

Perancangan Aplikasi Kompresi Citra Dengan Menerapkan Algoritma *Gopala-Hemachandra-Code 2 (GH-2(n))*

Sriani Oktavia Manik

Fakultas Ilmu Komputer Dan Teknologi Informasi, Prodi Studi Teknik Informatika, Universitas Budi Darma, Kota, Indonesia
Jl. Sisingamangaraja No. 338, Medan, Sumatera Utara, Indonesia
Email: sriyanimanik@gmail.com

Abstrak- Di era sekarang ini dalam segala hal bidang telah terjadi perkembangan yang begitu pesat terkhususnya di bidang teknologi, seiring dengan perkembangan itu mengakibatkan manusia kini menginginkan segala sesuatunya dapat dilakukan secara lebih efisien untuk mencapai tujuan yang terbaik dan efektif dalam penggunaan waktu, hal tersebut juga terjadi dalam bidang komunikasi. Perkembangan komunikasi didasarkan dengan kemunculan internet. Melalui internet banyak hal yang dapat dilakukan, salah satunya ialah dalam pertukaran informasi. Namun dengan adanya perkembangan tersebut membuat masalah baru, yakni berhubungan dengan ukuran data yang relatif semakin besar sehingga membutuhkan media penyimpanan agar dapat menampung semua data yang diterima. Dari permasalahan tersebut juga lah ditemukan suatu teknik agar dapat menyelesaikan segala permasalahan yang telah ditemukan sebelumnya. Adapun teknik tersebut dinamakan teknik kompresi. Teknik kompresi ini merupakan proses pengubahan data masuk menjadi data keluaran dengan hasil yang memiliki ukuran yang lebih kecil dari sebelumnya. Pada kasus ini data yang penulis gunakan dalam kompresi merupakan data citra dan algoritma yang akan diterapkan dalam proses pengompresan ini adalah Gopala-Hemachandra Code 2 (GH-2(n)). Maka dari penelitian ini lah akan didapatkan berapa persentase dari data sebelum dilakukan proses kompresi dengan yang sudah dikompresi. Sehingga para pengguna juga dapat lebih menghemat media penyimpanan tanpa kehilangan informasi yang dimaksud. Selain itu waktu yang digunakan dalam proses transmisi informasi lebih sedikit.

Kata Kunci: Kompresi, File Citra, Gopala-Hemachandra Code 2 (GH-2(n))

Abstract- In today's era in all fields, there has been rapid development, especially in the field of technology, along with these developments, humans now want everything to be done more efficiently to achieve the best and most effective use of time. communication field. The development of communication is based on the emergence of the internet. Through the internet many things can be done, one of which is the exchange of information. However, with these developments, a new problem is created, which is related to the relatively large size of the data so that it requires storage media in order to accommodate all the data received. From these problems, a technique was also found to be able to solve all the problems that had been found previously. The technique is called the compression technique. This compression technique is the process of converting incoming data into output data with results that have a smaller size than before. In this case the data that I use in compression is image data and the algorithm that will be applied in this compression process is Gopala-Hemachandra Code 2 (GH-2(n)). So from this research it will be obtained what percentage of the data before the compression process is carried out with those that have been compressed. So that users can also save more on storage media without losing the information in question. In addition, the time used in the process of transmitting information is less.

Keywords: Compression, Image File, Gopala-Hemachandra Code 2 (GH-2(n))

1. PENDAHULUAN

Pesatnya kemajuan teknologi sekarang mengarahkan kita kepada zaman yang semuanya serba canggih, terutama dalam bidang komunikasi. Yang menjadi dasar dari perkembangan komunikasi tersebut ialah dengan kemunculan internet. Semua hal kini dapat dilakukan jauh lebih mudah terkhususnya dalam hal pertukaran informasi (komunikasi). Sehubungan dengan hal tersebut maka pengaliran data akan lebih cepat dan bahkan bisa jadi lebih akurat [1]. Namun dengan perkembangan tersebut dan kebutuhan manusia yang tanpa batas akan menghadirkan permasalahan baru. Permasalahan yang ditemukan yakni berhubungan dengan ukuran data yang relatif besar sehingga membutuhkan media penyimpanan yang besar pula. Karena besarnya ukuran daripada sebuah data sangat berpengaruh terhadap kapasitas memori yang dimana akan semakin besar ukuran data maka semakin besar pula kapasitas penyimpanan yang diperlukan[2].

Akibat yang muncul dari permasalahan tersebut yakni proses pertukaran data atau informasi nya akan sangat memakan waktu. Berdasarkan permasalahan tersebut maka dilakukan satu cara untuk menyelesaiannya, yakni dengan menggunakan teknik kompresi data. Kompresi data adalah proses mengubah suatu input data menjadi data lain dengan format berbeda dan ukuran yang lebih kecil, atau proses pengkodean dari suatu data untuk mengurangi kebutuhan akan media penyimpanan[3]. Terdapat berbagai algoritma dalam kompresi, namun pada penelitian ini akan menggunakan algoritma *Gopala-Hemachandra Code 2 (GH-2(n))*. *Gopala-Hemachandra Code 2 (GH-2(n))* adalah kode *universal* baru yang merupakan variasi dari rangkaian deret *Fibonacci*. Keunikan algoritma ini adalah mengadopsi konsep dari Teorema Zeckendorf dengan suku pertama deret *Fibonacci* yang dijakan bilangan bulat negatif, sehingga berubahlah urutan dari *Fibonacci sequence* [4].

Sebelumnya dilakukan Penelitian oleh Raras Krasmala, Arif Budimansyah Purba, dan U. Tresna Lenggana pada tahun 2017 yang menyatakan bahwa kompresi citra dilakukan agar membuat kapasitas file gambar menjadi kecil sehingga dapat menghemat media penyimpanan dan tidak lambat jika melakukan pengiriman citra dari satu tempat ke tempat lain. Hanya saja akan sedikit mengurangi warna pada citra akan tetapi tidak terlalu menunjukkan perbedaan antara citra asli dengan hasil kompresi[5].

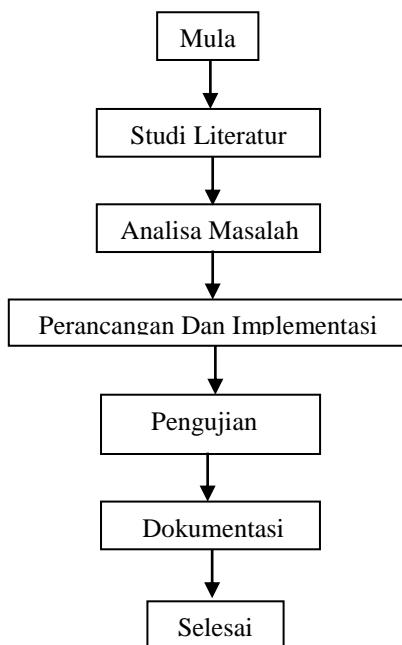
Penelitian lain yang dilakukan oleh Khairunnada pada tahun 2019 menjelaskan bahwa algoritma *Gopala-Hemachandra Code 2 (GH-2(n))* dan algoritma *start-step-stop code* menghasilkan nilai *Compression Ratio*, *Space Savings*, dan *Bit Rate* yang sama pada pengkompresan file teks, namun algoritma *Gopala-Hemachandra Code 2 (GH-2(n))* lebih tepat digunakan dalam proses dekompresi *string homogen* yang telah di kompresi [6].

Dari semua penjelasan yang telah dipaparkan maka, pada penelitian ini akan dilakukan proses perancangan aplikasi kompresi. Dimana yang digunakan sebagai objek dari penelitian ialah sebuah citra. Adapun tujuan dari penelitian ini ialah agar kita mengetahui bagaimana kinerja dari algoritma *Gopala-Hemachandra Code 2 (GH-2(n))*, dengan melakukan perhitungan terhadap nilai *Compression Ratio*, *Space Savings*, dan *Bit Rate* sebagai parameter penelitian.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Adapun tahapan penelitian yang di laksanakan adalah seperti gambar 1 di bawah ini:



Gambar 1. Tahapan Penelitian

Berdasarkan Gambar di atas dapat dijelaskan:

- a. **Studi Literatur**
Tahap ini penulis akan mencari berbagai referensi dan tinjauan pustaka dari berbagai sumber, yang membahas tentang teori-teori yang berhubungan dengan permasalahan yang terdapat pada penelitian ini.
- b. **Analisa Masalah**
Menguraikan prosedur pengkompresan ukuran citra berdasarkan algoritma *Gopala-Hemachandra Code 2 (GH-2(n))* disertai dengan batasan dan cara kerja algoritma dengan menggunakan data-data yang telah didapat dari sumber-sumber untuk melakukan analisa terhadap studi literatur yang telah didapat.
- c. **Perancangan dan Implementasi**
Tahap ini merupakan perancangan aplikasi untuk mengimplementasikan algoritma *Gopala-Hemachandra Code 2 (GH-2(n))* guna mendapatkan gambaran hasil akhir terhadap kompresi ukuran *file* citra.
- d. **Pengujian Sistem**
Melakukan pengujian terhadap sistem yang telah dirancang, sehingga dapat diketahui apakah sistem telah bekerja dengan benar dalam melakukan proses kompresi pada ukuran *file* citra.
- e. **Dokumentasi**
Pada tahap ini akan dilaksanakan dengan melakukan pendokumentasian hasil analisa dan pengujian secara tertulis dalam bentuk laporan skripsi.

2.2 Kompresi

Kompresi data adalah proses mengubah suatu *input* data menjadi data lain dengan format berbeda dan ukuran yang lebih kecil, atau proses pengkodean dari suatu data untuk mengurangi kebutuhan akan media penyimpanan [3]. Kompresi data dilakukan guna untuk menghasilkan output data dengan ukuran yang lebih kecil(berkurang dari ukuran

aslinya) sehingga lebih menghemat kapasitas penyimpanan. Teknik kompresi ini sendiri dapat diimplementasikan pada data teks, gambar, audio dan video [7]. Ada beberapa beberapa langkah yang biasanya digunakan untuk mengekspresikan kinerja setiap metode pada teknik data kompresi atau yang sering disebut sebagai parameter analisis kompresi [6].

2.2.1 Dekompreksi

Dekompreksi merupakan proses pengembalain data-data yang sebelumnya telah dilakukan proses kompresi. Proses ini dilakukan agar *user* dapat melihat kembali bahkan menggunakan data asli apabila sewaktu-waktu data tersebut dibutuhkan. Jadi pada saat dekompreksi akan terdapat catatan *header* yang berupa *byte-byte* yang berisi catatan mengenai isi dari *file* tersebut [8].

2.3 Gopala-Hemacandra Code 2 (GH-2(n))

Gopala-Hemachandra Code merupakan variasi dari deret *Fibonacci* orde kedua yang mendefenisikan urutan *Gopala-Hemacandra Code*, yang mana dapat dirumuskan [9].

Dengan nilai $b = 1 - a$

$$V Fa(1) = a \quad \{a \in \mathbb{Z}\}$$

$$V Fa(2) = 1 - a$$

Dan untuk $n \geq 3$

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

Sehingga menghasilkan:

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

Sebagai keterangan:

$V Fa$ = Variasi angka fibonacci

a = Suku pertama

Variasi angka fibonacci orde kedua yang dapat dibentuk berdasarkan rumus diatas dapat dilihat pada table dibawah ini:

Tabel 1. Variasi Angka Fibonacci Orde Kedua

K	1	2	3	4	5	6	7	8	9	10	11	12	13	14
VF-2(k)	-2	3	1	4	5	9	14	23	37	60	97	157	254	411
VF-3(k)	-3	4	1	5	6	11	17	28	45	73	118	191	309	500
VF-4(k)	-4	5	1	6	7	13	20	33	53	86	139	225	364	589
VF-5(k)	-5	6	1	7	8	15	23	38	61	99	160	259	419	678
VF-6(k)	-6	7	1	8	9	17	26	43	69	112	181	293	474	767
VF-7(k)	-7	8	1	9	10	19	29	48	77	125	202	327	529	856
VF-8(k)	-8	9	1	10	11	21	32	53	85	138	223	361	584	1168
VF-9(k)	-9	10	1	11	12	23	35	58	93	151	244	395	639	1034
VF-10(k)	-10	11	1	12	13	25	38	63	101	164	265	429	694	1123

Berkaitan dengan deret Fibonacci dinyatakan bahwa setiap bilangan bulat positif dapat ditulis secara unik sebagai penjumlahan angka Fibonacci yang tidak bersebelahan yang dikenal dengan Teorema Zeckendorf. Sebuah bilangan bulat yang ditulis sedemikian rupa dikatakan berada dalam representasi Zeckendorf. Dengan mengadopsi konsep dari teorema Zeckendorf dapat dilakukan encoding algoritma Gopala-Hemachandra Code dengan langkah-langkah sebagai berikut[5]:

- Sesuai dengan teorema Zeckendorf, diatur sedemikian rupa sehingga:

$$n = GH(i1) + (i2) + \dots + (ip)$$

Dengan:

$GH(i1)$ = angka Gopala-Hemachandra terbesar yang kurang dari atau sama dengan n

$GH(i2)$ = angka Gopala Hemachandra terbesar yang kurang atau sama dengan $n - GH(i1)$, begitu untuk seterusnya.

- Letakkan angka 1 di posisi ke- $i1$, ke- $i2$, ..., ke- ip , sementara posisi yang tersisa semuanya diisi dengan nol.
- Letakkan angka 1 di ujung untuk mengkodekan bilangan bulat positif n , sehingga kata kunci berkakhir dengan "11". Dengan begitu, sesuai dengan teorema Zeckendorf tidak ada angka 1 yang berurutan sebelum akhir dari kode yang menunjukkan bahwa setiap bilangan bulat positif dapat ditulis sebagai penjumlahan angka Fibonacci yang tidak bersebelahan.

Dengan mengikuti langkah-langkah encoding algoritma Gopala-Hemachandra Code, dihasilkan kode Gopala-Hemachandra untuk GH-2, GH-3, GH-4, GH-5 seperti pada Tabel 2.

Tabel 2. Beberapa Gopala-Hemachandra Code

n	GH-2(N)	GH-3(N)	GH-4(N)	GH-5(N)
1	0011	0011	0011	0011
2	10011	10011	10011	10011
3	011	100011	100011	100011
4	00011	011	101011	101011
5	0000011	00011	011	N/A
6	001011	000011	00011	011
7	01011	001011	000011	00011
8	1010011	1000011	001011	000011
9	0000011	1010011	1000011	001011
10	0010011	010011	1010011	1000011

2.4 Citra

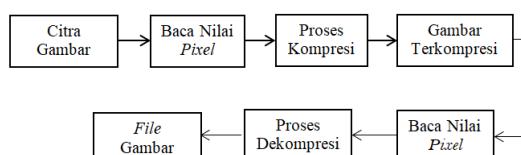
Citra merupakan komponen multimedia yang berperan penting dalam bentuk informasi *visual*. Kumpulan titik dengan intesitas warna tertentu yang membentuk suatu kesatuan dan memiliki pengertian artistik dapat diartikan sebagai citra. Informasi yang diberikan pada suatu citra dapat memacu imajinasi penglihatan dan menyimpulkan informasi dari citra tersebut[2]. Nilai suatu *pixel* memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0–255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra *integer*. Secara sistematis citra digital dapat dituliskan sebagai fungsi $f(x,y)$, dimana x merupakan baris dan y merupakan kolom.

3. HASIL DAN PEMBAHASAN

3.1 Analisa

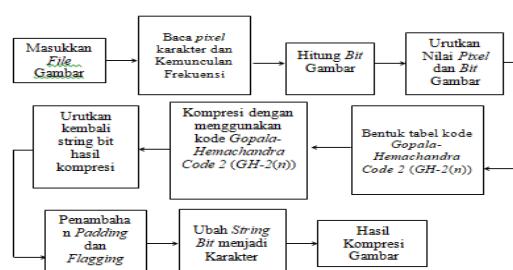
Pengompresan citra gambar di lakukan guna untuk mengurangi ruang penyimpanan serta mempercepat proses pengiriman citra tersebut. Dalam penelitian ini yang akan dibahas adalah bagaimana cara kerja dari algoritma *Gopala-Hemachandra Code 2 (GH-2(n))* dalam mengompresi citra gambar dengan menggunakan analisis kompresi. Citra gambar yang akan digunakan dalam penelitian ini berekstensi BMP. Gambar BMP merupakan gambar yang tanpa proses kompresi yang menyebabkan ukuran dari format gambar ini cukup besar. Selain itu format BMP ini sangat tergantung pada kualitas gambar, semakin bagus kualitas gambar maka semakin besar pula ukuran dari gambar tersebut, yang mengakibatkan tempat penyimpanan dan waktu transmisi nya semakin besar.

Dalam melakukan proses kompresi citra gambar terdapat beberapa tahapan, yakni yang pertama sekali ialah dengan menginputkan citra gambar kemudian akan dilakukan pembacaan nilai *pixel* langkah selanjutnya ialah melakukan proses kompresi dengan menerapkan algoritma yang telah ditentukan, setelah itu akan keluar hasil kompresi citra. Selanjutnya ialah proses dekompresi, adapun langkah-langkahnya ialah: petama sekali akan dilakukan penginputan citra gambar kemudian dilakukan pembacaan nilai pixel, selanjutnya ialah proses dekompresi dengan menerapkan algoritma yang telah ditentukan, setelah itu akan keluar hasil output berupa citra gambar asli. Untuk membaca nilai *pixel* penulis menggunakan aplikasi *MATLAB2010a*. Adapun prosedurnya dapat dilihat pada gambar dibawah ini:



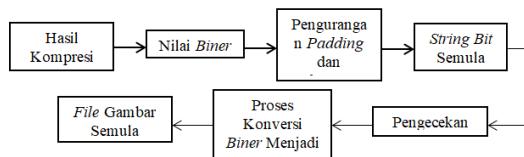
Gambar 2. Prosedur Kompresi dan Dekompresi Citra Gambar

Berikut ini dijelaskan tahapan algoritma kompresi pada gambar 2 dibawah ini:



Gambar 3. Tahapan Algoritma Kompresi

Selanjutnya dijelaskan tahapan dekompresi:



Gambar 4. Tahapan Dekompreksi

Pada penelitian ini, penulis telah menetapkan bahwa yang objek yang digunakan adalah sebuah citra gambar dengan resolusi *Bitmap*, dikarena gambar dengan resolusi BMP memiliki ukuran yang cukup besar. Terdapat 2 proses yang akan dilaksanakan pada penelitian ini, dan pada sub bab ini lah penulis akan menejelaskan kedua proses tersebut, yakni proses kompresi dan proses dekompreksi. Algoritma yang digunakan penulis dalam mengkompresi citra gambar yakni algoritma *Gopala-Hemachandra Code 2 (GH-2(n))* yang mana algoritma ini merupakan salah satu algoritma yang termasuk dalam metode *lossless*. Dalam menganalisa citra gambar harus dilakukan pengambilan *sampel* agar kita mengetahui nilai *pixel*-nya. Adapun langkah untuk mengkompresi dan mendekompreksi citra gambar ialah sebagai berikut:

- Langkah pertama yakni mempersiapkan citra gambar yang akan di kompresi dengan format BMP.



Gambar 5. Citra Gambar berformat BMP

Berdasarkan proses analisa pembacaan nilai piksel yang telah dilakukan, maka diambil sampel sebesar 5×5 *pixel* dengan kedalaman warna 24 *bit*. Dengan begitu dapat dihitung ukuran dari citra tersebut ialah $5 \times 5 \times 24$ *bit* = 600 *bit*. Apabila diubah kedalam satuan *bytes* menjadi $600/8 = 75$ *bytes*. Berikut sampel gambar yang akan dikompresi.



Gambar 6. Sampel Gambar 5×5 *pixel*

- Langkah selanjutnya ialah dengan melakukan pembacaan nilai citra gambar.

$$\begin{bmatrix} 225 & 225 & 225 & 226 & 226 \\ 225 & 226 & 227 & 226 & 226 \\ 226 & 226 & 226 & 226 & 226 \\ 226 & 226 & 226 & 225 & 226 \\ 226 & 227 & 226 & 224 & 225 \end{bmatrix}$$

Berdasarkan gambar matriks diatas maka didapat nilai *pixel* sebagai berikut: 225, 225, 225, 225, 226, 225, 226, 227, 226, 226, 226, 226, 226, 226, 226, 226, 226, 226, 225, 226, 226, 227, 226, 226, 224, 225. Kemudian nilai ini akan dimasukkan kedalam sebuah tabel untuk melakukan pembacaan frekuensi. Pembacaan frekuensi dilakukan guna untuk menghitung jumlah nilai yang sama pada nilai desimal yang muncul. Dapat dilihat pada tabel dibawah ini:

Tabel 3. Pembacaan Nilai Desimal

No.	Desimal	Frekuensi
1	225	7
2	226	15
3	227	2
4	224	1
Jumlah		25

Berdasarkan tabel di atas, didapatkan beberapa nilai *pixel* yang sama. Sebelum proses kompresi citra, langkah awal adalah membaca nilai *pixel* citra kemudian pembuatan tabel nilai *pixel* yang telah diurutkan dari nilai frekuensi

terbesar (nilai *pixel* yang mempunyai nilai yang sama) ke nilai frekuensi terkecil. Urutan nilai *pixel* setelah dilakukan pembacaan nilai *pixel* terbesar dapat dilihat pada tabel dibawah ini:

Tabel 4. *Pixel yang Belum Dikompresi*

No	Nilai Heksa	Binary	Bit	Frekuensi	Bit x Frek
1	226	11100010	8	15	120
2	225	11100001	8	7	56
3	227	11100011	8	2	16
4	224	11100000	8	1	8
Jumlah Bit					200 Bit

- c. Selanjutnya ialah dengan membentuk tabel kode *Gopala-Hemachandra Code 2 (GH-2(n))*. Adapun aturan dalam pembentukan kode bilangan dapat dilihat pada sub landasan teori yang sebelumnya sudah dijelaskan. Dibawah ini merupakan tabel kode bilangan pada algoritma *Gopala-Hemachandra Code 2 (GH-2(n))* yang proses pembentukannya sesuai dengan cara pembentukan kode Fibonacci, jika n=1 maka:

Tabel 5. Kode bilangan *Gopala-Hemachandra Code 2 (GH-2(n))*

Urutan bilangan Fibonacci dalam deret fibonacci	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)	F(7)
Bilangan fibonacci	-2	3	1	4	5	9	14	23
Kode Fibonacci sementara	0	0	1					

Namun pada akhir pembentukan kode Fibonacci adalah menambahkan angka “1” pada posisi paling kanan dari kode, maka angka Fibonacci dengan $n=1$ adalah 0011 dan seterusnya. Sehingga terbentuklah tabel sebagai berikut:

Tabel 6. Kode Gopala-Hemachandra Code 2 (GH-2(n))

n	GH2(N)	GH-3(N)	GH-4(N)	GH5(N)
1	0011	0011	0011	0011
2	10011	10011	10011	10011
3	011	100011	100011	100011
4	00011	011	101011	101011

Hasil *encode* dari algoritma *Gopala-Hemachandra Code 2 (GH-2(n))* pada nilai desimal ‘225, 225, 225, 225, 225, 226, 225, 226, 227, 226, 226, 226, 226, 226, 226, 226, 226, 226, 226, 226, 225, 226, 226, 227, 226, 226, 224, 225’ dapat dilihat pada tabel berikut ini:

Tabel 7. Desimal yang sudah dikompresi dengan algoritma *Gopala-Hemachandra Code 2 (GH-2(n))*

Nilai Heksa	Frekuensi	Gopala-Hemchandra Code	Bit	Bit x Frekuensi
226	15	0011	4	60
225	7	10011	5	35
227	2	011	3	6
224	1	00011	5	5
Total				106 Bit

Berdasarkan pada tabel 3.4 diatas dapat dibentuk nilai *bit* baru hasil kompresi dari susunan nilai desimal citra gambar sebelum kompresi yaitu 225, 225, 225, 225, 225, 226, 225, 226, 227, 226, 226, 226, 226, 226, 226, 226, 226, 226, 226, 226, 225, 226, 226, 227, 226, 224, 225 menjadi nilai bit *biner*:

- d. Kemudian sebelum didapatkan hasil keseluruhan akhir kompresi dilakukan penambahan *string bit* atau dinamakan *padding bit* dan *flagging bit*. Jika sisa bagi panjang *string bit* terhadap 8 adalah 0 maka tambahan 00000001. Nyatakan dengan *bit akhir*. Sedangkan jika sisa bagi panjang *string bit* terhadap 8 adalah $n(1,2,3,4,5,6,7)$ maka tambahkan 0 sebanyak $7 - n + "1"$ di akhir *string bit*, nyatakan dengan L. Lalu tambahkan bilangan *biner* dari $9 - n$, nyatakan dengan *bit akhir*. Karena jumlah *string bit* hasil kompresi 106 tidak habis dibagi 8 dengan sisa 2 bit, nyatakan sisa bagi tersebut dengan nilai n . maka tambahkan 0 sebanyak $7 - n + "1"$ diakhir *string bit*. Nyatakan dengan L. Lalu tambahkan bilangan biner dari $9-n$. nyatakan dengan *bit akhir*.

$$\begin{array}{r} 7 - n + "1" \\ 7 - 2 + "1" = 000001 \\ \text{Bit Akhir } 9 - n \\ \text{Bit Akhir} = 9 - 2 = 7 = 00000111 \end{array}$$

Gambar 7. Perhitungan Penambahan Bit

Gambar 8. *String Bit* Yang Telah Dilakukan Penambahan

Total panjang *bit* keseluruhan setelah ada penambahan *bit* adalah $106 + 6 + 8 = 120$. Selanjutnya lakukan pemisahan *bit* menjadi beberapa kelompok. Setiap kelompok terdiri dari 8 *bit* seperti gambar di bawah ini.

10011100	11100111	00110011	10011001	10110011
00110011	00110011	00110011	00110011	00110011
10011001	10110011	00011100	11000001	00000011

Gambar 9. Pembagian String Bit

Berdasarkan pada pembagian kelompok nilai *biner*, didapatkan 15 kelompok nilai *biner* baru yang sudah terkompresi beserta nilai *biner* penambahan *bit*. Setelah pembagian dilakukan, maka nilai *biner* yang sudah dibagi dirubah kedalam suatu karakter dengan terlebih dahulu mencari nilai desimal dari *string bit* tersebut menggunakan kode *ASCII*. Adapun nilai desimal yang sudah terkompresi dapat dilihat pada tabel di bawah ini.

Tabel 8. Nilai Desimal Terkompresi

Biner	Urutan Desimal terkompresi	Karakter
10011100	156	œ
11100111	231	ç
00110011	51	3
10011001	153	TM
10110011	179	³
00110011	51	3
00110011	51	3
00110011	51	3
00110011	51	3
00111001	57	9
10011001	153	TM
10110011	179	³
00011100	28	[file separator]
11000001	193	Á
00000111	7	[bell]

- e. Setelah nilai desimal diketahui, maka mengubah nilai desimal kedalam suatu karakter. Karakter hasil dari proses kompresi yang dihasilkan tersimpan dalam suatu *file* dengan ekstensi “ghc”, dan jika *file* tersebut dibuka dengan aplikasi *notepad*, maka akan tampil karakter seperti gambar berikut :



Gambar 10. Hasil Karakter Kompresi

Berdasarkan hasil dari kompresi dengan menerapkan algoritma *Gopala-Hemachandra Code 2 (GH-2(n))* diatas dapat dihitung kinerja kompresinya yaitu:

a. *Compression Ratio(CR)*

$$CR = \frac{\text{Ukuran data output}}{\text{Ukuran Data input}} = 100\%$$

$$CR = \frac{106}{200} \times 100\%$$

$$CR = 53\%$$

b. *Space Saving (SS)*

$$SS = 100\% - CR$$

$$SS = 100\% - 53\%$$

$$SS = 47\%$$

Sehingga dari perhitungan diatas, dapat diambil kesimpulan bahwa kompresi citra gambar yang dilakukan dengan menggunakan algoritma *Gopala-Hemachandra Code 2 (GH-2(n))* memperkecil ukuran data sebesar 53% dari ukuran asli dan juga meningkatkan penyimpanan sebesar 47%.

3.1.1 Proses Dekompresi *Gopala-Hemachandra Code 2 (GH-2(n))*

Proses dekompresi untuk file hasil kompresi menggunakan algoritma *Gopala Hemachandra Code* dilakukan dengan menggunakan metode *Brute Force*. Pembacaan *string bit* dilakukan dari indeks terkecil sampai indeks terakhir dengan terus menambahkan nilai pada indeks sebelumnya yang tidak mewakili karakter *Gopala-Hemachandra Code*. Proses dekompresi hal yang dilakukan adalah menganalisa keseluruhan *bit* hasil dari kompresi sebelumnya. Adapun *bit* keseluruhan hasil kompresi dapat dilihat pada tabel berikut :

Tabel 9. Nilai Desimal Terkompresi

Biner	Urutan Desimal terkompresi	Karakter
10011100	156	Œ
11100111	231	Ҫ
00110011	51	3
10011001	153	™
10110011	179	ȝ
00110011	51	3
00110011	51	3
00110011	51	3
00110011	51	3
00111001	57	9
10011001	153	™
10110011	179	ȝ
00011100	28	[file separator]
11000001	193	Á
00000111	7	[bell]

Berdasarkan pada tabel diatas maka diambil seluruh nilai *biner* dan digabungkan menjadi:

Selanjutnya adalah dengan mengembalikan *binary* menjadi *string bit* semula dengan menghilangkan *biner* yang ditebalkan. Untuk mengembalikan *binary* menjadi *string bit* semula dapat dilakukan melalui langkah berikut ini. Lakukan pembacaan pada 8 *bit* terakhir, hasil pembacaan berupa bilangan desimal. Nyatakan hasil pembacaan dengan n . Hilangkan *bit* pada bagian akhir sebanyak $7 + n$. Setelah dilakukan perhitungan pembacaan *bit* akhir . Nilai *biner* yang dihilangkan sebanyak 8 *bit* pada akhir. $n = 7$ hilangkan $7 + 7$ atau $7+7 = 14$. Penjelasan diatas menunjukan bahwa *bit* akhir harus dihilangkan. Hasil pengembalian *binary* menjadi *string bit* semula dapat dilihat sebagai berikut ini:

Maka dari penjabaran diatas dapat disimpulkan hasil dekompreksi seperti pada tabel dibawah ini:

Tabel 10. Nilai Desimal dan *Gopala-Hemachandra Code*

No	Gopala-Hemachandra Code	Nilai Desimal
1	10011	225

No	Gopala-Hemachandra Code	Nilai Desimal
2	10011	225
3	10011	225
4	10011	225
5	0011	226
6	10011	225
7	0011	226
8	011	227
9	0011	226
10	0011	226
11	0011	226
12	0011	226
13	0011	226
14	0011	226
15	0011	226
16	0011	226
17	0011	226
18	0011	226
19	10011	225
20	0011	226
21	0011	226
22	011	227
23	0011	226
24	00011	224
25	10011	225

4. KESIMPULAN

Berdasarkan dari analisa dan penelitian yang telah dilakukan sebelumnya, maka akan diperoleh hasil akhir dari pembahasan, dimana hasil tersebut dapat dijadikan sebagai kesimpulan yang kemungkinan hal tersebut merupakan sebuah kelebihan ataupun kekurangan. Adapun beberapa kesimpulan yang dimaksud ialah Prosedur kompresi dengan menerapkan algoritma *Gopala-Hemachandra Code 2(GH-2(n))* yang dilakukan dalam mengkompres citra gambar dengan format bmp telah berhasil dilakukan dan berjalan sesuai dengan teknik kompresi, untuk penerapan algoritma *Gopala-Hemachandra Code 2(GH-2(n))* dalam mengkompres citra gambar berhasil memperkecil ukuran gambar yang relatif besar menjadi ukuran yang lebih kecil. Dimana *compression ratio* yang berhasil diperoleh ialah sebesar 53% dan *space saving* yang ditingkatkan sebesar 47%. Aplikasi *Microsoft Visual basic Studio 2008* dapat digunakan dalam mengkompresi citra gambar dengan menerapkan algoritma *Gopala-Hemachandra Code 2(GH-2(n))* dan mampu memudahkan penulis dalam mengkompresi ukuran gambar.

REFERENCES

- [1] A. P. U. Siahaan, "Implementasi Teknik Kompresi Teks Huffman," *None*, vol. 10, no. 2, p. 101651, 2016.
- [2] D. Ardiyanto and B. H. Purwoto, "Kompresi citra dengan menggunakan metode delta modulation," *J. Emit.*, vol. 14, no. 01, pp. 1–12, 2012.
- [3] E. Haryanty, "Analisa Kompresi Citra Digital Menggunakan Metode Hadamard," *Teknika*, vol. 1, no. 1, pp. 13–20, 2012, doi: 10.34148/teknika.v1i1.2.
- [4] J. H. Thomas, *Variations on the Fibonacci universal code*. arXiv preprint, 2007.
- [5] R. Krasmala, A. Budimansyah, and U. T. Lenggana, "Kompresi Citra Dengan Menggabungkan Metode Discrete Cosine Transform (DCT) dan Algoritma Huffman," *J. Online Inform.*, vol. 2, no. 1, p. 1, Jul. 2017, doi: 10.15575/join.v2i1.79.
- [6] U. S. Utara, U. S. Utara, and U. S. Utara, "Analisis Perbandingan Kinerja Algoritma Start-Step-Stop Code dan Gopala-Hemachandra Code 2 (GH-2 (n)) pada Kompresi File Teks," vol. 2, 2019.
- [7] M. Ali, "Analisis Perbandingan Algoritma Even-Rodeh Code dan Algoritma Fibonacci Code untuk Kompresi File Teks," 2018.
- [8] C. T. Utari, "Implementasi Algoritma Run Length Encoding Untuk Perancangan Aplikasi Kompresi Dan Dekompresi File Citra," *J. TIMES*, vol. V, no. 2, pp. 24–31, 2016.
- [9] Hendri, "Kompresi Citra dari Format BMP ke Format PNG," *Time*, vol. III, no. 1, pp. 27–31, 2014.