [2]. Jika algoritma yang digunakan dalam proses kompresi dapat bekerja dengan baik, seharusnya terdapat perbedaan ukuran file yang signifikan antara file yang asli dan file yang telah dikompresi. Dalam proses kompresi juga terdapat rasio yaitu perbandingan antara ukuran file yang asli dan file yang telah dikompresi.

Kompresi file dilakukan untuk mereduksi ukuran file. Dengan melakukan kompresi atau pemadatan data maka ukuran file atau data akan lebih kecil sehingga dapat mengurangi waktu transmisi sewaktu file dikirim dan tidak banyak banyak menghabiskan ruang media penyimpan[3].

$$\text{Rasio kompresi} = \frac{\text{Ukuran file setelah dikompresi}}{\text{Ukuran file sebelum dikompresi}} \times 100 \% \quad (1)$$

## 2.2 Video

Video adalah teknologi pemrosesan sinyal elektronik mewakili gambar bergerak. Video merupakan sekumpulan gambar bergerak yang diperoleh dari hasil rekaman kamera atau hasil animasi komputer. Pada mulanya informasi video ini disimpan secara analog, sebagai perubahan bentuk gelombang secara kontinyu yang mewakili adanya perubahan warna dan kecerahan (*brightness*) dari gambar yang direkam. Di sisi lain, komputer digital hanya dapat menyimpan dan mengolah data yang bersifat biner. Untuk itu di kalangan industri komputer didefinisikan warna dalam besaran 24-bit yang dapat digunakan untuk menyimpan sekitar 16,7 juta kemungkinan warna yang berbeda. Dengan demikian data video dapat disimpan secara digital sebagai titik-titik yang masing-masing memiliki warna tertentu dan titik-titik tersebut jika disusun sebagai satu kesatuan akan membentuk suatu gambar secara utuh[5].

## 2.3 Algoritma Elias Delta Code

Elias Delta Code merupakan satu dari tiga Elias Code yang dipelopori oleh Peter Elias. Pada Elias Delta Code ditambahkan pada binary ( $\beta$ ). Elias Delta Code juga digunakan untuk melakukan pengkodean pada bilangan bulat positif [10].

Adapun aturan untuk mengkodekan sebuah bilangan dengan menggunakan Elias Delta Code adalah sebagai berikut :

1. Tuliskan  $n$  dalam bilangan biner (binary). Bit yang paling kiri (paling signifikan) akan menjadi 1.
2. Hitung jumlah bit-nya, hapus bit paling kiri dari  $n$  dan tambahkan perhitungan dalam bilangan biner (binary) pada bagian kiri dari  $n$  setelah bit paling kiri dari  $n$  dihapus.
3. Kurangi 1 dari perhitungan langkah ke-2 dan tambahkan jumlah nol ke kode.

Ketika langkah-langkah ini diterapkan pada integer ke-17, hasilnya adalah :  $17 = 100012$  (lima bit). Hapus angka 1 yang paling kiri dan tambahkan  $5 = 1012$  sehingga hasilnya  $101|0001$ . Tiga bit sudah ditambahkan, kemudian tambahkan 2 nol untuk mendapatkan kode delta  $00|101|0001$ .

Ketika langkah-langkah ini diterapkan pada  $n = 17$ , hasilnya adalah :  $17 = 2N + L = 24 + 1$ . Kode gamma dari  $N + 1 = 5$  adalah  $00101$ , dan tambahkan  $L = 0001$  sehingga hasilnya adalah  $00101|0001$ .

Tabel Elias Delta Codes yang menunjukkan 18 kode Elias Delta Codes dapat dilihat pada tabel 2.1 di bawah ini.

**Tabel 1.** Kode Elias Delta Code

$1=2^0+0 \rightarrow  L =0 \rightarrow 1$	$10 = 2^3+2 \rightarrow  L =3 \rightarrow 00100010$
$2=2^1+0 \rightarrow  L =1 \rightarrow 0100$	$11 = 2^3+3 \rightarrow  L =3 \rightarrow 00100011$
$3=2^1+1 \rightarrow  L =1 \rightarrow 0101$	$12 = 2^3+4 \rightarrow  L =3 \rightarrow 00100100$
$5=2^2+1 \rightarrow  L =2 \rightarrow 01101$	$14 = 2^3+6 \rightarrow  L =3 \rightarrow 00100110$
$6=2^2+2 \rightarrow  L =2 \rightarrow 0110$	$15 = 2^3+7 \rightarrow  L =3 \rightarrow 00100111$
$7=2^2+3 \rightarrow  L =2 \rightarrow 01111$	$16 = 2^4+0 \rightarrow  L =4 \rightarrow 001010000$
$8=2^3+0 \rightarrow  L =3 \rightarrow 00100000$	$17 = 2^4+1 \rightarrow  L =4 \rightarrow 001010001$
$9=2^3+1 \rightarrow  L =3 \rightarrow 00100001$	$18 = 2^4+2 \rightarrow  L =4 \rightarrow 001010010$

## 3. HASIL DAN PEMBAHASAN

### 3.1 Analisa Masalah

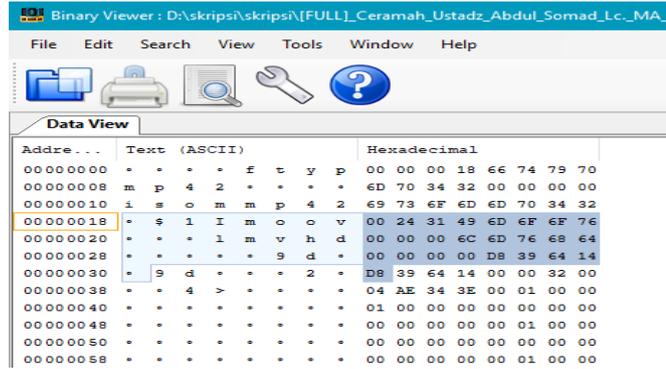
Pada tahap penelitian ini, analisa digunakan untuk menganalisis cara kerja dan mengukur efisiensi suatu perangkat lunak yang akan dirancang untuk kompresi *file* video dengan menggunakan algoritma *Elias Delta Code*. Analisa merupakan tahapan awal yang harus di lakukan sebelum melakukan perancangan terhadap sebuah sistem atau program yang akan dibangun. Tujuan dibuatnya analisa sistem ini adalah untuk mengetahui apa alasan sebuah sistem itu harus dibangun. Analisa juga berguna untuk merumuskan apa saja kebutuhan-kebutuhan yang di perlukan dari sistem tersebut dalam meminimalisirkan sumber daya yang berlebih serta membantu merencanakan penjadwalan atau pembentukan sebuah sistem.

#### 3.1.1 Penerapan Algoritma Elias Delta Code

Kompresi Berdasarkan Algoritma Elias Delta Code.

- a. Memasukkan file

Dari samplefile video di dapat nilai hexadesimal menggunakan bantuan software Binary Viewer seperti pada gambar di bawah ini:



**Gambar 1.** Nilai Hexadecimal File video Sample

Berdasarkan pada gambar di atas di dapat nilai hexadecimal file video sample. Untuk keperluan hitungan manual hanya diambil sample nilai sebanyak 25 karakter nilai hexadecimal file video sample. Nilai hexadecimal diambil dari sisi kiri baris ke-3 bilangan ke-25 sampai dengan baris ke-7 bilangan ke-49.

- b. Melakukan pembacaan isi file  
Adapun bilangan hexadecimal file video sample tersebut adalah 00, 24, 31, 49, 6D, 6F, 6F, 76, 00, 00, 00, 6C, 6D, 76, 68, 64, 00, 00, 00, 00, D8, 39, 64, 14, D8 (tanpa koma dan spasi).
- c. Menampilkan nilai asli  
Adapun nilai asli sebelum dilakukannya proses kompresi adalah sebagai berikut:

**Tabel 2.** Nilai asli sebelum dikompresi

Nilai	
Hexadecimal	Binary
00	00000000
24	00100100
31	00110001
49	01001001
6D	01101101
6F	01101111
6F	01101111
76	01110110
00	00000000
00	00000000
00	00000000
6C	01101100
6D	01101101
76	01110110
68	01101000
64	01100100
00	00000000
00	00000000
00	00000000
00	00000000
D8	11011000
39	01101000
64	01100100
14	00010100
D8	11011000

- d. Tahap selanjutnya adalah pembentukan set karakter. Karakter yang diulang akan dihilangkan sehingga hanya satu karakter yang hidup. Hasil dari proses set karakter dapat dilihat pada tabel berikut ini:

**Tabel 3.** Pembentukan set karakter

Nilai			
Hexadecimal	Binary	Frekuensi	Bits x Frex
00	00000000	8	64
24	00100100	1	8
31	00110001	1	8

Nilai			
Hexadecimal	Binary	Frekuensi	Bits x Frex
49	01001001	1	8
6D	01101101	2	16
6F	01101111	2	16
76	01110110	2	16
6C	01101100	1	8
68	01101000	1	8
64	01100100	2	16
D8	11011000	2	16
39	01101000	1	8
14	00010100	1	8
Total			200

Berdasarkan tabel di atas, satu nilai pixel hexadesimal (karakter) bernilai 8 bit bilangan biner. Sehingga 13 bilangan hexadesimal mempunyai nilai biner sebanyak 200 bit. Untuk mengubah satuan menjadi byte maka jumlah keseluruhan bit dibagikan 8. Maka dihasilkan  $200/8 = 25$  byte.

e. Pembentuk kode Elias Delta Code

Aturan dalam pembentukan kode bilangan dengan menggunakan Elias Delta Code dapat dilihat pada landasan teori pada bab sebelumnya. Adapun nilai dari Elias Delta Code dapat dilihat pada tabel berikut:

**Tabel 4.** Proses pengkodean algoritma Elias Delta Code.

Hexa	Keterangan	Kode Delta
00	1	1
64	$2 = 10_2$ (2 bit), Hapus angka 1 paling kiri dan tambahkan $2 = 10$ (2 bit), sehingga menjadi $10   0$ . 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 0100.	0100
6D	$3 = 11_2$ (2 bit), Hapus angka 1 paling kiri dan tambahkan $2 = 10$ (2 bit), sehingga menjadi $10   1$ . 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 0101.	0101
6F	$4 = 100_2$ (3 bit), Hapus angka 1 paling kiri dan tambahkan $3 = 11$ (2 bit), sehingga menjadi $11   00$ . 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 01100.	01100
76	$5 = 101_2$ (3 bit), Hapus angka 1 paling kiri dan tambahkan $3 = 11$ (2 bit), sehingga menjadi $11   01$ . 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 01101.	01101
D8	$6 = 110_2$ (3 bit), Hapus angka 1 paling kiri dan tambahkan $3 = 11$ (2 bit), sehingga menjadi $11   10$ . 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 01110.	01110
14	$7 = 111_2$ (3 bit), Hapus angka 1 paling kiri dan tambahkan $3 = 11$ (2 bit), sehingga menjadi $11   11$ . 2 bit dikurang dengan 1 sama dengan 1. Tambahkan 1 nol paling kiri menjadi 01111.	01111
24	$8 = 1000_3$ (4 bit), hapus angka 1 paling kiri dan tambahkan $4 = 100$ (3 bit), sehingga menjadi $100   000$ . 3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100000.	00100000
31	$9 = 1001_3$ (4 bit), hapus angka 1 paling kiri dan tambahkan $4 = 100$ (3 bit), sehingga menjadi $100   001$ . 3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100001.	00100001
39	$10 = 1010_3$ (4 bit), hapus angka 1 paling kiri dan tambahkan $4 = 100$ (3 bit), sehingga menjadi $100   010$ . 3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100010.	00100010
49	$11 = 1011_3$ (4 bit), hapus angka 1 paling kiri dan tambahkan $4 = 100$ (3 bit), sehingga menjadi $100   011$ . 3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100011.	00100011
68	$12 = 1100_3$ (4 bit), hapus angka 1 paling kiri dan tambahkan $4 = 100$ (3 bit), sehingga menjadi $100   100$ . 3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100100.	00100100
6C	$13 = 1101_3$ (4 bit), hapus angka 1 paling kiri dan tambahkan $4 = 100$ (3 bit), sehingga menjadi $100   101$ . 3 bit dikurang dengan 1 sama dengan 2. Tambahkan 2 nol paling kiri menjadi 00100101.	00100101

f. Proses kompresi untuk algoritma Elias Delta Code

Proses kompresi untuk algoritma Elias Delta Code dapat dilihat pada tabel berikut ini:

**Tabel 5.** Proses kompresi untuk algoritma Elias Delta Code

Nilai					
Hexadecimal	ASCII	Frekuensi	Elias Delta Code	Length	Bits
00	0	8	1	1	8
64	100	2	0100	4	8

Nilai					
Hexadecimal	ASCII	Frekuensi	Elias Delta Code	Length	Bits
6D	109	2	0101	4	8
6F	111	2	01100	5	10
76	118	2	01101	5	10
D8	216	2	01110	5	10
14	20	1	01111	5	5
24	36	1	00100000	8	8
31	49	1	00100001	8	8
39	57	1	00100010	8	8
49	73	1	00100011	8	8
68	104	1	00100100	8	8
6C	108	1	00100101	8	8
Total					107

Dari perhitungan tabel diatas setelah dikompresi dengan menggunakan algoritma Elias Delta Code adalah 107 bit. Untuk diubah menjadi satuan byte maka akan dibagi 8 yaitu  $107/8=13,3$  byte.

- g. Menperoleh string bit dan penambahan padding dan flag.

Proses ini adalah merubah nilai pixel menjadi string bit nilai biner seperti berikut ini.

10010000 00010000 10010001 10101011 00011000 11011110 01001010 10101101 00100100 01001111  
01110001 00010010 00111101 110

Kemudian sebelum di dapatkan hasil keseluruhan akhir kompresi dilakukan penambahan string bit itu sendiri yaitu padding bit dan flagbit. Jika sisa bagi panjang string bit terhadap 8 adalah 0 maka tambahan 00000001. Nyatakan dengan bit akhir. Sedangkan jika sisa bagi panjang 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  $9 - n$  dan nyatakan dengan bit akhir, karena jumlah string bit 110 tidak habis dibagi delapan dan sisanya 3 bit, nyatakan sisa bagi tersebut dengan nilai n. maka tambahkan 0 sebanyak 0 sebanyak  $7 - n + "1"$  di akhir string bit. Nyatakan dengan L. Lalu tambahkan bilangan biner dari  $9 - n$ . Nyatakan dengan bit akhir.

$$7 - n + "1"$$

$$7 - 3 + "1" = 5 \quad \mathbf{00101}$$

Bit Akhir  $9 - n$

$$\text{Bit Akhir} = 9 - 3 = 6 \rightarrow \mathbf{00000110}$$

**Gambar 2.** Perhitungan Penambahan Bit

10010000	00010000	10010001	10101011	00011000
11011110	01001010	10101101	00100100	01001111
01110001	00010010	00111101	<b>11000101</b>	<b>00000110</b>

**Gambar 3.** String Bit Yang Telah Dilakukan Penambahan

Total panjang bit keseluruhan setelah ada penambahan bit adalah  $107+5+6=118$ .

- h. Selanjutnya lakukan pemisahan bit menjadi beberapa kelompok. Setiap kelompok terdiri dari 8 bit yang dapat dilihat pada tabel di bawah ini:

**Tabel 6.** Pembagian String Bit

10010000	00010000	10010001	10101011	00011000
11011110	01001010	10101101	00100100	01001111
01110001	00010010	00111101	11000101	00000110

Berdasarkan pada pembagian kelompok nilai biner, didapatkan 15 kelompok nilai biner baru yang sudah terkompresi beserta nilai biner penambahan bit.

- i. Setelah pembagian dilakukan, maka pixel yang sudah dibagi dirubah kedalam suatu karakter dengan terlebih dahulu mencari nilai desimal dari string bit tersebut menggunakan kode ASCII untuk mengetahui nilai dari pixel yang sudah terkompresi. Adapun nilai pixel yang sudah terkompresi dapat dilihat pada tabel di berikut ini:

**Tabel 7.** Nilai desimal pixel terkompresi

Indeks Terkompresi (n)	Nilai Desimal Terkompresi
10010000	144

<b>Indeks Terkompresi (n)</b>	<b>Nilai Desimal Terkompresi</b>
00010000	16
10010001	145
10101011	171
00011000	24
11011110	222
01001010	74
10101101	173
00100100	36
01001111	79
01110001	113
00010010	18
00111101	61
11000101	197
00000110	6

Berdasarkan hasil kompresi dengan Elias Delta Codediatas dapat dihitung kinerja kompresinya yaitu :

$$\begin{aligned} \text{Compression Ratio(Cr)} &= \frac{\text{Ukurandatasetelahdikompresi}}{\text{Ukurandatasetelumdikompresi}} \times 100\% \\ &= \frac{118}{200} \times 100\% = 54\% \end{aligned}$$

2. Dekompresi Berdasarkan Algoritma Elias Delta Code.

a. Pada proses dekompresi hal yang dilakukan adalah menganalisa keseluruhan bit hasil dari kompresi sebelumnya.

Adapun bit keseluruhan hasil kompresi dapat dilihat pada tabel berikut :

**Tabel 8.** Nilai desimal dan biner pixel yang akan di dekompresi

<b>Nilai Desimal</b>	<b>Nilai Biner</b>
144	10010000
16	00010000
145	10010001
171	10101011
24	00011000
222	11011110
74	01001010
173	10101101
36	00100100
79	01001111
113	01110001
18	00010010
61	00111101
197	11000101
6	00000110

Berdasarkan pada tabel di atas maka diambil seluruh nilai biner dan digabungkan menjadi seperti berikut ini :

“10010000 00010000 10010001 10101011 00011000 11011110 01001010 10101101 00100100 01001111 01110001 00010010 00111101 11000101 00000110”

b. Selanjutnya adalah dengan mengembalikan binary menjadi string bit semula dengan menghilangkan bilangan biner yang penulisannya ditebalkan. Untuk mengembalikan binary menjadi string bit semula dapat dilakukan melalui langkah berikut ini.

1. Lakukan pembacaan pada 8 bit terakhir, hasil pembacaan berupa bilangan desimal. Nyatakan hasil pembacaan dengan n. Hilangkan bit pada bagian akhir sebanyak 7+n.

2. Setelah dilakukan perhitungan pembacaan bit akhir. Nilai biner yang dihilangkan sebanyak 8 bit pada akhir. n = 1. Hilangkan 7 + n atau 7+3 = 10. Penjelasan diatas menunjukan bahwa bit akhir harus dihilangkan. Hasil pengembalian binary menjadi string bit semula dapat dilihat sebagai berikut ini:

“1001000000010000100100011010101100011000110111100100101  
0101011010010010001001111011100010001001000111101110”

Berdasarkan perhitungan dengan algoritma elias omega codestring bit pada diatas berjumlah 107 bit seperti diawal sehingga dilakukan pembacaan string bit awal. Adapun tabel hasil perhitungan diatas adalah sebagai berikut:

**Tabel 9.** Nilai indeks keterangan

Nilai	Index	Keterangan
1	1	Ada Pada Tabel
2	0	Tidak Ada
3	00	Tidak Ada
4	001	Tidak Ada
5	0010	Tidak Ada
6	00100	Tidak Ada
7	001000	Tidak Ada
8	0010000	Tidak Ada
9	00100000	Ada Pada Tabel
10	0	Tidak Ada
11	00	Tidak Ada
12	001	Tidak Ada
13	0010	Tidak Ada
14	00100	Tidak Ada
15	001000	Tidak Ada
16	0010000	Tidak Ada
17	00100001	Ada Pada Tabel
18	0	Tidak Ada
19	00	Tidak Ada
20	001	Tidak Ada
21	0010	Tidak Ada
22	00100	Tidak Ada
23	001000	Tidak Ada
24	0010001	Tidak Ada
25	00100011	Ada Pada Tabel
26	0	Tidak Ada
27	01	Tidak Ada
28	010	Tidak Ada
29	0101	Ada Pada Tabel
30	0	Tidak Ada
31	01	Tidak Ada
32	011	Tidak Ada
33	0110	Tidak Ada
34	01100	Ada Pada Tabel
35	0	Tidak Ada
36	01	Tidak Ada
37	011	Tidak Ada
38	0110	Tidak Ada
39	01100	Ada Pada Tabel
40	0	Tidak Ada
41	01	Tidak Ada
42	011	Tidak Ada
43	0110	Tidak Ada
44	01101	Ada Pada Tabel
45	1	Ada Pada Tabel
46	1	Ada Pada Tabel
47	1	Ada Pada Tabel
48	0	Tidak Ada
49	01	Tidak Ada
50	001	Tidak Ada
51	0010	Tidak Ada
52	00100	Tidak Ada
53	001001	Tidak Ada
54	0010010	Tidak Ada
55	00100101	Ada Pada Tabel
56	0	Tidak Ada
57	01	Tidak Ada
58	010	Tidak Ada

Nilai	Index	Keterangan
59	0101	Ada Pada Tabel
60	0	Tidak Ada
61	01	Tidak Ada
62	011	Tidak Ada
63	0110	Tidak Ada
64	01101	Ada Pada Tabel
65	0	Tidak Ada
66	00	Tidak Ada
67	001	Tidak Ada
68	0010	Tidak Ada
69	00100	Tidak Ada
70	001001	Tidak Ada
71	0010010	Tidak Ada
72	00100100	Ada Pada Tabel
73	0	Tidak Ada
74	01	Tidak Ada
75	010	Tidak Ada
76	0100	Ada Pada Tabel
77	1	Ada Pada Tabel
78	1	Ada Pada Tabel
79	1	Ada Pada Tabel
80	1	Ada Pada Tabel
81	0	Tidak Ada
82	01	Tidak Ada
83	011	Tidak Ada
84	0111	Tidak Ada
85	01110	Ada Pada Tabel
86	0	Tidak Ada
87	00	Tidak Ada
88	001	Tidak Ada
89	0010	Tidak Ada
90	00100	Tidak Ada
91	001000	Tidak Ada
92	0010001	Tidak Ada
93	00100010	Ada Pada Tabel
94	0	Tidak Ada
95	01	Tidak Ada
96	010	Tidak Ada
97	0100	Ada Pada Tabel
98	0	Tidak Ada
99	01	Tidak Ada
100	011	Tidak Ada
101	0111	Tidak Ada
102	01111	Ada Pada Tabel
103	0	Tidak Ada
104	01	Tidak Ada
105	011	Tidak Ada
106	0111	Tidak Ada
107	01110	Ada Pada Tabel

Maka dari penjabaran diatas dapat dibentuk tabel Elias Delta Code dan nilai hexadecimal awal :

**Tabel 10.** Nilai hexa dan Elias Delta Code

Nilai	
Elias Delta Code	Hexadecimal
1	00
00100000	24
00100001	31
00100011	49
0101	6D
01100	6F

Nilai	
Elias Delta Code	Hexadecimal
01100	6F
01101	76
1	00
1	00
1	00
00100101	6C
0101	6D
01101	76
00100100	68
1	00
1	00
1	00
1	00
0100	64
01110	D8
00100010	39
0100	64
01111	14
01110	D8

### 3.2. Implementasi

Kebutuhan sistem merupakan spesifikasi tentang perangkat yang akan digunakan yaitu software (perangkat lunak) dan hardware (perangkat keras) yang dibutuhkan oleh aplikasi. Berdasarkan kebutuhan sistem terhadap pengimplementasian suatu sistem yang merupakan proses menerjemahkan hasil analisa dan perancangan sistem kedalam sebuah bahasa pemrograman. Pengimplementasian suatu sistem akan berpengaruh pada spesifikasi mobile yang digunakan, agar program bisa berjalan dengan baik berdasarkan spesifikasi perangkat keras (hardware) dan perangkat lunak (software) harus sesuai. Sistem pengolahan suatu program terdiri dari prosedur dan pelaksanaan data, mobile sebagai sarana pengolahan program haruslah menyediakan fasilitas-fasilitas yang mendukung dalam pengolahan nantinya. Adapun kebutuhan sistem terbagi 2 yaitu perangkat keras dan perangkat lunak.

Perangkat keras adalah peralatan dari sistem mobile yang secara fisik terlihat dan dapat disentuh. Perangkat keras yang diperlukan mobile dengan spesifikasi minimal sebagai berikut:

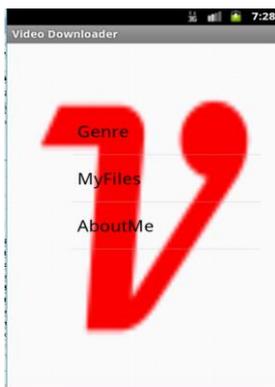
1. Chipset : Exynos 7870 Octa
2. CPU : Octa-core 1.6 GHz Cortex-A53
3. GPU : Mali-T830MP2
4. RAM : 3GB
5. Storage : 32GB (microSD, up to 256 GB)
6. Layar : PLS TFT 5.5 inci (1.920 x 1.080 piksel)
7. Jaringan : GSM/HSPA/LTE
8. Baterai : Li-ion 3.300 mAh

Perangkat lunak merupakan sebuah sistem yang digunakan untuk pengolah data atau aplikasi tertentu, sehingga perangkat lunak (software) yang digunakan dapat dikategorikan sebagai berikut:

1. Sistem Operasi Android 8.0 (Oreo).
2. Aplikasi yang digunakan sebagai emulator adalah Eclipse Juno

Tampilan program yang dibutuhkan diantaranya yaitu tampilan yang menjelaskan beberapa form aplikasi yang digunakan sebagai kebutuhan program yang dibutuhkan diantaranya yaitu form home, form genre yang digunakan untuk menampilkan list video yang akan di download, form my files yang digunakan untuk menampilkan file video yang sudah terunduh dan form about me untuk informasi tentang penulis.

Form Home merupakan form yang pertama kali muncul pada saat aplikasi dijalankan. Form utama memiliki beberapa sub menu diantaranya adalah menu form genre, form my files dan form about me. Adapun halaman form home pada aplikasi ini dapat dilihat pada gambar 4.



**Gambar 4.** Tampilan Form Home

Form Genre merupakan form yang dapat dibuka dan dijalankan perannya untuk menampilkan list video yang bisa di unduh apabila aplikasi video downloader terkoneksi oleh internet. Form genre memiliki beberapa sub menu diantaranya adalah menu form religion, form romantic, form comedy, form other yang berisikan genre video yang lain dan satu tombol exit yang berfungsi untuk kembali kehalaman sebelumnya. Adapun halaman Form Genre pada aplikasi ini dapat dilihat pada gambar 5.



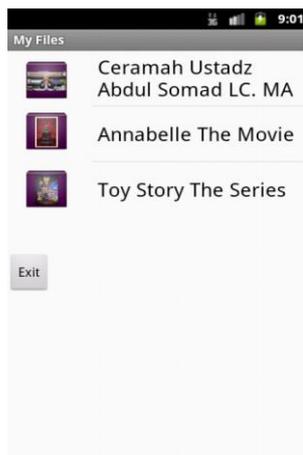
**Gambar 5.** Tampilan Form Genre

Form Religion merupakan form yang terdapat di dalam menu genre yang menampilkan list video-video religi. Dalam form religion ini, semua list video dapat diunduh namun aplikasi video downloader harus terkoneksi oleh internet. Setelah selesai mengunduh, maka sistem juga langsung melakukan kompresi pada file video yang baru di unduh. Setelah proses kompresi selesai dilakukan oleh sistem maka video akan tersimpan otomatis ke tempat yang sudah kita atur tempat penyimpanannya. Adapun Halaman form religion pada aplikasi ini dapat dilihat pada gambar 6.



**Gambar 6.** Tampilan Form Religion

Form My Files merupakan form yang menampilkan list video hasil dari unduhan yang telah terunduh dan terkompresi. Adapun halaman Form My Files pada aplikasi ini dapat dilihat pada gambar 7.



**Gambar 7.** Tampilan Form My Files

#### 4. KESIMPULAN

Berdasarkan dari penelitian yang telah dilakukan, maka hasil akhir dari penelitian tersebut dapat diambil beberapa kesimpulan. Adapun kesimpulan tersebut sebagai berikut Algoritma Elias Delta Code dapat diterapkan untuk mengkompres ukuran file video sehingga ukurannya menjadi lebih kecil dari ukuran sebelumnya. Berdasarkan dari hasil pengujian terhadap sistem bahwa ukuran file video lebih kecil setelah dilakukan kompresi. Aplikasi kompresi file video telah selesai dirancang dengan menggunakan aplikasi Eclipse Juno dan dengan menerapkan algoritma Elias Delta Code di dalamnya sehingga diharapkan memudahkan penulis dalam mengkompresi ukuran file video.

#### REFERENCES

- [1] Jubilee Enterprise, *Rahasia Manajemen File*. Jakarta: PT. Elex Media Komputindo, 2010.
- [2] M. T. Heru Nugroho, S.Si., *Matematika Diskrit dan Implementasinya dalam Dunia Teknologi Informasi*, 1st ed. Yogyakarta: deepublish, 2015.
- [3] K. Sayood, *Introduction to Data Compression*, Third. San Fransisco: Elsevier, 2006.
- [4] A. Stephen, *File Piracy*. Jakarta: PT. Elex Media Komputindo, 2007.
- [5] Iwan Binanto, *Multimedia Digital - Dasar Teori dan Pengembangannya*, 1st ed. Yogyakarta: CV. Andi Offset, 2010.
- [6] Jubilee Enterprise, *Rahasia Menjadi Jago Download*. Jakarta: PT. Elex Media Komputindo, 2010.
- [7] H. A. Mangkulo, *Aneka Tools Video Downloader*. Jakarta: PT. Elex Media Komputindo, 2011.
- [8] A. Kadir, *Pengenalan Algoritma Pendekatan Secara Visual dan Interaktif menggunakan Raptor*, 1st ed. Yogyakarta: ANDI OFFSET, 2013.
- [9] dkk. Erick Paulus, *Perangkat Komputasi Numerik Scilab Berbasis Open-Source: Algoritma Dan Penerapannya*. Yogyakarta: deepublish, 2018.
- [10] F. Edition, *Hand BOOK Of Data Compression*, Fifth. London: Springer-Verlag, 2010.
- [11] R. A.S-M.Salahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, 4th ed. Bandung: Informatika Bandung, 2016.
- [12] S. K. Alfa Satyaputra, M.Sc, Eva Maulina Aritonang, *Java for Beginners with Eclipse 4.2 Juno*. Jakarta: PT. Elex Media Komputindo, 2012.
- [13] 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.
- [14] 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.
- [15] 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.