

Penerapan Algoritma Gopala Hemachandra Code 2 (GH2(N)) Untuk Kompresi Record pada Database MYSQL

Diana Novita

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: diananov@gmail.com

Abstrak—Dalam perkembangan teknologi yang semakin pesat, sebuah media penyimpanan data sudah menjadi kewajiban yang harus dimiliki oleh setiap manusia yang selalu berhubungan dengan komputer, baik itu penyimpanan data internal maupun penyimpanan data eksternal. Sama halnya seperti basis data (database) sudah menjadi sebuah kebutuhan pokok bagi setiap perorangan atau organisasi untuk menjalankan suatu kegiatan yang bersifat komputerisasi. Basis data (database) merupakan kumpulan dari data-data teks yang membentuk suatu file yang saling berhubungan dengan tata cara yang tertentu untuk membentuk data baru atau informasi baru. Dari permasalahan tersebut maka ditemukan suatu teknik agar dapat menyelesaikan segala permasalahan yang ditemukan sebelumnya. Adapun teknik tersebut dinamakan teknik kompresi. adalah proses mengubah data yang asli (mentah) menjadi suatu aliran data lain dengan format berbeda yang memiliki ukuran lebih kecil atau proses pengubahan sekumpulan data menjadi bentuk kode dengan tujuan untuk menghemat ruang penyimpanan. Pada kasus ini yang penulis gunakan dalam kompresi adalah algoritma Gopala-Hemachandra Code 2 (GH-2(n)) untuk kompresi file record. Maka dari penelitian ini akan didapatkan berapa presentase dari data sebelum dilakukan proses kompresi dengan yang sudah didekompresi. Sehingga para pengguna juga dapat lebih menghemat ruang penyimpanan tanpa kehilangan informasi, selain itu waktu yang digunakan dalam proses transmisi akan menjadi lebih sedikit.

Kata Kunci: Kompresi Record; Algoritma Gopala Hemachandra Code 2(GH2(n))

Abstract—In the rapid development of technology, a data storage medium has become an obligation that must be owned by every human being who is always in contact with computers, be it internal data storage or external data storage. Similarly, a database has become a basic need for every individual or organization to carry out a computerized activity. A database is a collection of text data that forms a file that is interconnected in a certain way to form new data or new information. From these problems, a technique was found in order to solve all the problems found previously. The technique is called the compression technique. is the process of converting the original (raw) data into another data stream with a different format that has a smaller size or the process of converting a set of data into code form with the aim of saving storage space. In this case the author uses in compression is the Gopala-Hemachandra Code 2 (GH-2(n)) algorithm for file record compression. So from this research will be obtained what percentage of the data before the compression process is done with the one that has been decompressed. So that users can also save more storage space without losing information, besides that the time used in the transmission process will be less.

Keywords: Record Compression; Gopala Hemachandra Code Algorithm 2(GH2(n))

1. PENDAHULUAN

Pesatnya perkembangan teknologi membawa pada zaman yang serba canggih, terutama dalam bidang pengolahan data dan sistem informasi. Penggunaan database dalam suatu sistem informasi bisa di katakan sangat penting, karena tanpa adanya database maka data yang akan diolah tidak dapat tersimpan. Database yang saat ini banyak digunakan adalah MySQL, karena database MySQL merupakan software database yang gratis dan mudah digunakan. Banyak data atau record yang tersimpan di dalam suatu database membuat ukuran databasenya menjadi besar. Selain ukuran database nya yang menjadi besar, banyak record juga menyebabkan pembacaan record nya menjadi lebih lama. Banyak cara yang dapat digunakan untuk mengoptimalkan kecepatan pembacaan record pada database, tetapi cara-cara tersebut tidak mengurangi ukuran database dalam menggunakan ruang penyimpanan pada hardisk atau Solid State Drive (SSD). Salah satu solusi yang dapat diberikan untuk mengurangi ukuran database dan yang dapat mengoptimalkan database yaitu menggunakan kompresi.

Saat ini ada banyak aplikasi kompresi, tetapi tidak ada yang dapat mengompres catatan basis data. Saat ini sudah banyak algoritma kompresi dan hasil yang didapat juga bervariasi. Salah satu algoritma kompresi yang dapat digunakan adalah GopalaHemachandra Code 2 (GH2(n)). Algoritma GopalaHemachandra Code 2 (GH2 (n)) adalah algoritma kompresi lossless, yang berarti bahwa ukuran file tidak akan mengalami kehilangan data selama proses dekompresi.

Berdasarkan penelitian yang dilakukan pada tahun 2020 oleh Sinta M. Panjaitan, Surya Darma Nasution, Bister Purba berjudul "Penerapan Algoritma Gopala Hemachandra Code2 (GH2(n)) dalam Kompresi File Audio". Disimpulkan bahwa algoritma GopalaHemachandra Code 2 (GH 2 (n)) telah diterapkan dan telah terbukti bahwa file audio yang besar dapat dikompresi ke ukuran yang lebih kecil[1]. Penelitian yang dilakukan oleh Ratiya Tari, dkk pada tahun 2021 yang berjudul "Penerapan Algoritma Arithmetic Coding Untuk Mengkompresi Record Database Pada Aplikasi Ensiklopedia Flora Berbasis Android". Disimpulkan bahwa Pengukuran kinerja hasil kompresi dengan menerapkan algoritma arithmetic coding dalam mengkompresi record database aplikasi ensiklopedia flora yaitu menghasilkan Ratio of compression (2,2), Compression Ratio (45%), dan Redudancy (55%)[2]. Penelitian lainnya dengan Penerapan Algoritma Lzy Untuk Mengkompresi Record Database Mysql dilakukan oleh Nuraini Siregar pada tahun 2023[3].

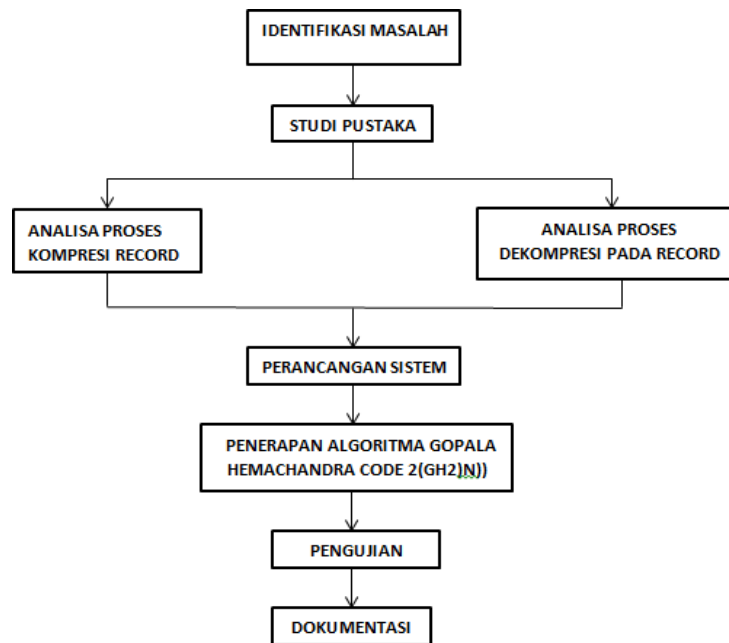
Mengompresi entri database dalam database bertujuan untuk mengurangi jumlah entri duplikat (identik) yang disimpan dalam database. Tentu saja, ini mengurangi jumlah karakter input yang akan disimpan dan secara otomatis mengurangi ukuran *file* di memori komputer.

Berdasarkan hasil penelitian sebelumnya yang telah dibahas di atas, algoritma Gopala Hemachandra Code 2 (GH 2(n)) merupakan algoritma yang mampu mengompresi dengan menghilangkan bit yang sama (*redundancy*) untuk menghasilkan bit yang lebih kecil dari ukurannya. Kompresi record dalam database MySQL harus menjadi solusi untuk mengompresi ukuran database.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Pada penelitian akan dilakukan analisa dan perancangan lunak pengkompresian record dengan menggunakan algoritma Gopala-Hemachandra code 2(GH-2(n)). Tahapan yang akan dilakukan dapat dilihat dalam kerangka kerja yang ada pada gambar 1.



Gambar 1. Tahapan Penelitian

Pada gambar 1 dapat dilihat tahapan pelaksanaan penelitian yang akan dilakukan, berikut penjelasan dari tahapan penelitian:

1. Identifikasi Masalah
Yaitu salah satu proses penelitian yang yang boleh dikatakan paling penting di antara proses lainnya.
2. Studi Pustaka
Suatu studi yang digunakan dalam mengumpulkan informasi dan data dengan bantuan berbagai macam material yang ada di perpustakaan seperti dokumen, buku, majalah, kisah-kisah sejarah.
3. Analisa proses Kompresi pada Record
Adalah teknik untuk mengecilkan data sehingga dapat diperoleh file dengan ukuran yang lebih kecil daripada ukuran file aslinya. kompresi dibutuhkan untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi/pertukaran data.
4. Analisa peroses Dekompresi Record
Adalah proses pembelian data yang sudah dikompres menjadi data seperti semula.
5. Perancangan Sistem
Adalah tahapan dari sebuah proses pendefinisian kebutuhan - kebutuhan dari siklus perkembangan sistem baru atau sistem yang akan dibekuk.
6. Penerapan algoritma GopalaHemachandra Code 2 (GH2) n))
Merupakan teknik kompresi lossless yang dapat mereduksi data berdasarkan frekuensi karakter pada objek yang akan dikompresi. Menggunakan algoritma GopalaHemachandra Code 2 (GH2(n)) didasarkan pada karakter yang paling umum dan memiliki bit paling sedikit menurut GopalaHemachandra Code 2 (GH2(n)), sedangkan karakter yang paling sedikit memiliki jumlah panjang bit terpanjang.
7. Pengujian
Peneliti menyimpulkan suatu kesimpulan berdasarkan hasil pengujian melalui perhitungan dan perancangan aplikasi yang dilakukan. Apakah hasil yang diperoleh sesuai dengan maksud dan tujuan penelitian dalam menyelesaikan suatu permasalahan yang dibuat.
8. Dokumentasi

Adalah suatu cara yang digunakan untuk memperoleh data dan informasi dalam bentuk buku, arsip, dokumen, tulisan angka, gambar yang berupa laporan serta keterangan yang dapat mendukung penelitian.

2.1 Kompresi Data

Proses kompresi mengurangi ukuran data untuk menghasilkan representasi digital yang padat atau kompak tetapi masih dapat mewakili jumlah informasi yang terkandung dalam data. Dalam literatur, istilah kompresi sering disebut sebagai source encoding, kompresi data, kompresi bandwidth, dan kompresi sinyal. Data dan informasi adalah dua hal yang berbeda. Informasi yang terkandung dalam data. Namun, tidak semua bagian data terkait dengan informasi, atau ada bagian data yang berulang dalam data, yang dikenal sebagai data redundansi. Kompresi data bertujuan untuk mengurangi redundansi data sehingga ukuran data menjadi lebih kecil dan ringan dalam proses transmisi. Kompresi data juga dapat diartikan sebagai proses pengurangan ukuran data dengan mengubah suatu kumpulan data menjadi kumpulan kode, yang dapat menghemat ruang dan waktu untuk transfer data[4][5][6][7][8][9].

Pada suatu teknik yang digunakan dalam proses kompresi Record terdapat beberapa faktor atau variabel yang biasa digunakan untuk menganalisa kualitas dari suatu teknik kompresi data tersebut yaitu[10][11][12][13][14]:

1. Ratio of Compression (RC)

Ratio of Compression (RC) adalah nilai perbandingan antara ukuran bit data sebelum di kompresi dengan ukuran bit data yang telah dikompresi.

Rumus:

$$RC = \frac{\text{Jumlah bit sebelum dikompres}}{\text{Jumlah bit sudah dikompres}} \quad (1)$$

2. Compression Ratio (CR)

Compression Ratio (CR) adalah persentase perbandingan antara data yang sudah dikompresi dengan data yang belum dikompresi.

Rumus:

$$CR = \frac{\text{Jumlah bit sebelum dikompres}}{\text{Jumlah bit sudah dikompres}} * 100\% \quad (2)$$

3. Space Savings (SS)

Space Saving (SS) adalah selisih antara data yang belum dikompresi dengan besar data yang dikompresi.

Rumus:

$$SS = 100\% - CR \quad (3)$$

4. Time Compression

Time compression adalah waktu yang dibutuhkan untuk melakukan proses kompresi dan dekompresi. Semakin kecil waktu yang diperoleh maka semakin efisien metode yang digunakan dalam proses kompresi dan dekompresi itu.

2.2 Gopala Hemachandra Code 2 (GH2(n))

Gopala Hemachandra Code merupakan variasi dari deret Fibonacci code kedua yang mendefinisikan urutan Gopala Hemachandra Code dengan nilai $b = 1 - a$ yang dapat dirumuskan sebagai berikut[15][1]:

$$V Fa(0) = a$$

$$(a \in \mathbb{Z})$$

$$V Fa(1) = 1 - a$$

Dan untuk $n \geq 2$

$$V Fa(n) = V Fa(n - 1) + V Fa(n - 2)$$

Menghasilkan:

$$V Fa(k) = (a, 1 - a, 1, 2 - a, 3 - a, 5 - 2a, 8 - 3a, 13 - 5a, 21 - 8a, \dots)$$

Keterangan:

$V Fa$ = Variasi Angka

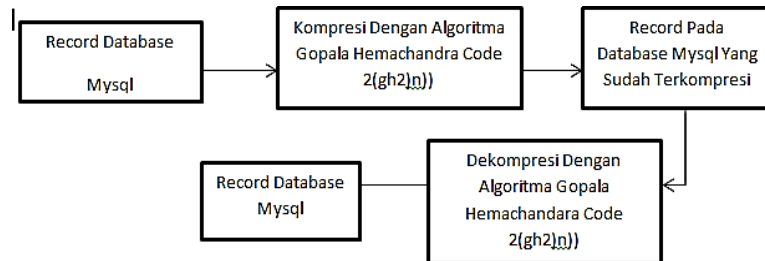
Fibonaccissa = Suku Pertama

3. HASIL DAN PEMBAHASAN

3.1 Analisa dan Penerapan Metode

Analisa yang dilakukan dalam penelitian ini adalah menerapkan algoritma Gopala-Hemachandra Code 2 (GH-2(n)), yang digunakan untuk mengompresi file record. Algoritma Gopala-Hemachandra Code 2 (GH2(n)) merupakan teknik kompresi lossless yang dapat mengecilkan data berdasarkan character rate dari objek yang dikompresi. Menggunakan

algoritma GopalaHemachandra Code 2 (GH2(n)) didasarkan pada karakter yang paling umum dan memiliki bit paling sedikit menurut GopalaHemachandra Code 2 (GH2(n)), sedangkan karakter yang paling sedikit memiliki panjang yang lebih panjang jumlah bit. Pada proses kompresi, algoritma GopalaHemachandra Code 2 (GH2(n)) diterapkan pada file record, yang memiliki ukuran besar. Rekam kompresi adalah proses mengompresi data menjadi file yang lebih kecil dari aslinya. Tahapan analisa terhadap suatu sistem yang dilakukan sebelum tahapan perancangan dilakukan. Langkah-langkah prosedur dalam melakukan kompresi dan dekomposisi file record. Kemudian file record yang sudah terkompresi tersebut didekompresi lalu kembali kehasil file record. Untuk mengetahui prosedur kompresi dan dekomposisi suatu file record.



Gambar 2. Prosedur Kompresi dan Dekompresi

3.2 Contoh Kasus

Terdapat 2 (dua) bagian dalam proses mengkompresi record database. Keempat proses tersebut yaitu:

1. Analisa proses kompresi file record menggunakan algoritma Gopala- Hemachandra Code 2(GH-2(n))

a. Mencari frekuensi

Sampel data yang digunakan untuk proses analisa diambil dari gambar terdapat banyak record yang ada dan dari sekian banyak record, diambil “METRONIDAZOLE SYR 125MG/SML (MEPRO)” untuk dikompresi. Setelah diketahui apa yang akan dikompresi, maka berikutnya dilakukan pencarian frekuensi serta pengurutannya dari frekuensi terbesar sampai frekuensi terkecil.

Tabel 1. Pencarian Frekuensi

Karakter	Frekuensi	Karakter	Frekuensi
M	4	1	1
E	3	2	1
T	1	5	1
R	3	M	4
O	3	G	1
N	1	/	1
I	1	S	2
D	1	M	4
A	1	L	2
Z	1	(	1
O	3	M	4
L	2	E	3
E	3	P	1
S	2	R	3
Y	1	O	3
R	3	)	1

Dari tabel 1 dilakukan pengurutan karakter dengan jumlah frekuensi terbesar sampai dengan frekuensi terkecil, serta dibuat konversi dari karakter tersebut ke bilangan biner agar dapat dihitung jumlah bitnya. Hasil pengurutannya dapat dilihat pada tabel 2 berikut.

Tabel 2. Pengurutan Frekuensi Dari Terbesar Sampai Terkecil

Karakter	Nilai Biner	Jumlah Bit	Frekuensi	Bit * Frekuensi
M	01001101	8	4	32
E	01000101	8	3	24
O	01001111	8	3	24
R	01010010	8	3	24
L	01001100	8	2	16
S	01010011	8	2	16
T	01010100	8	1	8
N	01001110	8	1	8

Karakter	Nilai Biner	Jumlah Bit	Frekuensi	Bit * Frekuensi
I	01001001	8	1	8
D	01000100	8	1	8
A	01000001	8	1	8
Z	01011010	8	1	8
Y	01011001	8	1	8
1	00000001	8	1	8
2	00000010	8	1	8
5	00000101	8	1	8
G	01000111	8	1	8
P	01010000	8	1	8
/	00101111	8	1	8
(	00101000	8	1	8
)	00101001	8	1	8
Total				256

Berdasarkan tabel 2, satu karakter bernilai 8 bit bilangan biner. Sehingga dari sampel data yang digunakan terdapat sebanyak 256 bit. Untuk mengubah satuan menjadi byte maka jumlah keseluruhan bit dibagi 8. Maka dihasilkan $256/8 = 32$ byte.

- b. Langkah selanjutnya adalah dengan membentuk tabel kode Gopala Hemachandra Code 2 (GH-2(n)) Aturan dalam pembentukan kode bilangan dengan menggunakan Gopala- Hemachandra Code 2 (GH-2(n)) dapat dilihat pada sub kajian pustaka bab sebelumnya, dimana dalam melakukan pembentukan tabel Gopala- Hemachandra Code 2 (GH-2(n)) tersebut dengan cara melakukan pembentukan kode variasi angka fibonacci orde kedua terlebih dahulu. Adapun kode pembentukan tersebut dapat dilihat pada tabel 3 dibawah ini:

Tabel 3. Fibonacci Orde Kedua

Deret Fibonacci	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Fibonacci Orde Kedua	-2	3	1	4	5	9	14	23	37	60	97

Tabel 4. Kode Gopala-Hemachandra Code 2(GH-2(n))

N	GH-2(N)
1	0011
2	10011
3	100011
4	00011
5	0000011
6	001011
7	1000011
8	010011
9	0000011
10	0010011
11	1000011
12	0100011
13	0001011
14	00000011
15	00100011
16	10000011
17	01000011

Hasil encode dari algoritma Gopala-Hemachandra Code 2 (GH-2(n)) pada sampel data “METRONIDAZOLE SYR 125MG/SML (MEPRO)” dapat dilihat pada tabel 5 dibawah ini:

Tabel 5. Karakter Yang Sudah Dikompresi Dengan Algoritma Gopala-Hemachandra Code 2 (GH-2(N))

No	Karakter	Frekuensi	Gopala-Hemachandra Code	Bit	Bit * Frekuensi
1	M	4	0011	4	16
2	E	3	10011	5	15
3	O	3	100011	6	18
4	R	3	00011	5	15
5	L	2	0000011	7	14
6	S	2	001011	6	14
7	T	1	1000011	7	7
8	N	1	010011	6	6

No	Karakter	Frekuensi	Gopala-Hemachandra Code	Bit	Bit * Frekuensi
9	I	1	0000011	7	7
10	D	1	0010011	7	7
11	A	1	1000011	7	7
12	Z	1	0100011	7	7
13	Y	1	0001011	7	7
14	1	1	00000011	8	8
15	2	1	00100011	8	8
16	5	1	10000011	8	8
17	G	1	01000011	8	8
18	P	1	00010011	8	8
19	/	1	00001011	8	8
20	(	1	00101011	8	8
21	)	1	01010011	8	8
Total					202

Setelah kode berhasil digenerate berdasarkan perhitungan algoritma Gopala-Hemachandra Code 2 (GH-2(n)), kemudian melakukan penyusunan kembali kode - kode yang dihasilkan dari proses kompresi sesuai dengan posisi karakter pada sampel data yaitu “METRONIDAZOLE SYR 125MG/SML (MEPRO)” seperti yang disajikan pada tabel 6 dibawah.

Tabel 6. String Bit Hasil Kompresi Dengan Algoritma Gopalahemachandra Code 2 (GH-2(N))

M	E	T	R	O	N	I
0011	10011	1000011	00011	100011	010011	0000011
D	A	Z	O	L	E	S
0010011	1000011	0100011	100011	0000011	10011	001011
Y	R	1	2	5	M	G
0001011	00011	00000011	00100011	10000011	0011	01000011
/	S	M	L	(	M	E
00001011	001011	0011	0000011	00101011	0011	10011
P	R	O	(			
00010011	00011	100011	01010011			

Berdasarkan pada tabel 6, string bit hasil kompresi dengan algoritma Gopala-Hemachandra Code 2 (GH-2(n)) dapat dituliskan sebagai berikut:

Tabel 7. Gabungan String Bit

00111001110000110001110001101001100000110010011100001101000111000110000011001100101100010
110001100000011001000111000001100110100001100001011001011001100000110010101100111001100010
0110001110001101010011

Sebelum mendapatkan hasil akhir kompresi keseluruhan, akan ditambahkan bit string, yaitu padding bit dan flagging bit.. Jika sisa hasil bagi setiap panjang string bit terhadap 8 adalah 0 maka akan dilakukan penambahan 00000001 di bit akhir. Apabila hasil sisa bagi dari panjang string bit terhadap yaitu $n=1,2,3,4,5,6,7,8,9$ maka, ditambahkan 0 sebanyak $7 - n + "1"$ diakhir string bit dinyatakan dengan L. Kemudian ditambahkan bilangan biner dari $9 - n$, dinyatakan di bit akhir karena jumlah string bit 202 tidak habis dibagi 8 (delapan) dan sisanya 2 bit, nyatakan sisa bagi tersebut dengan nilai n. Kemudian ditambahkan 0 sebanyak $7 - n + "1"$ diakhir string bit nyatakan dengan L. Kemudian tambahkan bilangan biner dari $9 - n$, dinyatakan dengan bit akhir.

Padding = $7 - n + "1" = 7 - 2 + "1" = 000001$

Flagging = $9 - n = 9 - 2 = 7 = 00000111$ (biner)

Tabel 8. String Bit Yang Telah Dilakukan Penambahan

0011100111000011000111000110100110000011001001110000110100
01110001100000011100110010110001011000110000001100100011100
0001100110100001100001011001011001100000110010101100111001
100010011000111000110101001100000100000111

Total panjang bit keseluruhan setelah ada penambahan bit adalah $202 + 6 + 8 = 216$. Selanjutnya lakukan pemisahan bit menjadi beberapa kelompok. Setiap kelompok terdiri dari 8 bit seperti tabel 9 di bawah ini:

Tabel 9. Pembagian String Bit

00111001	11000011	00011100	01101001	10000011
00100111	00001101	00011100	01100000	11100110
01011000	10110001	10000001	10010001	11000001
10011010	00011000	01011001	01100110	00001100
10101100	11100110	00100110	00111000	11010100
11000001	00000111			

Berdasarkan pembagian kelompok nilai biner di atas, diperoleh nilai biner baru yang dikompresi bersama dengan nilai biner dari bit tambahan yang terdiri dari 27 kelompok. Setelah pembagian selesai, nilai biner yang dibagi diubah menjadi karakter dengan terlebih dahulu mencari nilai desimal dari string bit menggunakan kode ASCII untuk menemukan nilai biner terkompresi. Nilai biner terkompresi dapat dilihat pada tabel 10 berikut:

Tabel 10. Nilai Desimal Terkompresi

No	Biner	Nilai Desimal	Karakter
1	00111001	57	9
2	11000011	195	Ã
3	00011100	28	
4	01101001	105	i
5	10000011	131	f
6	00100111	39	'
7	00001101	13	
8	00011100	28	
9	01100000	96	,
10	11100110	230	æ
11	01011000	88	x
12	10110001	177	±
13	10000001	129	
14	10010001	145	,
15	11000001	193	Á
16	10011010	154	š
17	00011000	24	↑
18	01011001	89	Y
19	01100110	102	f
20	00001100	12	◻
21	10101100	172	¬
22	11100110	230	Æ

c. Menghitung kinerja kompresi

Ukuran file awal sebelum dikompresi adalah 256 bit, sehingga rasio kompresinya adalah:

$$RC = \frac{(\text{Ukuran File Asli})}{(\text{Ukuran File Terkompresi})} = \frac{256}{202} = 1,22$$

Jika dinyatakan dalam bentuk persen maka dapat dituliskan dalam rumus sebagai berikut:

$$\begin{aligned}
 SS &= 1 - \frac{(\text{Ukuran File Terkompresi})}{(\text{Ukuran File Asli})} * 100\% \\
 &= 1 - \frac{202}{256} * 100\% \\
 &= 1 - 0,79 * 100\% \\
 &= 0,79 * 100\% \\
 &= 79\%
 \end{aligned}$$

Setelah melakukan perhitungan di atas, maka dapat disimpulkan bahwa dengan menggunakan algoritma Gopala Hemachandra Code karakter di atas dapat di kompresi sebanyak 79%.

2. Analisa proses Dekompresi File Record dengan menggunakan Gopala- Hemachandra Code 2(GH-2)n

Proses dekompresi dimulai dari merubah karakter hasil kompresi, menjadi nilai biner. Karakter hasil kompresi yang digunakan diambil dari hasil analisa prose kompresi diatas. Untuk merubah karakter hasil analisa proses kompresi menjadi nilai biner, bisa dilihat pada tabel 11.

Tabel 11. Nilai Desimal Terkompresi

No	Karakter	Nilai Desimal	Biner
1	9	57	00111001
2	Ã	195	11000011
3		28	00011100
4	i	105	01101001
5	f	131	10000011
6	,	39	00100111
7		13	00001101
8		28	00011100
9	,	96	01100000
10	æ	230	11100110
11	x	88	01011000
12	±	177	10110001
13		129	10000001
14	,	145	10010001
15	Á	193	11000001
16	š	154	10011010
17	↑	24	00011000
18	Y	89	01011001
19	f	102	01100110
20	▯	12	00001100
21	¬	172	10101100
22	Æ	230	11100110
23	&	38	00100110
24	8	56	00111000
25	Ô	212	11010100
26	Á	193	11000001
27	•	7	00000111

Dari tabel 11 didapatkan string Bit:

“001110011100001100011100011010011000001100100111000011010001110001100000110011001011000101100011000000110010001110000011001000111000001100110100001100001011001011001100000110010101100111001100010011000111000110101001100000100000111”.

Dari string bit tersebut, maka langkah selanjutnya adalah menghilangkan padding dan flagging. Menghilangkan padding dan flagging dilakukan dengan mengambil 8 bit terakhir, lalu merubahnya ke nilai desimal nyatakan dengan N. Masukkan rumus $7 + N$ untuk menghilangkan padding dan flagging. Dari string bit yang dihasilkan tersebut, diambil 8 bit terakhir yaitu 00000111 yang dirubah ke desimal menjadi “7” lalu dinyatakan dengan N, $7+7=14$, sehingga hilangkan 14 bit terakhir dari string bit yang dihasilkan. Hasil dari “0011100111000011000111000110100110000011001001110000110100011100011000001100110010110001011000110000001100100011100000110010001110000011001101000011000010110010110011000001100101011001110011000100110001110001101010011”.

Maka dari penjabaran di atas dapat disimpulkan hasil dekompresi yang dilakukan dengan algoritma Gopala-Hemachandra Code 2 (GH-2(n)) dengan code Gopala-Hemachandra yang dilakukan dengan mengubah hasil seluruh karakter menjadi string bit semula yang terdapat pada file awal, seperti pada tabel 12 dibawah ini:

Tabel 12. Hasil Dekompresi

No	Gopala Hemachandra Code	Nilai Desimal	Karakter
1	0011	77	M
2	10011	69	E
3	1000011	84	T
4	00011	82	R
5	100011	79	O
6	010011	78	N
7	0000011	73	I
8	0010011	68	D
9	1000011	65	A
10	0100011	90	Z
11	100011	79	O
12	0000011	76	L
13	10011	69	E

No	Gopala Hemachandra Code	Nilai Desimal	Karakter
14	001011	83	S
15	0001011	89	Y
16	00011	82	R
17	00000011	49	1
18	00100011	50	2
19	10000011	53	5
20	0011	77	M
21	01000011	71	G
22	00001011	47	/
23	001011	83	S
24	0011	77	M
25	0000011	76	L
26	00101011	40	(
27	0011	77	M
28	10011	69	E
29	00010011	80	P
30	00011	82	R
31	100011	79	O
32	01010011	41	)

Dari hasil dekompresi Tabel 12 di atas, maka dapat dijabarkan hasil dekompresi yaitu “METRONIDAZOLE SYR 125MG/SML (MEPRO)”.

4. KESIMPULAN

Sebagai penutup pembahasan dalam penulisan laporan penelitian ini, penulis menyimpulkan bahwa prosedur kompresi file record menggunakan Algoritma Gopala Hemachandra Code 2 (GH2(n)) telah berhasil dilakukan, di mana proses kompresi berjalan sesuai dengan teknik yang telah ditentukan. Algoritma ini terbukti efektif dalam mengompresi file record berukuran besar menjadi lebih kecil. Hasil yang diperoleh dengan nilai SS sebesar 79% dengan mengompres METRONIDAZOLE SYR 125MG/SML (MEPRO) menjadi 9Å if' `æ x ± 'Áš†Yf^α→Æ.

REFERENCES

- [1] S. M. Panjaitan, S. D. Nasution, and B. Purba, “Penerapan Algoritma Gopala-Hemachandra Code2 (GH-2 (n)) Pada Kompresi File Audio,” vol. 4, pp. 170–177, 2020, doi: 10.30865/komik.v4i1.2676.
- [2] R. Tari, S. D. Nasution, and T. Zebua, “Penerapan Algoritma Arithmetic Coding Untuk Mengkompresi Record Database Pada Aplikasi Ensiklopedia Flora Berbasis Android,” *TIN Terap. Inform. Nusantara*, vol. 2, no. 7, pp. 433–444, 2021.
- [3] N. Siregar, “Penerapan Algoritma Lzy Untuk Mengkompresi Record Database Mysql,” *J. Ilmu Komputer, Teknol. Dan Inf.*, vol. 1, no. 2, pp. 43–50, 2023.
- [4] B. W. Transform and A. S. Harahap, “Analisis Dan Implementasi Kompresi File Citra Menggunakan Algoritma,” vol. 1, no. 1, pp. 6–12, 2021.
- [5] D. Anggraini, “Penerapan Algoritma Fibonacci Code Pada File Transfer Berbasis Android,” *J. Informatics Manag. Inf. Technol.*, vol. 1, no. 3, pp. 91–94, 2021, [Online]. Available: <http://hostjournals.com/jimat/article/view/109>
- [6] P. Fitria, “Penerapan Algoritma Rice Codes Pada Aplikasi Kompresi File Gambar,” *J. Comput. Syst. Informatics*, vol. 1, no. 3, pp. 158–165, 2020.
- [7] M. A. Syahputra and P. Ramadhani, “Implementasi Metode Fibonacci Dalam Kompresi Iklan Pada Website PT. Amanah Wisata Travel,” *KOMIK (Konferensi Nas. ...)*, vol. 4, pp. 228–231, 2020, doi: 10.30865/komik.v4i1.2685.
- [8] I. Hasan, N. Irsa Syahputri, and U. Harapan Medan, “Analisis Parameter Kompresi Algoritma Elias Omega Code dan Fibonacci Code Pada File Digital,” *Algoritm. J. Ilmu Komput. dan Inform.*, vol. 6341, no. April, p. 1, 2021.
- [9] H. Hafidzah, “Perbandingan Performa Kompresi Video Berdasarkan Software Kompresi,” *UG J.*, vol. 15, no. 10, 2022.
- [10] R. Fadillah, “Penerapan Algoritma Fibonacci Codes Dalam Aplikasi Kompresi File Citra Digital,” *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. 1, pp. 225–233, 2023.
- [11] R. Fadillah, A. S. Idris, D. Marlina, G. Lubis, and R. Meisiri, “Implementasi Algoritma Fast Encryption Algorithm (FEAL) Dan Algoritma Fibonacci Mengamankan File Teks,” pp. 295–300, 2022.
- [12] D. Cahayati, A. M. H. Pardede, and H. Khair, “Implementasi Algoritma Elias Gamma Kompresi Pada File Teks,” *Algoritm. J. ILMU Komput. DAN Inform.*, vol. 6, no. 1, 2022.
- [13] M. L. Imani, R. R. Muhima, and S. Agustini, “Penerapan Metode Huffman dalam Kompresi Data,” in *Prosiding Seminar Nasional Sains dan Teknologi Terapan*, 2021, vol. 9, no. 1, pp. 457–462.
- [14] K. M. R. Hutahaean, “Penerapan Algoritma Inverted Elias Delta Untuk Kompresi Konten Pada Aplikasi Psikologi Berbasis Android,” *J. Ris. Tek. Inform. dan Data Sains*, vol. 1, no. 1, pp. 26–37, 2022.
- [15] S. O. Manik, “Perancangan Aplikasi Kompresi Citra Dengan Menerapkan Algoritma Gopala-Hemachandra-Code 2 (GH-2 (n)),” *J. Ilmu Komputer, Teknol. Dan Inf.*, vol. 1, no. 2, pp. 51–59, 2023.