

Penerapan Algoritma Punctured Elias Codes Pada File Transfer Berbasis Android

Muhammad Rezki Wasallam* Muhammad Syahrizal, Ilhamsyah

Fakultas Ilmu Komputer Dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email : reskiranto2017@gmail.com

Abstrak-Ukuran File teks terkadang relatif besar dimana semakin baik kualitas file teks yang dihasilkan, maka ukuran file yang dibutuhkan untuk menyipkan file teks tersebut semakin besar. Dengan ukuran file teks yang sangat besar, pada saat melakukan proses pemindahan bisa saja proses pemindahan gagal karena media ruang penyimpanan melebihi batasnya. Adapun Solusi dalam permasalahan ini adalah bagaimana file teks tersebut dapat dikompresi guna untuk mempercepat pemindahan dan penyimpanan file teks. Kompresi pada file teks dilakukan dengan memperkecil ukuran file teks dengan proses mengurangi bit pada file teks, akan tetapi tidak menghilangkan data informasi didalamnya.

Dengan melakukan kompresi, data yang besar akan berkurang ukurannya sehingga dapat menghemat ruang penyimpanan. Dalam penelitian ini, algoritma yang digunakan adalah punctured elias codes, dengan menggunakan algoritma tersebut, hasil kompresi dari nilai mempunyai hasil yang berbeda-beda dari setiap nilai nya, dan hasil kompresi akan menguntungkan dalam melakukan pengiriman, dan pemindahan file teks akan semakin mudah.

Kata kunci: kompresi file teks; algoritma punctured elias; code

Abstract-Text file size is sometimes relatively large where the better the quality of the resulting text file, the larger the file size needed to store the text file. With a very large text file size, when carrying out the transfer process the transfer process may fail because the media storage space exceeds the limit. The solution to this problem is how the text file can be compressed in order to speed up the transfer and storage of text files. Compression in text files is done by reducing the size of the text file by reducing the bits in the text file, but not eliminating the information data in it. By compressing, large data will be reduced in size so as to save storage space. In this study, the algorithm used is punctured elias codes, by using this algorithm, the compression results of the values have different results for each value, and the compression results will be beneficial in sending, and moving text files will be easier.

Keywords: Text File Compression; Punctured Elias; Code Algorithm

1. PENDAHULUAN

Dunia teknologi berkembang dengan cepat dan semakin maju, tidak mudah dipungkiri kecanggihan teknologi banyak berperan dalam meringankan pekerjaan manusia. Salah satunya kebutuhan terhadap *transfer* atau pengiriman *file* yang semakin besar saat ini karena segala sesuatu sekarang bisa dilakukan hanya melalui *smartphone*. Pengguna *smartphone* sangat sering melakukan *transfer file*, baik berupa gambar, audio, video, dokumen, ataupun *file* sejenis lainnya [1]-[2]. Saat ini banyak aplikasi *transfer file* yang dapat ditemukan di *smartphone* salah satunya *share it* dan aplikasi *file transfer* lainnya, yang sangat banyak ditemukan untuk melakukan *mentransfer file*. Ukuran *file* yang ditransfer akan menjadi salah satu masalah jika ukuran suatu *file* tersebut sangat besar. Seiring data yang berkembang dan proses saat ini menjadi semakin besar dari waktu ke waktu mengakibatkan lamanya proses pengiriman karena ukuran *file* yang terlalu besar yang memungkinkan aplikasi tersebut lama mengirimkan *file*. Ukuran *file* yang besar juga menjadi masalah dalam penggunaan ruang penyimpanan pada sebuah *smartphone* [3].

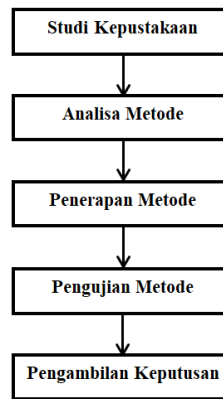
PDF singkatan dari *Portal Document Format*, yaitu *file* yang dibuat oleh perusahaan Adobe pada 1993 untuk pertukaran dokumen. Seiring dengan berlalunya waktu, dilakukan peningkatan terhadap versi PDF [4]. Hingga saat ini, versi PDF terakhir adalah 1.7 yang dirilis pada 2009 sesuai standar ISO 32000. Pada penyimpanan data hanya dibatasi dengan kapasitas tertentu [5]-[6]. Ukuran *File* PDF yang terlalu besar selalu menjadi masalah pada penyimpanan memori. Dengan munculnya masalah ini maka data tersebut dapat di atasi yaitu dengan cara mengkompresi *file* PDF yang asli menjadi lebih kecil. *File* yang telah di kompresi dapat di kembalikan ke data yang asli. Sehingga sebuah penyimpanan dapat menyimpan lebih banyak data dan proses pengiriman data pun semakin berjalan dengan cepat setelah data tersebut di kompresi [7]-[8].

Berdasarkan hasil penelitian sebelumnya yang dilakukan oleh Dedek Andri Yansyah pada tahun 2015 telah berhasil mengkompresin *file text* dengan menggunakan metode *punctured elias codes* yang berektensi **doc* hasil *output* dekompresi telah diatur dengan format perataan *text justify* dan jenis *font times new roman* dengan ukuran *font* 12.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Pada penelitian ini dapat dijelaskan langkah-langkah dalam proses pengambilan keputusan untuk memecahkan suatu masalah dengan menggunakan metode-metode yang digunakan seperti pada Gambar 1.



Gambar 1. Tahapan Proses Penelitian

Dari gambar diatas dapat dijelaskan dengan sebagai berikut:

1. Studi Kepustakaan
Studi kepustakaan adalah teknik dalam mengumpulkan suatu data informasi untuk proses penelitian dengan tujuan tinjauan pustaka yang berada di perpustakaan berdasarkan dari referensi-referensi yang telah tertulis yang relevan dilihat dari suatu penelitian yang akan dilakukan oleh penulis penelitian ini.
2. Analisa Metode
Analisa Metode sekelompok data yang merupakan suatu tahapan-tahapan dari proses dalam suatu penelitian, yang telah dikumpul agar diolah didalam penelitian ini.
3. Penerapan Metode
Penerapan metode ini memiliki peran yang begitu penting didalam sebuah penelitian. Dalam penelitian metode yang diterapkan ialah Punctured Elias Codes.
4. Pengujian Metode
Pengujian Metode adalah suatu cara dalam menentukan pengujian untuk data yang diuji secara langsung dan memiliki kemauan dalam menentukan masalah yang dapat dikerjakan. Dalam penelitian ini diterapkan menggunakan Punctured Elias Codes.
5. Pengambilan Kesimpulan
Pengambilan Kesimpulan memiliki tujuan untuk memberi penjelasan perbandingan kepada kelayakan metode Punctured Elias Codes.

2.2 Algoritma

Dalam sistem pengkompresian file transfer, metode yang digunakan pada sistem ini adalah dengan menggunakan algoritma fibonacci code [9]-[10]. Sub bab ini menjelaskan tentang tahapan atau langkah-langkah dari proses algoritma yang digunakan. Tahapan pertama yang dilakukan yaitu proses pengkompresian dan tahap kedua yaitu proses dekomposisi terhadap file transfer yang telah terkompresi. Adapun penyusunan algoritma yang digunakan berbentuk flowchart [11]-[12].

2.3 Algoritma Punctured Elias Codes Pada Kompresi Aplikasi

Kode *punctured* ini dinamakan kode P1 dan kode ini dimulai dengan 1 (paling sedikit terdapat satu *flag*, kecuali untuk kode P1 dengan $n=0$) dan juga diakhiri dengan 1 (karena n yang asli, yaitu bit MSB (*Most Significant Bit*) adalah 1, telah dibalikkan). Berikut ini adalah langkah-langkah dalam algoritma kompresi *Punctured Elias Code*:

1. Membaca string yang terdapat pada teks sms.
2. Membuat tabel karakter dan frekuensi kemunculan karakter .
3. Membangkitkan kode *punctured p1* berdasarkan banyaknya karakter yang telah Dibuat.
4. Sistem melakukan proses kompresi berdasarkan tabel karakter dan tabel kode yang telah dibangkitkan.

Dan berikut adalah langkah-langkah dekomposisi pada algoritma *punctured elias code*.

1. Proses awal dekomposisi adalah membaca nilai *flag* bit dari keseluruhan bit tersebut dengan cara merubah nilai 8 bit terakhir kedalam nilai decimal.
2. Selanjutnya hapus *flag* bits dan *padding* dari nilai keseluruhan bit.
3. Pengecekan bit yaitu dengan cara melakukan cek bit dari bit pertama dengan tabel kode *punctured elias code*. Jika ditemukan bit yang sesuai dengan tabel kode *punctured elias code* maka ubah nilai string yang sesuai sehingga akan mendapatkan hasil semula sebelum dikompresi.

3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

Berdasarkan penelitian ini, akan dilakukan dalam *aplikasi* yaitu dengan cara menerapkan algoritma *Punctured Elias Code* pada kompresi *aplikasi*. Untuk menerapkannya terlebih dahulu mencari nilai biner *aplikasi* pada aplikasi binary viewer, setelah mendapatkan nilai biner maka biner tersebut disusun untuk mencari frekuensinya, setelah nilai frekuensi tersusun selanjutnya melakukan kompresi. Kompresi dengan menggunakan metode algoritma *Punctured Elias Code* dapat memperkecilkan ukuran *aplikasi* yang besar menjadi lebih kecil setelah mendapatkan hasil kompresi maka tahap selanjutnya mengembalikan hasil *aplikasi* yang sudah dikompresi mejadi *aplikasi* asli yaitu dengan cara mendekompresi *file* tersebut. *File* yang didekompresi akan kembali ke *aplikasi* sebelumnya. Cara kerja algoritma *Punctured Elias Code* pada kompresi dan dekompresi *aplikasi* yaitu dengan menginputkan *aplikasi* setelah *aplikasi* diinput maka nilai biner *aplikasi* terbaca dan nilai biner tersebut diurutkan berdasarkan *frekuensi*.

3.2 Penerapan Metode Punctured Elias Codes Pada Kompresi Aplikasi

Contoh kasus dari masalah ini adalah sebuah teks PDF, dengan menggunakan algoritma *Punctured Elias Code*. Berikut ini merupakan contoh proses kompresi dengan metode *Punctured Elias Code*.

1. Input karakter yang akan dijadikan teks PDF, sebagai berikut :
 mampu mempermudah mahasiswa
 Bit String : mampu mempermudah mahasiswa
 [string] : 26
 Σ : {m, a, p, u, sp, e, r, d, h, s, i, w}
 [Σ] : 12
2. Membuat tabel karakter dan frekuensi

Tabel 1. karakter dan frekuensi

Char	Frekuensi
M	5
A	3
P	7
U	2
Sp	1
E	4
R	2
D	4
H	2
S	2
I	1
W	3

3. Membangkitkan Kode *Punctured*

Tabel 2. String yang belum dikompresi

Char	Frekuensi	Binary	Bit	Frekuensi x Bit
m	6	01101101	8	48
a	5	01100001	8	40
p	2	01110000	8	16
u	2	01110101	8	16
Sp	2	00100000	8	16
e	2	01100101	8	16
r	1	01110010	8	8
d	1	01100100	8	8
h	2	01101000	8	16
s	2	01110011	8	16

Selanjutnya berikut tabel lanjutan string yang belum dikompresi:

Tabel 3. String yang belum dikompresi

Char	Frekuensi	Binary	Bit	Frekuensi x Bit
i	1	01101001	8	8
w	1	11101111	8	8
TOTAL BIT				216

4. Melakukan Proses kompresi

Tabel 4. Proses Kompresi dengan Algoritma *Punctured elias Code*

Char	frekuensi	Punctured Elias Code (P1)	Bit	Frekuensi x Bit
m	6	0	1	6
a	5	101	3	15
sp	2	10001	5	10
e	2	110101	6	12
h	2	110011	6	12
p	2	1001	4	8
s	2	1110111	7	14
u	2	11011	5	10
d	1	100001	6	6
i	1	1101001	7	7
r	1	1100101	7	7
w	1	11101101	8	8
Total				115

Dari table tersebut dapat dibentuk bit dan string sebelum dikompresi yaitu "mampu mempermudah mahasiswa" Menjadi string bit "0, 101,0,1001, 11011, 10001, 0, 110101, 0, 1001,110101, 1100101, 0, 11011, 100001, 101, 110011, 10001, 0, 101, 110011, 101, 1110111, 1101001, 1110111, 11101101, 101. Sebelum disimpan ke database akan di cek panjang string bit, jika ada sisa bagi n (1,2,3,4,5,6,7) maka dilakukan penambahan padding dan flagbits karena 115 tidak habis dibagi 8 dengan sisa n=4 maka akan ditambahkan padding 0 sebanyak 7-n+"1" dan flagbits sebanyak biner dari 9-n maka padding yang ditambahkan adalah 7-n+"1"="3+"1"="0001 kemudian flagbits yaitu biner dari 9-n=9-4=5=00000101. Agar habis dibagi 8 dan flag bits nya menjadi :01010100,11101110,00101101,01010011, 10101110, 01010110, 11100001,10111011, 10001010, 01100111, 01111011,11101001, 11101111, 11011011, 01000001, 00000101. Sehingga total panjang seluruhnya setelah penambahan padding dan flag bit adalah 115+12 = 127.

Dari hasil kompresi dengan *punctured elias code* (p1) diatas di dapatkan hasil dari kompresi mampu mempermudah mahasiswa menjadi

01010100= 11101110= 00101101= 01010011=
 10101110= 01010110= 11100001= 10111011=
 10001010= 01100111= 01111011= 11101001=
 11101111= 11011011= 01000001= 00000101=

Dan dapat dihitung kinerja kompresinya menurut parameter yang sudah ditentukan yaitu :

a. Ratio of compression (Rc)

$$Rc = \frac{\text{ukuran data sebelum dikompresi}}{\text{ukuran data setelah dikompresi}}$$

$$Rc = \frac{127}{115} = 1.1043$$

b. Compression ration (Cr)

$$Cr = \frac{\text{ukuran data setelah dikompresi}}{\text{ukuran data sebelum dikompresi}} \times 100\%$$

$$Cr = \frac{115}{127} \times 100\% = 90.55\%$$

c. Space Saving (SS)

SS = ukuran data sebelum dikompresi – ukuran data setelah dikompresi

$$SS = 127 - 115 = 12$$

$$SS = 127 - 115 = 12$$

$$SS = \frac{12}{127} \times 100\% = 9.45\%$$

3.2.1 Proses dekompresi berdasarkan algoritma *punctured elias code* (p1)

Pada proses dekompresi, langkah awalnya sistem akan membaca string hasil kompresi, setelah mendapatkan string bit hasil kompresi, tabel karakter dan kodenya, kemudian sistem melakukan proses dekompresi dengan mengembalikan string bit tersebut kebentuk semula berdasarkan tabel karakter dan tabel kode yang telah didapat dari hasil kompresi. Pada proses dekompresi hal yang dilakukan adalah menganalisa keseluruhan bit hasil dari kompresi sebelumnya. Adapun keseluruhan bit dan gabungan hasil kompresi yaitu :

"01010100,11101110,00101101,01010011, 10101110, 01010110, 11100001, 10111011, 10001010, 01100111, 01111011,11101001, 11101111, 11011011, 01000001, 00000101.

1. Baca nilai 8 bit akhir dari string bit dan ubah ke *decimal* maka 8 bit akhir dari string bit adalah 00000101 dengan nilai desimal 5, nyatakan dengan n.
2. Selanjutnya hapus bit pada akhir string bit sebanyak 7+n=7+5=12, maka 12 bit pada akhir string bit akan di hapus:" 01010100, 11101110, 00101101,

01010011, 10101110, 01010110, 11100001, 10111011, 10001010, 01100111, 01111011, 11101001, 11101111, 11011011, 01”.

- Lakukan pembacaan bit dari bit pertama sebelah kiri lalu tabel kode *punctured elias code*. Jika ditemukan bit yang sesuai dengan tabel kode *punctured elias code* maka ubah nilai string yang sesuai sehingga akan mendapatkan hasil tabel seperti dibawah ini.

Tabel 5. Hasil dekompresi menggunakan *punctured elias code*

<i>Punctured Elias Code</i>	Karakter
101	m
110101	a
101	m
0	p
1110111	u
1100011	Sp
1001	m
101	e
110101	m
1101001	p
11011	e
11101101	r
1001	m
110011	u
11011	d
11100111	a
10001	h
11101011	Sp
0	m
1110111	a
100001	h
0	a
1100101	s
1001	i

Tabel 6. Lanjutan Tabel Hasil Dekompresi Menggunakan *Punctured Elias Code*

<i>Punctured Elias Code</i>	Karakter
0	s
11101101	w
0	a

Berdasarkan hasil dekompresi diatas didapatkan karakter awal sebelum dikompresi sebagai berikut : “susah; sulit dipecahkan atau diselesaikan”.

4. KESIMPULAN

Berdasarkan dari penelitian yang telah dilakukan, maka hasil akhir dari penelitian tersebut dapat diambil beberapa kesimpulan dari pembahasan sebelumnya. Adapun kesimpulan tersebut ialah Prosedur pada aplikasi file transfer dalam mengirim sebuah file yaitu dengan memilih file yang akan dikompresi, setelah dipilih maka lakukan proses kompresi dan file hasil kompresi tersebut yang akan dikirim melalui jaringan, lalu proses dekompresi dilakukan oleh si penerima setelah file telah diterima. Penerapan algoritma *Punctured elias Code* untuk mengkompresi *file* yang akan dikirim telah berhasil membuat ukuran file menjadi lebih kecil dengan rasio 42% sehingga dapat memberikan proses transmisi yang cepat serta membantu mengurangi pemakaian paket data internet, dan Aplikasi *file transfer* dirancang dan dibangun dengan menggunakan *tools eclipse juno* dan menerapkan algoritma *Punctured elias Code* sehingga dapat memperkecil ukuran file yang akan dikirim.

REFERENCES

- R. D. Pratiwi, S. D. Nasution, and Fadlina, “Perancangan Aplikasi Kompresi File Teks Dengan Menerapkan Algoritma Fixed Length Binary Encoding (Flbe),” *Media Inform. Budidarma*, vol. 2, no. 1, pp. 10–14, 2018.
- M. R. Irliansyah, S. D. Nasution, and K. Ulfa, “Penerapan Metode Deflate Dan Algoritma Goldbach Codes Dalam Kompresi File Teks,” *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 1, no. 1, pp. 186–189, 2017.
- N. Dengen, “Perancangan Sistem Informasi Terpadu Pemerintah Daerah Kabupaten Paser,” vol. 4, no. 1, pp. 47–54, 2009.
- Marjito and G. Tesaria, “APLIKASI PENJUALAN ONLINE BERBASIS ANDROID (STUDI KASUS: DI TOKO HOAX MERCH),” *J. Comput. Bisnis*, vol. 10, no. 1, pp. 40–49, 2016.
- D. Salomon and G. Motta, *Handbook of Data Compression*. Springer, 2010.

- [6] J. Enterprise, *Rahasia Manajemen File*. Jakarta: PT. Elex Media Komputindo, 2010.
- [7] A. Nugroho, *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*. Andi, 2010.
- [8] R. A. S. M. Shalahuddin, *Rekayasa Perangkat Lunak*. Bandung, 2015.
- [9] A. Kadir, *REKAYASA PERANGKAT LUNA*. 2013.
- [10] M. Sadeli, *7 Jam Belajar Interaktif Visual Basic.net 2008 Untuk Orang Awam*. Palembang: Maxikom, 2009.
- [11] Huda, Asrul, and Noper Ardi. *Teknik Multimedia Dan Animasi*. UNP PRESS, 2021.
- [12] Ardiyanto, Deni. *Implementasi Metode Dwt (Discrete Wavelet Transform) Dan Metode Huffman Terhadap Kompresi Citra Digital Menggunakan Google Collaboratory*. Diss. University of Technology Yogyakarta, 2020.