

PEMANFAATAN ALGORITMA RICE CODES DAN GOLD BACH CODES UNTUK MENGOPTIMALKAN MEDIA PENYIMPANAN PADA MEMORI BERUPA FILE TEXT

AISHIYAH SAPUTRI LASWI¹

¹Prodi Teknik Informatika, Universitas Andi Djemma Palopo Sulawesi Selatan

Email: Aishiyah.rustam@gmail.com¹

ABSTRAK

Pemanfaatan algoritma Rice Codes dan GoldBach Codes sangat berpengaruh untuk melakukan pengoptimalan file text yang berkapasitas besar dari ukuran. Algoritma Rice Code dan GoldBach juga dapat dijadikan sebagai bahan evaluasi untuk pengujian file text yang bersektensi txt, doc dan lain-lain sebagainya. Sementara dari hasil pengujian kedua algoritma tersebut memiliki rentan nilai yang tidak jauh berbeda dengan menggunakan metode exponential. Walaupun dilihat dari hasil pengujiannya Algoritma Goldbach cocok digunakan untuk mengoptimalkan file text berdasarkan hasil yang diperoleh Algoritma Goldbach ada pada posisi pertama, dapat dilihat pada tabel 7 hasil dari perbandingan.

Kata Kunci: Algoritma rice codes, goldBach Codes mengoptimalkan File Text

I. PENDAHULUAN

Kapasitas memori merupakan hal yang sangat dibutuhkan karena mengingat kapasitas data yang semakin hari semakin bertambah dan akan menghambat proses penyimpanan apabila media penyimpanan tersebut tidak lagi mencukupi memori yang tersedia. Seperti halnya pada beberapa dunia industri, instansi pemerintahan baik negeri maupun swasta. Pada dasarnya membutuhkan media penyimpanan dengan skala besar sebagai wadah untuk menyimpan file-file pada perusahaan atau instansi terkait.

Cara yang dilakukan untuk mengoptimalkan penyimpanan data agar mengurangi penggunaan memori adalah dengan cara kompresi atau memperkecil ukuran data sehingga data yang tadinya memuat kapasitas yang cukup besar dapat dimaksimalkan. Maka media penyimpanan dapat mengolah data lebih efisien lagi untuk menghindari beberapa hal yang biasanya terjadi akibat memori penyimpanan yang tidak mencukupi kapasitas terkadang membuat data menjadi rusak bahkan hilang, sehingga penulis tertarik untuk melakukan penelitian dengan menggunakan dua algoritma sebagai tolak

ukur untuk mengoptimalkan skala penyimpanan yang menggunakan memori yang cukup banyak. Dalam hal ini file text adalah sebagai alat uji coba, bagaimana sehingga data yang ada pada text tidak berulang-ulang yang mengakibatkan pemborosan media penyimpanan.

Pengoptimalan ini adalah salah satu cara yang dipakai untuk mengubah kapasitas data lebih kecil dari ukuran sebelumnya. Manfaat pengoptimalan data ini sangat penting karena mengurangi pengulangan karakter yang digunakan dalam data sehingga menjadikan data lebih efisien dalam media penyimpanan

Adapun algoritma yang digunakan untuk mengoptimalkan data, yaitu algoritma Rice Codes dan Goldbach Codes. Dengan pemanfaatan kedua algoritma di atas sangat membantu dalam meminimalkan file text yang kapasitasnya cukup besar sehingga berakibat pada penggunaan media penyimpanan atau memori yang ada pada PC. File text yang telah dioptimalkan tidak membutuhkan waktu yang lama serta ruang penyimpanannya lebih sedikit. Adapun tujuan digunakannya kedua algoritma ini dengan implementasi metode eksponensial adalah untuk mengetahui tingkat keefisienan dalam mengoptimalkan text pada saat di kompresi

II. METODE

Metode yang digunakan dalam implementasi algoritma Rice codes dan GoldBach adalah metode exponential yang digunakan untuk mengambil keputusan dalam menentukan urutan-urutan penting pada sebuah karakter.

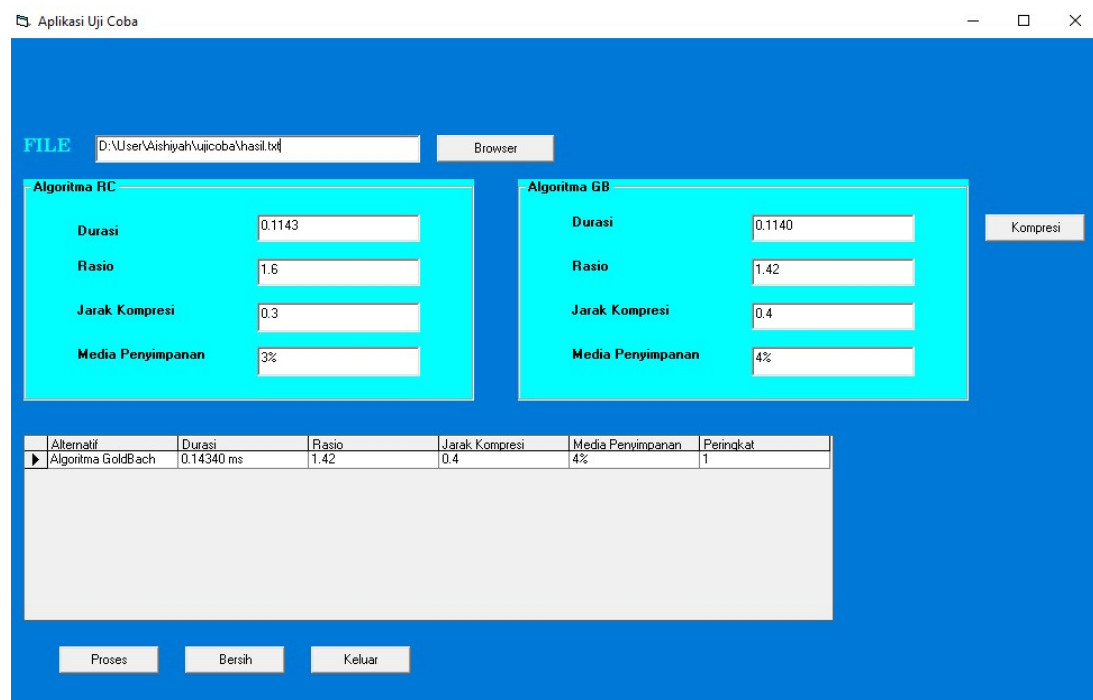
$$TotalNilai(TNi) = \sum (Vij) Bj \quad m_j=1 \dots\dots\dots(a)$$

Keterangan :

- Tni** = Total Nilai Alternatif Ke-
- Vij** = Derajat kepentingan relatif kriteria ke-j pada keputusan ke-i, yang dapat dinyatakan dengan skala ordinal (1,2,3,4,5)
- Bj** = Derajat kepentingan kriteria keputusan, yang dinyatakan dengan bobot
- M** = Jumlah kriteria keputusan
- N** = Jumlah pilihan keputusan
- J** = 1,2,3,... ; m = jumlah kriteria
- I** = 1,2,3,...; n = jumlah pilihan alternati

III. HASIL PENELITIAN

Pengoptimalan data berupa file text yang dilakukan dengan menggunakan algoritma rice code dan goldbach sebagai alat ukur untuk membandingkan kualitas kerja kedua algoritma tersebut. Apabila ada sebuah file yang berukuran besar dan terdapat perulangan karakter akan mengakibatkan pembengkakan media penyimpanan maka hal pertama yang dilakukan adalah dengan cara pemanfaatan data menggunakan algoritma untuk memperkecil ukuran data dari ukuran sebelumnya. seperti pada hasil implementasi yang ada di bawah ini:



Gambar 1. Hasil Perhitungan kompresi data

3.1 Pengoptimalan Text menggunakan Rice Codes

Gambar 1 diatas merupakan hasil kompresi data menggunakan rice code dan goldBach codes pada sebuah file text yang mengandung data *String* sebagai sample : **AISHIYAH SAPUTRI LASWI**. Adapun langkah-langkah yang dilakukan dalam pemampatan data menggunakan rice codes sebagai berikut:

1. Langkah Pertama : Menentukan urutan string dari text **AISHIYAH SAPUTRI LASWI.**
2. Langkah Dua : Menentukan nilai decimal dari Code Ascii berdasarkan data String dari text **AISHIYAH SAPUTRI LASWI.**
3. Langkah Tiga : Menentukan nilai Binner dari Ascii Binner berdasarkan data String **AISHIYAH SAPUTRI LASWI**
4. Langkah Empat : Menentukan nilai Binner dari Ascii Binner berdasarkan Ascii Binner **AISHIYAH SAPUTRI LASWI.**
5. Langkah Lima : Menentukan jumlah kemunculan karakter berdasarkan perhitungan frekuensi dari salah satu huruf data String **AISHIYAH SAPUTRI LASWI.**
6. Langkah Enam : Perkalian nilai Ascii Bit dan Frekuensi berdasarkan nilai Ascii bit dari kemunculan karakter yang terdapat dalam text **AISHIYAH SAPUTRI LASWI.**

Table 3. Text sebelum dioptimalisasikan

Σ	Kode ASCII	Bil BINER	BIT	FK	BIT x FK
A	77	01000001	6	5	30
I	73	01001001	6	1	6
S	83	01010011	6	1	6
H	72	01001000	6	1	6
Y	89	01011001	6	1	6
Sp	83	01010011	6	2	12
P	80	01010000	6	3	18
U	85	01010101	6	1	6
T	84	01010100	6	1	6
R	82	01010010	6	1	6
L	76	01001100	6	2	12
W	87	01010111	6	1	6
Jumlah Bit x Fk =120					

Data diatas merupakan data text yang belum dioptimasi menggunakan rice codes.

Table 4. Text yang sudah dioptimalisasikan

Σ	Rice Kode	BIT	FK	BIT x FK
A	000	3	5	15
Sp	010	3	2	6
I	011	3	1	3
H	1001	4	1	4
S	1011	4	1	4
P	0101	4	3	12
U	1000	4	1	4
Y	110011	6	1	6
R	010100	6	1	6
T	011011	6	1	6

L	110010	6	2	12
W	110001	6	1	6
Bit x FK= 84				

Berdasarkan data diatas jumlah karakter yang terdapat pada text adalah 20 karakter. Dengan kapasitas penyimpanan file text yang dibutuhkan pada media penyimpanan setelah dilakukan pengoptimalisasian menggunakan rice code adalah **58 bit**.

3.2 Pengoptimalan Text menggunakan Goldbach

AISHIYAH SAPUTRI LASWI merupakan text yang memiliki beberapa karakter yang berulang-ulang, sehingga untuk memperkecil ukuran text kembali dilakukan optimalisasi dengan menggunakan *goldbach* dengan sesuai dengan langkah-langkah dibawah ini:

1. Proses pembacaan selurh karakter yang terdapat pada text untuk mengetahui seberapa sering frekuensi kemuncul karakter tersebut.
2. Menentukan codeword untuk setiap karakter yang dengan menggunakan bilangan prima yang sudah ditentukan. Berikut text yang belum di kompresi:

Table 5. Text sebelum dioptimalkan ke goldbach

KARAKTER	FK	ASCII DECIMAL	ASCII BINER	BIT	BIT x FK
P	3	80	01010000	6	6
L	2	76	01001100	6	12
I	1	73	01001001	6	6
H	1	72	01001000	6	6
S	1	83	01010011	6	6
U	1	85	01010101	6	6
Y	1	89	01011001	6	6
R	1	82	01010010	6	6
T	1	84	01010100	6	6
L	2	76	01001100	6	12
W	1	87	01010111	6	12

Dan dibawah ini adalah table hasil optimasi dari table 5 diatas dengan menggunakan goldbach:

Table 6. Hasil Optimalisasi GoldBach

KARAKTER	KF	N	2(N+3)	BIL.PRIMA	CODE WORD	BIT	BIT x FK
A	5	1	8	3+5	11	2	10
P	3	2	10	3+7	0011	4	12
Sp	2	3	12	5+7	101	3	6
I	1	4	14	3+11	011	3	3
H	1	5	16	5+11	1001	4	4
S	1	6	18	7+11	0101	4	4
U	1	7	20	7+19	00101	5	5
Y	1	8	22	5+17	0000101	7	7
R	1	9	24	11+13	010001	6	6
T	1	10	26	7+19	00011	5	5
L	1	11	28	11+17	0010001	7	7
W	1	12	30	13+17	101010	6	6
Bit x FK = 75							

AISHIYAH SAPUTRI LASWI merupakan data String yang diperoleh dari bit
01000001 01001001 01010011 01001000 01001001 01011001 01000001 00100000
01010011 01000001 01010000 01010101 01010100 01010010 01001001 00100000
01001100 01000001 01010011 01010111 01001001.

Sebelum dilakukan pengoptimalan file text diatas dengan menggunakan kode ASCII(6)Bit adalah 20 x 6 bit =120 Bit untuk menampung sebanyak 20 karakter.

Sehingga berdasarkan hasil dari kedua algoritma diatas untuk melihat tingkat keakuratan keduanya kembali dilakukan pengujian dengan menggunakan metode exponential.berikut ini:

1. Menentukan Bagian Pengoptimalan

Rc= Ukuran data sebelumnya / Ukuran data sesudahnya

$$Rc = \frac{120}{84}$$

$$Rc = 1,42$$

GC= Ukuran data sebelumnya / Ukuran data sesudahnya

$$Rc = \frac{120}{75}$$

$$Rc = 1,6$$

2. Pengehamatan Ruang Penyimpanan

$$SS = 1 - Cr$$

$$\begin{aligned} RC &= 1 - 0,7 \\ &= 0,3 \\ &= 3\% \end{aligned}$$

$$\begin{aligned} GC &= 1 - 0,6 \\ &= 0,4 \\ &= 4\% \end{aligned}$$

Berdasarkan hasil perhitungan kedua algoritma

Table 7. Hasil Optimalisasi GoldBach

Alternatif	RS	SS	Peringkat
GoldBach	0,4	4%	1
RiceCodes	0,3	3%	2

Dari table diatas dapat kita lihat bahwa nilai tertinggi yang diperoleh dari kedua alternatif diatas akan ada urutan pertama. Dan berdasarkan hasil optimalisasi algoritma Gold Bach dapat digunakan dalam pemampatan data untuk mengurangi kapasitas penyimpanan berupa file text.

IV. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil pengujian dengan menggunakan metode exponential den 2 algoritma diatas dapat simpulkan bahwa.Penggunaan kedua algoritma tersebut sangat

berpengaruh untuk melakukan pengoptimalan file text dari ukuran sebelumnya untuk mengurangi pembengkakan pada memori atau media penyimpanan. Algoritma Rice Code dan GoldBach juga dapat dijadikan sebagai bahan evaluasi untuk pengujian file text yang bersektensi txt, doc dan lain-lain sebagainya.

Sementara dari hasil pengujian kedua algoritma tersebut memiliki rentan nilai yang tidak jauh berbeda dengan menggunakan metode exponential. Walaupun dilihat dari hasil pengujiannya Algoritma Goldbach cocok digunakan untuk mengoptimalkan file text yang memiliki banyak karakter yang berulang-ulang sehingga dapat memperkecil penggunaan memori pada PC atau perangkat keras. Berdasarkan hasil yang diperoleh Algoritma Goldbach ada pada posisi pertama, dapat dilihat pada table 7 hasil dari perbandingan.

4.2 Saran

Berdasarkan hasil pengoptimalan dari kedua algoritma diatas maka penulis memberikan saran untuk melakukan pengujian lebih lanjut dengan menggunakan perbandingan dengan beberapa algoritma yang memang khusus digunakan untuk mengoptimalkan media penyimpanan berupa file text. Bukan berarti kedua algoritma diatas tidak tepat hanya perlu dilakukan pengujian secara detail kembali dengan melibatkan beberapa algoritma yang memang khusus digunakan untuk mengkompresi data berupa file text.

Mengingat kebutuhan akan data hari ini semakin meningkat. Karena sebesar apapun kapasitas memori yang tersedia pada PC yang kita miliki saat ini itu tidak akan bisa menampung banyaknya kebutuhan data pada masa mendatang. Sehingga penulis memberikan saran untuk melakukan pengujian lebih lanjut dalam pengoptimalan media penyimpan guna mendapatkan hasil yang lebih baik dan akurat untuk kedepannya.

DAFTAR PUSTAKA

- Mhd.Citra Ali Pasaribu,2017. Analisis Kinerja Algoritma GlodBach Codes dengan Algoritma Variable Length Binary Endcoding dalam Kompresi File Teks. Universitas Sumatera Utara Medan.Program Studi Ilmu Komputer
- Mohammed J.Zaki dan Wagner Meira JR. 2014. *Data Mining And Analysis Fundamental Concepts And Algorithms*.CAMBRIDGE UNIVERSITY PRESS
- Surya Darma Nasution.2016 . “Perancangan Aplikasi Kompresi File Text Menggunakan Algoritma Gldbah Codes” Jurnal Ilmiah Infotek, Vol, 1, Februari 2016.
- Vincent Gaspersz.2005. *Planning And Inventory Control* Berdasarkan Pendekatan Sistem Terintegrasi MRP II dan JIT Menuju Manufakturung 21.PT. SUN,Jakarta.