



# Kompresi File Video Menggunakan Algoritma Punctured Elias Code

Tondhita Syah Nur Rangkuti

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia  
Jalan Sisingamangaraja No. 338, Medan, Sumatera Utara, Indonesia

Email: [tondhirangkuti124@gmail.com](mailto:tondhirangkuti124@gmail.com)

(\*: Coressponding Author)

**Abstrak-**File video memiliki ukuran yang besar, terutama bila kualitas video tersebut berkualitas tinggi dan berekstensi MP4. Dimana ekstensi MP4 merupakan format yang minim dikompresi. Hal ini menjadi sebuah masalah dalam proses pengolahan suatu file video dan juga perlu ruang penyimpanan yang besar. Apabila sebuah file video memiliki kapasitas yang besar maka dapat mempengaruhi proses dalam pengiriman file. Sehingga proses pengiriman yang dilakukan memiliki dua kemungkinan yaitu file video yang dikirim akan membutuhkan waktu proses yang lama atau file video tersebut akan mengalami kegagalan dalam pengiriman dikarenakan ukuran penyimpanan melebihi batas maksimum. Solusi dari hal tersebut dapat di selesaikan dengan cara mengkompresi data file video tersebut. Kompresi data dapat mengurangi ukuran penyimpanan data pada memori internal maupun eksternal, mempercepat pengiriman data dikarenakan ukuran penyimpanan menjadi lebih kecil. Kompresi data memiliki beberapa algoritma yang dapat digunakan. Dalam penelitian ini penulis menggunakan algoritma *punctured elias code* yang dimana merupakan salah satu teknik kompresi *lossless* sehingga hasil dekompresi dari data terkompresi sama dengan data aslinya. Perancangan sistem ini terdiri dari 2 proses utama yaitu proses kompresi dan proses dekompresi serta proses menghitung kinerja pembandingan *Ratio of Compression (RC)*, *Compression Ratio (CR)*, dan *Redudancy*. Pada proses kompresi file video berwarna yang berekstensi MP4 dengan menggunakan algoritma *Additive Code* menghasilkan *Ratio of Compression (RC)* sebesar 1,52, *Compression Ratio (CR)* sebesar 65 %, dan *Redudancy* sebesar 34 %. Pada penelitian kompresi file video menggunakan algoritma *punctured elias code* ini diharapkan dapat membantu user dalam mengkompresi file video yang awalnya berukuran besar menjadi lebih kecil.

**Kata Kunci:** Kompresi; File Video; Algoritma; Punctured Elias Code

**Abstract-**Video files are large in size, especially if they are high quality and have an MP4 extension. The MP4 extension is a minimally compressed format. This poses a problem in the processing of video files and also requires a large amount of storage space. If a video file is large in size, it can affect the file transfer process. Therefore, the file transfer process has two possible outcomes: the video file may take a long time to transfer, or the transfer may fail due to the file size exceeding the maximum storage limit. This issue can be resolved by compressing the video file data. Data compression reduces the storage size of the file on both internal and external memory, thereby accelerating the transfer process as the file size becomes smaller. Data compression has several algorithms that can be used. In this study, the author uses the *punctured Elias code* algorithm, which is a *lossless* compression technique, so that the decompressed result of the compressed data is the same as the original data. The design of this system consists of two main processes: the compression process and the decompression process, as well as the process of calculating the performance metrics of the *Compression Ratio (RC)*, *Compression Ratio (CR)*, and *Redundancy*. In the compression process of a colored video file with the MP4 extension using the *Additive Code* algorithm, the *Ratio of Compression (RC)* was 1.52, the *Compression Ratio (CR)* was 65%, and the *Redundancy* was 34%. In this video file compression research using the *punctured Elias code* algorithm, it is hoped that it can help users compress large video files into smaller ones.

**Keywords:** Compression; Video Files; Algorithm; Punctured Elias Code.

## 1. PENDAHULUAN

Durasi dan kualitas pada video merupakan salah satu faktor yang mempengaruhi ukuran dari file video. Terkadang dapat terjadi resiko ruang penyimpanan yang tidak cukup menampung file tersebut dan proses pengiriman data yang tidak tersampaikan dikarenakan ukuran dari file video yang sangat besar. Berdasarkan penelitian terdahulu mengatakan bahwa semakin besar ukuran dari file video, semakin besar pula ruang penyimpanan yang digunakan dan juga butuh waktu yang lebih lama untuk proses pengiriman atau transfer data [1]. Salah satu teknik yang dapat digunakan untuk menyelesaikan permasalahan tersebut adalah teknik kompresi. Kompresi adalah teknik untuk meminimalisir ukuran data agar dapat menyimpan jauh lebih banyak file dan juga mengurangi waktu transfer data.

Kompresi banyak digunakan karena kompresi dapat menghemat ruang penyimpanan dan juga dapat mempercepat pengiriman data dari satu orang ke orang lain. Teknik kompresi terbagi menjadi dua, yaitu teknik *Lossy* yang bekerja dengan cara membuang data yang tidak mungkin diperhatikandan teknik *Lossless* yang bekerja dengan cara mengambil file dan memindai pola, dan menerjemahkan pola-pola tersebut ke dalam hal lain yang membutuhkan ruang lebih sedikit [2]. Berdasarkan penelitian terdahulu yang menerapkan teknik kompresi terhadap file video yang memiliki ukuran besar, mengatakan bahwa ukuran file Video lebih kecil setelah dilakukan kompresi dengan rasio 15%, sehingga penelitian ini menunjukkan bahwa teknik kompresi dapat menyelesaikan permasalahan ukuran pada ukuran file video [3].

Algoritma *Punctured Elias Codes* merupakan salah satu algoritma yang dapat digunakan untuk mengimplementasikan teknik kompresi, dimana algoritma ini merupakan algoritma yang dirancang oleh Peter Fenwick dalam sebuah percobaan untuk meningkatkan performa *the Burrows-Wheeler transform*. Proses kompresi file video dengan cara mengurangi jumlah *bit rate* untuk memampatkan data digital. Beberapa istilah literatur, istilah kompresi sering disebut dengan *source coding*, *data compression*, *bandwidth compression*, dan *signal compression*. Berdasarkan penelitian yang menerapkan metode *punctured elias code* untuk mengkompresi file *hadistsoft* mengatakan bahwa file *hadist soft*

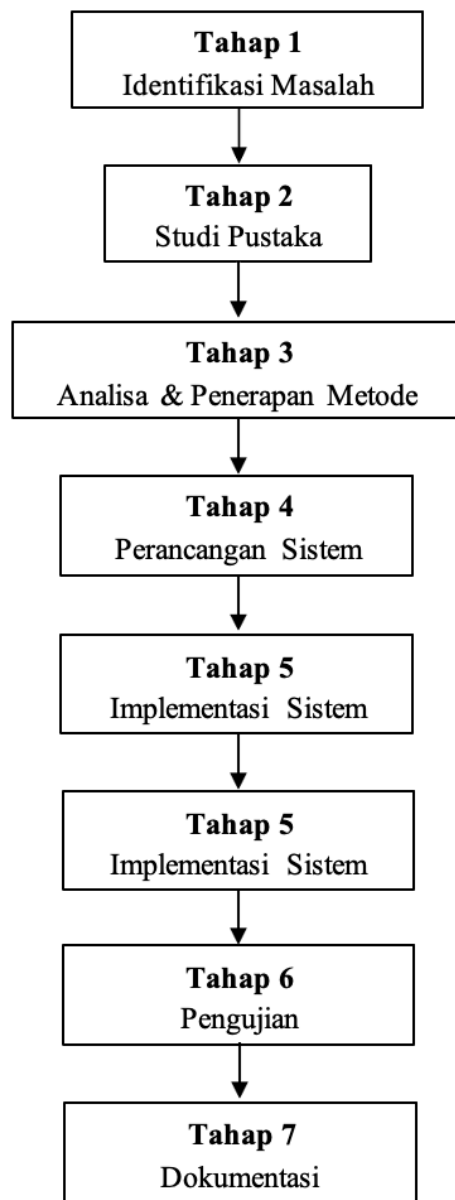
yang memiliki ukuran file *hadistoft* yang besar dapat menjadi lebih kecil setelah dilakukan proses kompresi yang ditunjukkan dengan hasil pengukuran rasio kompresi mencapai 68,75%[4].

Penelitian ini menguraikan bagaimana mengimplementasikan teknik kompresi berdasarkan metode *punctured Elias code* pada file video dengan tujuan agar ruang penyimpanan file video lebih kecil serta dapat mempercepat proses distribusinya. Format *file* video yang digunakan dalam penelitian ini adalah file video berformat MP4.

## 2. METODOLOGI PENELITIAN

### 2.1 Metodologi Penelitian

Metodologi penelitian menjabarkan kerangka kerja penelitian atau yang sering disebut dengan tahapan-tahapan pada penelitian. Kerangka kerja terdiri dari beberapa tahapan yang saling terkait secara sistematis. Kerangka kerja ini diperlukan untuk memudahkan dalam melakukan suatu penelitian. Berikut merupakan kerangka kerja dalam proses penelitian ini.



**Gambar 1.** Kerangka Kerja Penelitian

Berdasarkan diagram kerangka kerja penelitian di atas, maka dapat diuraikan pembahasan dari masing-masing tahapannya sebagai berikut:

a. Tahap Identifikasi Masalah

Tahap ini merupakan tahapan penulis untuk dapat menduga, memperkirakan dan memaparkan apa yang menjadi sumber masalah dalam penuhnya kapasitas penyimpanan serta lambatnya proses tranfer pada pengiriman *file* video dengan format MP4 hingga tidak dapat lagi tersimpan pada ruang penyimpanan.



- b. Tahap Studi Pustaka  
Tahapan ini dilakukan pemahaman terhadap objek yang akan diteliti dengan membaca berbagai sumber referensi seperti buku, jurnal, maupun sumber bacaan lainnya.
- c. Tahap Analisa Proses Kompresi File Video  
Tahap analisa ini dilakukan untuk mengetahui proses kompresi pada file video berdasarkan algoritma *punctured elias code* yang bertujuan untuk memperkecil ukuran data, dimana penerapan dari algoritma *punctured elias code* ditujukan untuk menghitung *code wode* dari setiap n yang ada.
- d. Tahap Analisa Proses Dekompresi Pada File Video  
Proses pengembalian file video yang telah dikompresi menjadi kebentuk file asli sebelum dikompresi menggunakan algoritma *punctured elias code*.
- e. Tahap Perancangan  
Tahap ini dilakukan untuk menguji hasil kompresi file video yang berformat MP4 Proses ini bertujuan untuk mengetahui hasil akhir dari penelitian yang dilakukan.
- f. Tahap Implementasi  
Tahap dokumentasi merupakan tahap akhir dari pelaksanaan penelitian yang dibuat dalam bentuk laporan. Tahap ini akan menjelaskan aplikasi yang telah dibuat agar memudahkan pembaca yang ingin mengembangkan aplikasi lebih lanjut.
- g. Tahap Pengujian  
Tahap ini dilakukan untuk menguji hasil kompresi file video berformat MP4. Proses ini bertujuan untuk mengetahui hasil akhir dari penelitian yang dilakukan
- h. Tahap Dokumentasi  
Tahap dokumentasi merupakan tahap terakhir dari pelaksanaan penelitian yang dibuat dalam bentuk laporan. Tahap ini akan menjelaskan aplikasi yang telah dibuat agar memudahkan pembaca yang ingin mengembangkan aplikasi lebih lanjut.

### 2.2 Algoritma Punctured Elias Code

Algoritma *Punctured Elias Code* yang merupakan temuan dari Peter Fenwick berhasil merancang kode bilangan integer ketika melakukan percobaan dalam meningkatkan performa transformasi Burrows Wheele. Istilah "*punctured*" datang dari tempat pengawasan Error-Control Code (ECC). ECC terdiri dari data asli digabung dengan sejumlah bilangan dari *check bit*. Beberapa *check bit* dihilangkan untuk mempersingkat rangkaian kode tersebut. Langkah-langkah untuk membangun *Punctured Elias Code* [8]. adalah sebagai berikut:

- a. Ubah n menjadi bilangan biner.
- b. Balikan bit-bit bilangan biner tersebut (Reversed).
- c. Buat flag dengan menuliskan bit 1 sebanyak jumlah bit 1 pada bilangan biner n ditambah 1 bit 0 diakhir flag.
- d. Gabung flag dengan biner yang sudah dibalikkan (*reversed*).

*Punctured Elias Code* yang dibentuk dari proses ini dinamakan kode P1, kode ini dimulai dengan bit 1 (paling sedikit terdapat satu flag, kecuali untuk P1 dengan nilai n = 0) dan juga diakhiri dengan bit 1. Untuk membentuk kode *punctured* yang lain yakni P2 adalah dengan  $P2 = P1(n+1)$ , di mana bit 1 yang paling besar akan dihapus. Berikut ini merupakan tabel kode *Punctured Elias* P1 dan P2 untuk beberapa bilangan.

Tabel 1. Tabel Beberapa *Punctured Elias Code*

n	Binary of n	Reversed	Flag	Flag  Reversed	P1	P2
0	0	-	-	-	-	01
1	1	1	10	10 1	101	001
2	10	01	10	10 01	1001	1011
3	11	11	110	110 11	11011	0001
4	100	001	10	10 001	10001	10101
5	101	101	110	110 101	110101	10011

## 3. ANALISA DAN PEMBAHASAN

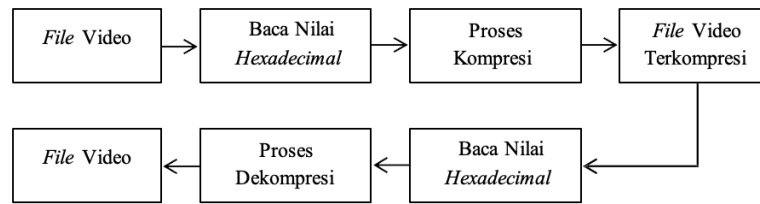
### 3.1 Analisa Kompresi File Video

Analisa permasalahan yang akan diuraikan pada penelitian ini merupakan perhitungan dan pengkompresian file video yang pada tahap prosesnya penulis menggunakan algoritma *punctured elias code*. Algoritma *punctured elias code* merupakan salah satu teknik kompresi lossless, dimana dapat memperkecil ukuran kapasitas suatu data pada file berdasarkan karakter yang terdapat pada objek yang akan dikompres. Penelitian ini dilakukan pengkompresian dengan menerapkan algoritma *punctured elias code* terhadap file video yang berformat MP4 yang biasanya memiliki ukuran yang cukup besar.

Sehingga hasil dari penelitian ini diharapkan akan bermanfaat dalam pengurangan ukuran data suatu file video. Proses awal analisa yang dilakukan penulis ialah dengan mencari file video yang akan dikompres kemudian diubah



menjadi nilai hexadecimal dengan menggunakan aplikasi HxD, setelah nilai hexadecimal pada file video telah di dapatkan. Tahapan yang akan dilakukan selanjutnya adalah proses kompresi dan diperoleh file hasil kompresi. Berikut merupakan alur sederhana proses pengkompresian file video:



**Gambar 2.** Prosedur Kompresi dan Dekompresi *File* video

### 3.2 Analisa Penerapan Algoritma Punctured Elias Code

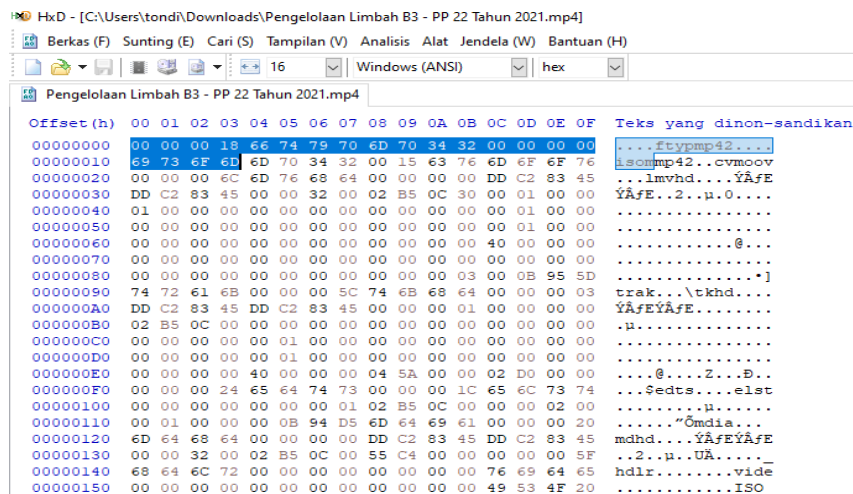
Penelitian ini akan membahas setiap proses dalam pengkompresian file video yaitu proses kompresi dan proses dekompresi. Penulis akan melakukan kompresi file video dengan menerapkan algoritma punctured elias code yang merupakan salah satu teknik kompresi lossless. Sebelum file MP4 dikompres, lebih dulu dilakukan pembacaan file MP4 agar didapatkan nilai hexadesimal menggunakan aplikasi HxD. Berikut contoh file MP4 yang dikompres serta didekompresi.

#### 3.2.1 Analisa proses kompresi file MP4 memakai Algoritma Punctured Elias Code

**Tabel 2.** Sampel File MP4 yang akan dikompresi

<u>Nama File Pengelolaan Limbah B3 – PP 22 Tahun 2021</u>	
Jenis Item	File MP4
Ukuran File	121.28 MB
Durasi	59.08 Menit

Untuk melakukan kompresi dengan algoritma Punctured Elias Code, setelah warna diurutkan sesuai dengan frekuensi tertinggi, frekuensi tertinggi akan didefinisikan sebagai  $n = 0$ , frekuensi tertinggi kedua sebagai  $n = 1$  dan seterusnya sampai mencakup seluruh pixel pada gambar. Nilai  $n$  tersebut akan diubah menjadi bilangan biner sehingga dihasilkan bit Reversed, kecuali untuk  $n = 0$ , kode PEC adalah 0. Selanjutnya akan dibuat suatu flag, dimana flag ini dihasilkan dengan menuliskan bit 1 sebanyak jumlah bit 1 pada  $n$  dan diakhiri dengan penambahan bit 0 diakhir flag, sehingga akan dihasilkan kode PEC dengan menggabungkan flag dengan bit Reversed sebelumnya. Kode Punctured Elias Code tersebut dapat dilihat pada gambar di bawah ini:



**Gambar 3.** Nilai Hexadecimal Sampel File Video

Berdasarkan gambar nilai hexadecimal di atas, penulis mengambil 20 sampel nilai hexadecimal untuk proses pengkompresian file video.

**Tabel 3.** Nilai Bilangan Hexadecimal

00	00	00	18	66	74	79	70	6D	70
34	32	00	00	00	00	69	73	6F	6D

Dari tabel di atas, maka menghasilkan bilangan hexadecimal file MP4 yang akan dihitung secara manual. Berikut nilai hexadecimal diurutkan berdasarkan frekuensinya, nilai frekuensi paling banyak maka akan berada di urutan pertama.

**Tabel 4.** Nilai Hexadecimal Yang Belum Dikompresi

n	Nilai hexa	Nilai biner	bit	freq	Bit x freq
1	00	00000000	8	7	56
2	18	00010010	8	1	8
3	66	01100110	8	1	8
4	74	01110100	8	1	8
5	79	01111001	8	1	8
6	70	01110000	8	2	16
7	34	01000110	8	1	8
8	32	00100000	8	1	8
9	69	001101111	8	1	8
10	73	1001001	8	1	8
11	6F	01101101	8	1	8
12	6D	01101001	8	2	16
<b>Total bit</b>					<b>160</b>

Setelah bilangan Hexadecimal diurutkan berdasarkan frekuensi kemunculan dan didapatkan nilai dari biner, maka selanjutnya menghitung nilai bit dan pengurutan kode punctured elias code serta memperoleh hasil bit pada file terkompresi. Adapun proses kompresi file video dapat dilihat pada table 5 dibawah ini

**Tabel 5.** Pengurutan Kode Punctured Elias Code

n	Nilai hexa	Codeword	bit	freq	Bit x freq
1	00	0	1	7	7
2	18	010	3	1	3
3	66	0110	4	1	4
4	74	10100	5	1	5
5	79	11001	5	1	5
6	70	110000	6	2	12
7	34	000110	6	1	6
8	32	0100000	7	1	7
9	69	101111	6	1	6
10	73	1001001	7	1	7
11	6F	1101101	7	1	7
12	6D	01101001	8	2	16
<b>Total bit x freq</b>					<b>85</b>

Setelah kode diurutkan berdasarkan perhitungan di atas, maka tahapan selanjutnya yaitu kembali string bit yang telah dihasilkan dari proses kompresi sesuai dengan posisi pada bilangan hexadecimal. Adapun prosesnya dapat dilihat pada table 6 dibawah ini.

**Tabel 6.** String Bit Hasil Kompresi Menggunakan Algoritma Punctured Elias Code

<b>00</b>	<b>00</b>	<b>00</b>	<b>18</b>	<b>66</b>
0	0	0	010	0110
<b>74</b>	<b>79</b>	<b>70</b>	<b>6D</b>	<b>70</b>
10100	11001	110000	01101001	110000
<b>34</b>	<b>32</b>	<b>00</b>	<b>00</b>	<b>00</b>
000110	0100000	0	0	0
<b>00</b>	<b>69</b>	<b>73</b>	<b>6F</b>	<b>6D</b>
0	101111	1001001	1101101	01101001

Berdasarkan table di atas string bit yang dihasilkan dari proses pengompresian file video menggunakan algoritma Punctured Elias Code dapat ditulis sebagai berikut:

**Tabel 7.** String Bit Hasil Kompresi

0	0	0	010	0110
10100	11001	110000	01101001	110000
000110	0100000	0	0	0
0	101111	1001001	1101101	01101001

Kemudian hasil dari total bit x frekuensi yang bernilai 90 maka tidak perlu dilakukan penambahan jumlah bit karena nilai 90 tidak bisa habis dibagi 8. Dimana didalam komputer suatu karakter dipresentasikan oleh bilangan ASCII yang dimana terdapat sebanyak 8 bit dari bilangan biner. Apabila bit dari suatu data lebih pendek dari 8 bit maka biasanya tidak akan bisa dibaca oleh komputer sebelum dilakukan proses encoding. Dikarenakan 85 tidak habis dibagi 8 atau bukan

kelipatan 8 maka dibentuk variabel yang disebut dengan padding dan flagging untuk penambahan bit data. Padding merupakan penambahan bit data yang telah dikompres agar seluruh jumlah bit data tersebut kelipatan 8.

Rumus padding ialah  $7 - n + "1"$ . Sedangkan flagging merupakan penambahan 8 bit bilangan biner setelah padding yang dimana untuk mempermudah dalam membaca bit-bit hasil kompresi pada saat proses dekompresi. Rumus flagging ialah  $9 - n$ . Apabila jumlah bit habis dibagi 8 atau kelipatan 8, maka tidak perlu dilakukan padding, tetapi tetap harus menambahkan flagging.

**Tabel 8.** Penambahan padding dan flagging

Padding	Flagging
$85 \text{ MOD } 8 = 5 = n$	$8 - n$
$7 - n + "1" =$	$8 - 5 = 3 = \mathbf{00000011}$
$7 - 5 + "1" = 3$	
<b>000</b>	

Jadi string bit penambahan padding dan flagging yang terbentuk adalah:  
 "00001001101010011001110000011010011100000001100100000000010111110  
 010011101101011010010000000011"

Apabila jumlah string bit di atas dibagi dengan 8, maka sisanya adalah 2 bit ( $85 \text{ mod } 8 = 5$ ), berdasarkan perhitungan tersebut, maka padding dibutuhkan  $7 - n + "1"$  atau  $7 - 5 + "1"$ , sehingga diperoleh bit untuk padding adalah "00" + "1" atau

001. Sehingga string biner terkompresi setelah ditambah padding adalah:  
 "00001001101010011001110000011010011100000001100100000000010111110  
 010011101101011010010000000011 001"

Berikutnya ditambahkan flagging sebanyak 8 bit dari biner nilai desimal flagging dengan rumus  $9 - n = 9 - 5 = 4$ , sehingga biner dari 4 (0000100) ini akan dijadikan sebagai flagging dan ditambahkan setelah biner padding.

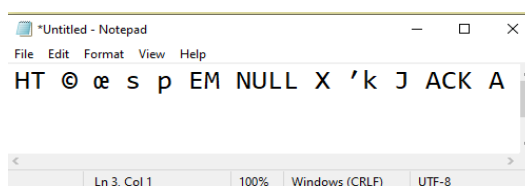
Sehingga bit akhir hasil kompresi adalah:  
 "00001001 10101001 10011100 00011010 01110000 00011001 00000000  
 01011111 00100111 01101011 01001000 00000110 01000001 00"

Sehingga total dari panjangnya bit akhir hasil kompresi adalah 106. Selanjutnya dilakukan pemisahan bit menjadi beberapa kelompok yang dimana setiap kelompok terdiri dari 8 bit, kemudian dikonversi menjadi karakter untuk dijadikan file text sebagai hasil kompresi.

**Tabel 9.** Pengelompokan Bit

No	Biner	Hexadecimal	Character
1	00001001	09	HT
2	10101001	A9	©
3	10011100	29	œ
4	00011010	73	s
5	01110000	70	p
6	00011001	19	EM
7	00000000	00	NULL
8	01011111	58	X
9	00100111	27	'
10	01101011	6B	k
11	01001000	4A	J
12	00000110	06	ACK
13	01000001	41	A

Berdasarkan pada hasil pemisahan bit di atas, maka terbentuklah 13 kelompok dengan nilai biner baru yang telah terkompresi. Karakter yang dihasilkan pada table 9 di atas akan dijadikan sebagai file text dengan ekstensi txt sebagai file hasil kompresi dari file video, seperti di bawah ini:



**Gambar 4.** File Hasil Kompresi dengan tipe txt



File video yang telah dikompresi dengan menggunakan algoritma Punctured Elias Code, tipe file nya akan diubah menjadi tipe baru sesuai dengan format yang telah ditentukan sebagai pengenalan bahwa file tersebut telah dikompres dengan aplikasi kompresi yang menerapkan algoritma Punctured Elias Code Format file yang telah ditentukan adalah MP4. Untuk mengetahui tingkat kinerja algoritma kompresi, maka penulis akan mengukur hasil kompresi file video tersebut sesuai dengan parameter yang telah ditentukan. Semakin kecil hasil file kompresinya maka semakin berkualitas kinerja kompresinya, begitu sebaliknya. Berdasarkan perhitungan di atas dapat disimpulkan bahwa persentase hasil kompresi file video dengan menggunakan algoritma Punctured Elias Code sebesar 34%.

## 4. KESIMPULAN

Berdasarkan dari pembahasan penelitian yang telah dilakukan, maka penulis mendapatkan hasil akhir dari penelitian tersebut yang disimpulkan menjadi beberapa kesimpulan yaitu Setelah mengikuti prosedur kompresi dengan menggunakan algoritma punctured elias code menghasilkan bahwa suatu file video yang memiliki ukuran yang cukup besar dapat dikompres menjadi file video yang baru dengan ukuran yang lebih kecil. Setelah menerapkan algoritma punctured elias code untuk mengkompresi file video telah membuktikan bahwa file video tersebut telah berhasil dikompresi, dengan hasil Ratio of Compression (RC) = 1,52, Compression Ratio (CR) = 65%, dan Redudancy (Rd) = 34%. Aplikasi kompresi file video dengan menerapkan algoritma punctured elias code yaitu menggunakan Microsoft Visual Studio 2008 dapat berjalan dengan baik. Sistem yang dirancang dalam penelitian ini masih memiliki kekurangan sehingga dapat dikembangkan dan diperbaiki lebih lanjut. Adapun saran yang dapat diberikan penulis yaitu Dalam penerapan algoritma punctured elias code dapat dikembangkan lagi dengan penambahan objek seperti file dokumen, file video, file audio, dan lain-lain. Sehingga memberikan manfaat yang lebih baik dimasa mendatang. Dalam penelitian ini, perancangan aplikasi kompresi file video hanya dapat mengkompresi file gambar yang berekstensi \*.MP4 dan diharapkan kedepannya dapat mengkompres file video dengan ekstensi yang lain seperti \*.AVI, \*.MPG, dan lainnya. Aplikasi kompresi file video kedepannya dapat dikembangkan menjadi berbasis Web sehingga dapat digunakan secara online

## REFERENCES

- [1] D. Riyansyah, "Perancangan Aplikasi Kompresi File Video Menggunakan Algoritma Interpolative Coding," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 3, no. 1, pp. 392–397, 2019, doi: 10.30865/komik.v3i1.1618.
- [2] M. R. Prasetyo, "Analisis Perbandingan Kinerja Alogaritma Shannon Fano dan Levenstein Code pada Kompresi File Video," Univ. Sumatera Utara, 2019, [Online]. Available: <https://library.usu.ac.id>
- [3] D. Iqbal, "Implementasi Algoritma Levenstein Untuk Kompresi File Video Pada Aplikasi Chatting Berbasis Android," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 3, no. 1, pp. 266–273, 2019, doi: 10.30865/komik.v3i1.1601.
- [4] D. Yuni, "Kompresi File Aplikasi Hadist Dengan Menggunakan Algoritma Punctured Elias Code," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, pp. 364–367, 2020, doi: 10.30865/komik.v4i1.2721.
- [5] J. A. Simatupang, G. L. Ginting, and F. T. Waruwu, "Perancangan Aplikasi Kompresi File Video Dengan Menggunakan Algoritma Start Step Stop Code," vol. 1, no. 2, pp. 41–48, 2022.
- [6] D. Riyansyah, "Perancangan Aplikasi Kompresi File Video Menggunakan," vol. 3, no. 2, pp. 392–397, 2019.
- [7] S. Simanjuntak, "Implementasi Metode Taboo Code Untuk Kompresi File Video," Explorer (Hayward), vol. 2, no. 1, pp. 32–38, 2022, doi: 10.47065/explorer.v2i1.156.
- [8] Simamora, "UNIVERSITAS SUMATERA UTARA Poliklinik UNIVERSITAS SUMATERA UTARA," J. Pembang. Wil. Kota, vol. 1, no. 3, pp. 82–91, 2018.
- [9] N. Rizka, S. D. Nasution, and K. Ulfa, "Penerapan Algoritma Elias Omega Code Untuk Kompresi File Video Pada Aplikasi Rekam Layar," vol. 9, no. April, pp. 257–265, 2021.
- [10] R. Syahputra, "Kompresi File Video Mp4 Dengan Menggunakan Metode," J. Ris. Komput., pp. 52–57, 2016.
- [11] M. Alfarizi and S. Aripin, "Penerapan Algoritma Prefix Code Dalam Kompresi File Video," KOMIK (Konferensi Nas. ..., vol. 4, pp. 249–252, 2020, doi: 10.30865/komik.v4i1.2686.
- [12] S. S. Dwi Megasari, Bakri Tanet, "Analisis dan Perancangan Kompresi Vidio Dengan Algoritma Run length Encoding," Jursan Tek. Inform. Skripsi Sarj. Komput., 2006.
- [13] K. E. Prasetyo, T. W. Purboyo, and R. E. Saputra, "Comparison of Text Data Compression Using Run Length Encoding, Arithmetic Encoding, Punctured Elias Code and Goldbach Code," Int. J. Eng. Technol., vol. 9, no. 5, pp. 3821–3827, 2017, doi: 10.21817/ijet/2017/v9i5/170905151.