JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 45-52
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Analisa Perbandingan Algortima Fibonacci Code Dan Rice Code Dalam Kompresi File Teks

Ratna Jelita
Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: rjelita100@gmail.com

Abstrak- Kebutuhan kapasitas media penyimpanan file yang semakin besar merupakan penyebab terjadinya kemunculan berbagai teknik kompresi. Sehingga dengan dilakukannya kompresi, data yang ukurannya besar akan menjadi kecil dapat menghemat kapasitas media penyimpanan dan juga mempercepat dalam mentrasnfer file. Kompresi bertujuan untuk mengurangi redudansi data menjadi sekecil mungkin. Untuk mengatasi permasalahan ini yaitu dengan mengetahui algoritma yang lebih efisien dalam melakukan kompresi file teks bereksistensi (*.txt) dan dalam penelitian ini juga membandingkan antara algoritma Fibonacci Code dan Rice Code berdasarkan parameter yang sudah ditentukan. Setiap algoritma memiliki hasil perbandingan yang berbeda baik itu dari algoritma Fibonacci Code maupun algoritma Rice Code. Seperti yang diketahui dari algoritma keduanya memiliki hasil nilai akhir dengan menggunakan Metode Perbandingan Eksponential, hasil nilai akhir algoritma Fibonacci Code adalah 5,577 sedangkan hasil nilai akhir algoritma Rice Code adalah 5,466. Dan semakin besar total nilai yang dihasilkan maka semakin tinggi jumlah usaha yang dilakukan algoritma tersebut. Berdasarkan analisa tersebut maka dapat ditentukan algoritma Fibonacci Code yang memiliki kinerja terbaik dalam melakukan kompresi file teks.

Kata Kunci: Teks; Kompresi; Eksponential; Fibonacci; Rice; Codes

Abstract- The need for increasingly large file storage media capacity is the cause of the emergence of various compression techniques. So, by compressing large data, it will become smaller, saving storage media capacity and also speeding up file transfers. Compression aims to reduce data redundancy to as little as possible. To overcome this problem, namely by knowing an algorithm that is more efficient in compressing existing text files (*.txt) and in this research we also compare the Fibonacci Code and Rice Code algorithms based on predetermined parameters. Each algorithm has different comparison results, both from the Fibonacci Code algorithm and the Rice Code algorithm. As is known, both algorithms have final value results using the Exponential Comparison Method, the final value result of the Fibonacci Code algorithm is 5.577 while the final value result of the Rice Code algorithm is 5.466. And the greater the total value produced, the higher the amount of effort carried out by the algorithm. Based on this analysis, the Fibonacci Code algorithm can be determined which has the best performance in compressing text files.

Keywords: Text; Compression; Exponential; Fibonacci; Rice; Codes

1. PENDAHULUAN

Pada masa pandemi ini, peran digital atau teknologi sangat diperlukan untuk kelangsungan kehidupan yang mengharuskan kegiatan tanpa kontak langsung. Komputer juga berfungsi untuk pengolah data seperti teks, angka, gambar dan video, *file* teks berukuran besar dapat menghambat kecepatan transmisi data dan penggunaan paket internet yang tinggi kemudian menyulitkan proses pengiriman jika dalam keadaan jaringan yang tidak stabil.

Kompresi ini diterapkan menggunakan algoritma *Fibonacci Code* dan *Rice Code* untuk *file* teks berekstensi *(*.txt)*. Alasan menggunakan algoritma kedua kompresi ini yang bertujuan untuk membandingan teknik kinerja algoritma yang paling baik dalam kompresi *file* teks. Tahapan dalam melakukan kompresi *file* terdapat dua teknik yaitu *lossless* dan *lossy*, untuk penelitian ini termasuk teknik kompresi *lossless*. Terkait kedua algoritma di atas terbilang cukup banyak digunakan dalam kompresi dan karena masih belum ada penelitian perbandingan kedua algoritma ini, maka penulis tertarik untuk melalukan penelitian ini.

2. METODOLOGI PENELITIAN

2.1 Kompresi

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 45-52
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Kompresi data merupakan bentuk data yang lebih padat atau mampat *(compact)* yang biasa dilakukan untuk mengubah ukuran *file* data tinggi atau besar menjadi relatif kecil. Kompresi data dapat juga mengurangi kelambatan penyimpanan data ketika volume data cukup besar dan data di dalamnya mengandung banyak fitur yang berulang. Dengan kompresi ini diharapkan dapat mengurangi ruang penyimpanan (mengurangi ukuran data).[1]

2.2 *File* Teks

File teks merupakan suatu proses yang bertujuan untuk mengurangi ukuran data dalam file tersebut dengan mengubah karakter-karakter menjadi kode-kode tertentu. *File* teks adalah *file* yang berisi informasi dalam bentuk teks. Data dari dokumen pengolah kata, angka-angka yang digunakan dalam perhitungan, nama dan alamat dalam database merupakan contoh masukan data teks yang terdiri dari karakter, angka, dan tanda baca.[2]

2.3 Algoritma *Fibonacci Code*

Fibonacci Code merupakan kode *length* variabel yang berkode pendek jika bilangan bulat-nya terkecil. Kode ini diakhiri dengan dua bit angka satu atau membubuhkan angka "1" di kanan dari *Fibonacci codes* sementara dan untuk bit yang ditentukan diperoleh dari jumlah nilai *fibonacci* yang telah disesuaikan (tidak termasuk bit terakhir) yang merupakan proses akhir atau proses *encoding*, Sedangkan menghilangkan angka "1" bagian kanan *Fibonacci Codes* tersebut merupakan proses *decoding*.[3]

Langkah-langkah dalam pembuatan sebuah kode *Fibonacci* adalah sebagai berikut:

1. Tentukan sebuah bilangan bulat positif n yang lebih besar atau sama dengan 1.
2. Carilah bilangan *Fibonacci* terbesar f yang lebih kecil atau sama dengan n, kurangkan nilai n dari f dan catat sisa pengurangan nilai n dengan f.
3. Jika bilangan yang dikurangkan adalah bilangan yang terdapat dalam deret *Fibonacci* F(i), tambahkan angka "1" dalam kode *Fibonacci* sementara.
4. Ulangi langkah ke-2, tukar nilai n dengan sisa pengurangan nilai n dengan f sampai sisa pengurangan nilai n dengan f adalah 0.
5. Tambahkan angka "1" pada posisi paling kanan kode *Fibonacci* yang akan dibentuk.

2.4 Algoritma *Rice Code*

Dalam algoritma *Rice Code*, yang mempunyai nilai k artinya adalah banyaknya angka 1 pada suffix dari kode terkompresi. Dalam proses *encoding*, dilakukan pemisahan pada prefix dan suffix.[4] Ketika proses *decoding*, membaca sign bit dan lompat ke angka 0 pertama sebelah kiri, yang mana akan berlanjut kembali untuk penambahan bit pada k selanjutnya. Untuk nilai k dalam proses kompresi menggunakan *Rice Code*, dapat dilihat pada tabel dibawah ini.[6]

Tabel 1. Tabel Nilai K [1]

N	K=1	K=2	K=3	K=4
0	0|	0|0	0|00	0|000
1	10|	0|1	0|01	0|001
2	110|1	10|1	0|10	0|010
3	110|1	10|1	0|11	0|011
4	1110|00	110|00	10|00	0|100
5	1110|01	110|01	10|01	0|101

2.5 Metode Perbandingan *Eksponential*

Metode *Exponential* adalah satu metode perbandingan untuk menentukan urutan pada alternatif keputusan dengan kriteria terbaik. Berikut merupakan rumus dari metode *Exponential* sebagai berikut:

$$\text{Total Nilai } (\text{TN}_\text{i}) = \sum_{j=1}^{m} \text{RK}_{ij})^{\text{TKKj}}$$

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

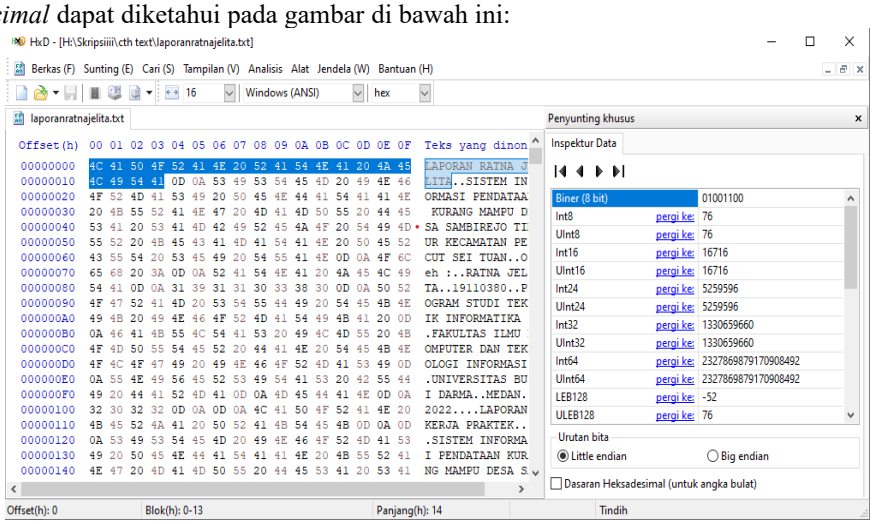
JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 45-52
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Keterangan :
TNi = Total nilai ke-i.
RKij = Derajat kepentingan relatif kriteria ke-j untuk keputusan ke-i, yang dapat dinyatakan dengan skala ordinal (1,2,3,4,5).
TKKj = Derajat kepentingan kriteria keputusan ke-j; TKKj >0;
m = Jumlah kriteria keputusan.
n = Jumlah pilihan keputusan.
j = 1,2,3,.....m; m = jumlah kriteria.
i = 1,2,3,.....n; n = jumlah pilihan alternatif.

3. HASIL DAN PEMBAHASAN

Analisis masalah yaitu proses mengembangkan struktur yang dirancang dan menghitung proses untuk menentukan pada kompresi *file* teks. Permasalahan yang diteliti dalam penelitian ini ialah tahapan proses perbandingan algoritma *Fibonacci Code* dan algoritma *Rice Code* agar mengidentifikasi algoritma terbaik diantara algoritma - algoritma ini dengan penggunaan teknik kompresian *lossless*, dan menggunakan parameter seperti *ratio of compressions, compresions ratio, space saving* serta tujuan kompresi dari *file* ini ialah dapat memperkecil ukuran dari *file* teks.

Untuk mendapatkan nilai *Hexadecimal* dari *file* "LAPORAN RATNA JELITA.txt" diperlukan *software* untuk mengetahui nilai *hexadecimal* dalam *file* teks ini. Berikut nilai *Hexadecimal* dapat diketahui pada gambar di bawah ini:



Gambar 1. Nilai *Hexadecimal File* Teks

3.1 Analisa Penerapan Algoritma *Fibonacci Code* dan *Rice Code*

1. Analisa Proses Kompresi Algoritma *Fibonacci Code*

Nilai *Hexadecimal* dari *file* LAPORAN RATNA JELITA.txt yaitu **4C 41 50 4F 52 41 4E 20 52 41 54 4E 41 20 4A 45 4C 49 54 41.**

Tabel 2. Frekuensi Data Sebelum Dikompresi

Biner	*Hexadesimal*	Bit	Frekuensi	Bit x Frek
01000001	41	8	5	40
01001100	4C	8	2	16
01010010	52	8	2	16
01001110	4E	8	2	16
01010100	54	8	2	16
00100000	20	8	2	16

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 45-52
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Biner	*Hexadesimal*	Bit	Frekuensi	Bit x Frek
01010000	50	8	1	8
01001111	4F	8	1	8
01001010	4A	8	1	8
01000101	45	8	1	8
01001001	49	8	1	8
			Total	160 bit

Berdasarkan tabel di atas nilai *hexadesimal* dan *string* yang diambil sebanyak 20 bilangan *character*. Sehingga mempunyai total nilai bit 160 bit. Selanjutnya pengurutan dengan cara *descending* dan seterusnya masuk ke fase kompresian.[7]

Tabel 3. Hasil Kompresi *Fibonacci Code*

Hexa	Frek	Pengurangan nilai n	Kode *Fibonacci* Sementara	*Codeword*	Bit	Frek x Bit
41	5	1-1	1	11	2	10
4C	2	2-2	01	011	3	6
52	2	3-3	001	0011	4	8
4E	2	4-3-1	101	1011	4	8
54	2	5-5	0001	00011	5	10
20	2	6-5-1	1001	10011	5	10
50	1	7-5-2	0101	01011	5	5
4F	1	8-8	00001	000011	6	6
4A	1	9-8-1	10001	100011	6	6
45	1	10-8-2	01001	010011	6	6
49	1	11-8-3	00101	001011	6	6
			Total			81 Bit

Total *string* sesudah di kompresikan yaitu 81 bit. Berikutnya penyusunan *character* sebelum dikompresi "4C 41 50 4F 52 41 4E 20 52 41 54 4E 41 20 4A 45 4C 49 54 41". *String* bit yang dihasilkan *character* sesuai *Codeword* yaitu,
"01111010110000110011111011100110011110001110111110011100011010011011001011000111 1".

Tabel 4. Penambahan *Padding* dan *Flagging*

Padding	*Flagging*
$81\ mod\ 8 = 1 = 1$ $7 - n + "1"$ $7 - 1 + "1" =$ 0000001	$9 - n$ $9 - 1 = 8 = 00001000$

Setelah dilakukan proses *padding* dan *flagging,* maka total bit menjadi **96** bit. "01111010110000110011111011100110011110001110111110011100011010011011001011000111 1000000100001000".

2. Analisa Proses Kompresi Algoritma *Rice Code*

Nilai *Hexadecimal* dari *file* LAPORAN RATNA JELITA.txt yaitu 4C 41 50 4F 52 41 4E 20 52 41 54 4E 41 20 4A 45 4C 49 54 41.

Tabel 5. Frekuensi Data Sebelum Dikompresi

Biner	*Hexadesimal*	Bit	Frekuensi	Bit x Frek
01000001	41	8	5	40

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 45-52
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Biner	*Hexadesimal*	Bit	Frekuensi	Bit x Frek
01001100	4C	8	2	16
01010010	52	8	2	16
01001110	4E	8	2	16
01010100	54	8	2	16
00100000	20	8	2	16
01010000	50	8	1	8
01001111	4F	8	1	8
01001010	4A	8	1	8
01000101	45	8	1	8
01001001	49	8	1	8
			Total	160 bit

Tahap kompresian tersebut dengan total nilai bit dapat didapatkan sesuai dengan nilai-nilai *codeword* atau dengan perkalian nilai-nilai *codeword* pada frekuensi.[5]

Tabel 6. Hasil Kompresi *Rice Code*

No	*Hexadecimal*	Frekuensi	*Codeword Rice Code*	Bit	Frek X Bit
1	41	5	00 00	4	20
2	4C	2	00 01	4	8
3	52	2	00 10	4	8
4	4E	2	00 11	4	8
5	54	2	010 00	5	10
6	20	2	010 01	5	10
7	50	1	010 10	5	5
8	4F	1	010 11	5	5
9	4A	1	011 000	6	6
10	45	1	011 001	6	6
11	49	1	011 011	6	6
		Total			92

Total *string* sesudah di kompresikan yaitu 92 bit. Berikutnya penyusunan *character* sebelum dikompresi “4C 41 50 4F 52 41 4E 20 52 41 54 4E 41 20 4A 45 4C 49 54 41”. *String* bit yang dihasilkan *character* sesuai *Codeword* yaitu,
“0001 0000 01010 01011 0010 0000 0011 01001 0010 0000 01000 0011 0000 01001 011000 011001 0001 011011 01000 0000”.

Tabel 7. Penambahan *Padding* dan *Flagging*

Padding	*Flagging*
$92\ mod\ 8 = 4 = 4$ $7 - n + \text{"1"}$ $7 - 4 + \text{"1"} = 0001$	$9 - n$ $9 - 4 = 8 = 00000101$

Setelah dilakukan proses *padding* dan *flagging*, maka total bit menjadi **104** bit. “000100000101001011001000000011010010010000001000001100000100101100001100100010 1101101000000000100000101”.

3.1.1 Penerapan Metode Perbandingan *Exponential*

a. Menentukan kriteria pengujian algoritma

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 45-52
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

Dalam proses perbandingan dari algoritma-algoritma tersebut, diperlukan kriteria-kriteria untuk menganalisa hasilnya. Adapun jenis kriterianya yaitu *Cr, Rc* dan *Ss*.

b. Menentukan bobot kriteria

Penulis memberikan bobot berdasarkan tingkat pengaruh dalam kriteria, dapat diketahui dibawah ini.

Tabel 8. Pembobotan Kriteria

Kriteria	Persentase Pengaruh Kriteria	Bobot Range (0-1)
Ratio of Compression	10%	0.1
Compression Ratio	10%	0.1
Space Saving	30%	0.3

Setelah menentukan bobot kriterianya, selanjutnya yaitu perhitungan metode eksponential, hasilnya berikut ini.

Tabel 9. Hasil Perbandingan Metode *Exponential*

Kriteria	Bobot	Algoritma Fibonacci Code	Algoritma Rice Code
RC	0.1	1,6	1,53
CR	0.1	60	65
SS	0.3	40	35
Total Nilai	0.5	5,577	5,466

Nilai Algoritma *Fibonacci Code*
$= (1,6)^{0,1} + (60)^{0,1} + (40)^{0,3}$
$= 1,048 + 1,505 + 3,024$
$= 5,577$
Nilai Algoritma *Rice Code*
$= (1,53)^{0,1} + (65)^{0,1} + (35)^{0,3}$
$= 1,043 + 1,518 + 2,905$
$= 5,466$

Kemudian perolehan alternatif telah didapatkan, Hasil dari keputusan prioritas ditunjukkan dalam *table* berikut ini:

Tabel 10. Prioritas Keputusan

Alternatif	Total Nilai	Rangking
Algoritma *Rice Code*	5,466	1
Algoritma *Fibonacci Code*	5,577	2

Terlihat diatas total-total dan alternatif yang rendah yang menjadi urutan pertama, sebab semakin rendah nilai total yang didaptkan maka teknik kompresi *file* teks akan semakin baik. Berdasarkan analisa diatas, algoritma *Rice Code* dinyatakan algoritma yang terbaik dalam mengkompresi *file* teks.

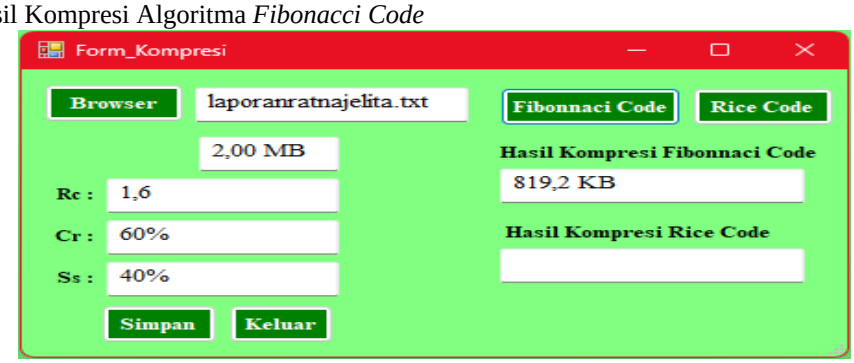
3.2 Implementasi

Impelementasi program ialah langkah pengujian suatu sistem yang telah dibuat sebelumnya. Pada tahap ini akan dilakukan pengujian pada *file* dengan ekstensi *.txt.

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

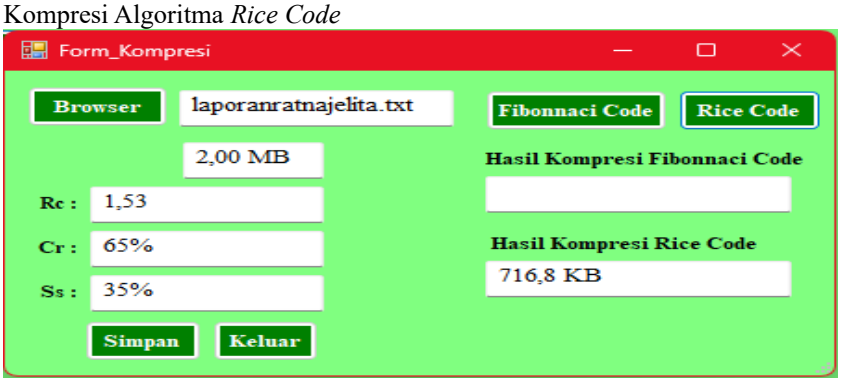
JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 45-52
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

a. Hasil Kompresi Algoritma *Fibonacci Code*



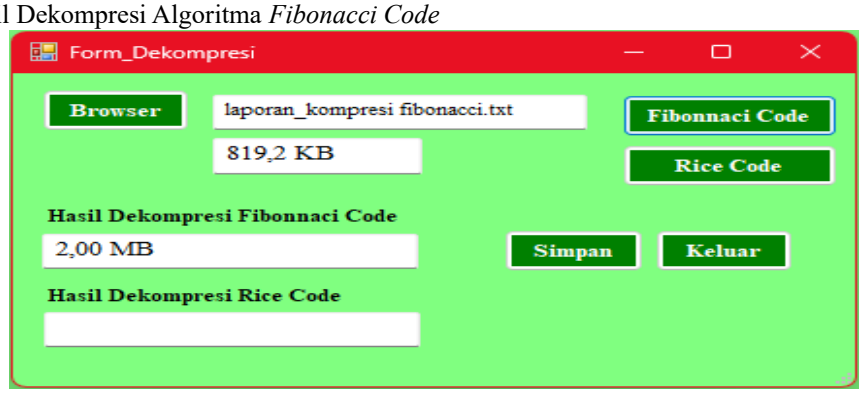
Gambar 2. Hasil Proses Kompresi Menggunakan *Fibonacci Code*

b. Hasil Kompresi Algoritma *Rice Code*



Gambar 3. Hasil Proses Kompresi Menggunakan *Rice Code*

c. Hasil Dekompresi Algoritma *Fibonacci Code*



Gambar 4. Hasil Proses Dekompresi Menggunakan *Fibonacci Code*

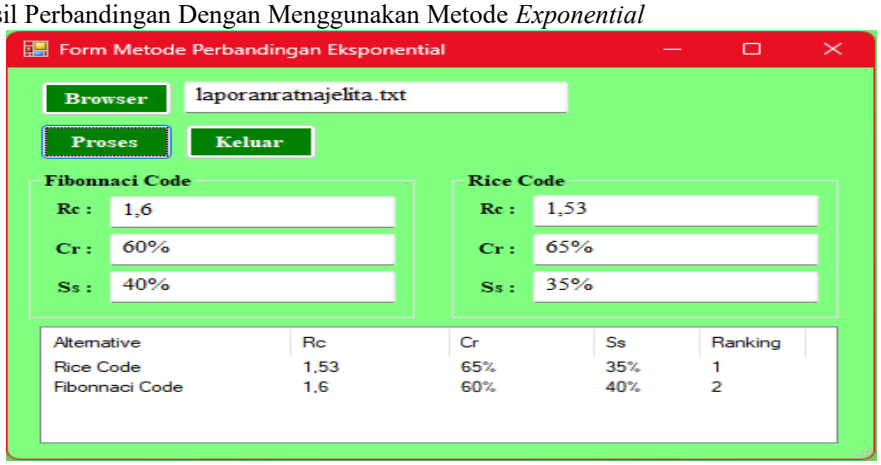
d. Hasil Dekompresi Algoritma *Rice Code*

Gambar 5. Hasil Proses Dekompresi Menggunakan *Rice Code*

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**

JURNAL MEDIA INFORMATIKA [JUMIN]
Volume 5, Nomor 1, Desember 2023, Page 45-52
ISSN 2808-005X (media online)
Available Online at http://ejournal.sisfokomtek.org/index.php/jumin

e. Hasil Perbandingan Dengan Menggunakan Metode *Exponential*



Gambar 6. Hasil Perbandingan Menggunakan Metode Perbandingan *Exponential*

4. KESIMPULAN

Dengan implementasi algoritma *Fibonacci Code* dengan nilai awal 2,00 MB dikompresi menjadi 0,8 MB atau 819,2 KB dan algoritma *Rice Code* dengan nilai awal 2,00 MB dikompresi menjadi 0,7 MB atau 716,8 KB kemudian untuk menentukan yang terbaik menggunakan eksponential dan nilai akhir algoritma *Rice Code* lebih baik dibandingkan *Fibonacci Code.* Algoritma *Fibonacci Code* dan *Rice Code* mempunyai hasil perbandingan kompresi *file* teks yang berbeda, metode perbandingan *eksponential* digunakan untuk menghitung *Ratio Of Compressions* (RC), *Compressions Ratio* (RC), *Space Saving* (SS). Hasil Algoritma *Fibonacci Code* sebesar 5.577, sedangkan hasil akhir Algoritma *Rice Code* sebesar 5.466. Jadi semakin rendah nilai keseluruhan yang diperoleh maka semakin baik. Berdasarkan analisa tersebut, Algoritma *Rice Code* merupakan algoritma terbaik untuk melakukan kompresi *file* teks. Aplikasi dirancang untuk mengkompresi *file* teks dengan penerapan algoritma *Fibonacci Code* dan algoritma *Rice Code*, kemudian memberikan hasil kompresian dan dekompresi berdasarkan algoritma *Fibonacci Code* dan algoritma *Rice Code*, dalam teknik kompresi ini menggunakan *lossless* yang artinya mengembalikan *file* secara sempurna seperti *file* aslinya.

REFERENCES

[1] M. A. Latif, S. D. Nasution, and Pristiwanto, "Analisa Perbandingan Algoritma Rice Codes Dengan Algoritma Goldbach Codes Pada Kompresi File Text Menggunakan Metode Exponential," *Maj. Ilm. INTI (Informasi dan Teknol. Ilmiah)*, vol. 13, no. 1, pp. 28–33, 2018.

[2] J. P. Informatika, S. D. Nasution, I. Pendahuluan, R. L. Encoding, R. Codes, and A. Kompresi, "PERANCANGAN APLIKASI KOMPRESI FILE TEKS DENGAN," vol. 7, pp. 219–222, 2018.

[3] J. Martina and B. Panjaitan, "Penerapan Algoritma Fibonacci Codes Pada Kompresi Aplikasi Audio Mp3 Berbasis Dekstop," vol. 1, no. 1, pp. 27–33, 2021.

[4] A. N. Purnama, "Implementasi Algoritma Rice Code dan Ternary Comma Code Pada Kompresi File PDF," vol. 6, no. November, pp. 34–42, 2022, doi: 10.30865/komik.v6i1.5787.

[5] N. Astuti Hasibuan, "Analisa Perbandingan Algoritma HuffmanDengan Rice Code Dalam Kompresi File Video," *Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. 1, 2022, doi: 10.30865/komik.v6i1.5751.

[6] R. Silaban, "Penerapan Metode Fibonacci Code Dalam Mengkompresi File Video," vol. 6, no. November, pp. 208–215, 2022, doi: 10.30865/komik.v6i1.5679.

[7] P. Fitria, "Penerapan Algoritma Rice Codes Pada Aplikasi Kompresi File Gambar," *J. Comput. Syst. Informatics*, vol. 1, no. 3, pp. 158–165, 2020.

Submitted: **17/12/2023**; Accepted: **26/12/2023**; Published: **29/12/2023**