

Implementasi Metode Felics Pada Kompresi Citra Format GIF

Theresia Lumbantobing

Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Jl. Sisingamangaraja No.338, Siti Rejo I, Kec. Medan Kota, Medan, Sumatera Utara, Indonesia

Email: Theresiatobiing@gmail.com

Email Penulis Korespondensi:-

Abstrak-Pada saat ini perkembangan ilmu pengetahuan dan teknologi yang semakin pesat membuat file citra dua dimensi menjadi sangat dibutuhkan dimana ukuran citra yang dihasilkan besar sehingga memerlukan media penyimpanan yang sangat besar, dan munculnya permasalahan lain saat pertukaran data dan informasi yang semakin menguras waktu dan bandwidth. Maka dari itu muncul jugalah solusi untuk masalah di atas, yaitu dengan cara kompresi data. Kompresi data atau pemampatan data adalah sebuah cara dalam ilmu komputer untuk memerlukan ruangan penyimpanan lebih kecil, sehingga lebih efisien dalam penyimpanannya. Ini adalah sebuah metode kompresi yang dirancang untuk gambar grayscale(skala abu-abu).Metode FELICS menyajikan sistem yang lebih sederhana untuk kompresi gambar lossless itu berjalan dengan sangat cepat dan hanya kehilangan efisiensi kompresi yang minimal. Implementasi merupakan Penerapan atau pelaksanaan, sebagai tindakan untuk menjalankan rencana yang telah dibuat.

Kata Kunci: Implementasi; FELICS; Citra.gif.

Abstract-At this time, the increasingly rapid development of science and technology makes two-dimensional image files very necessary, where the resulting image size is large, requiring very large storage media, and other problems arise when exchanging data and information which increasingly consumes time and bandwidth. Therefore, a solution to the problem above emerged, namely by data compression. Data compression or data compression is a way in computer science to require smaller storage space, so that storage is more efficient. This is a compression method designed for grayscale images. The FELICS method presents a simpler system for lossless image compression that runs very quickly and only loses minimal compression efficiency. Implementation is the application or implementation, as an action to carry out the plan that has been made.

Keywords: Implementation; FELIC; Citra.gif.

1. PENDAHULUAN

Pada saat ini perkembangan ilmu pengetahuan dan teknologi yang semakin pesat membuat file citra dua dimensi menjadi sangat dibutuhkan. Oleh karena itu, dapat dilihat dari kualitas citra dan semakin tingginya resolusi kamera digital saat ini sehingga menghasilkan resolusi citra yang tinggi dengan ukuran yang tinggi pula.Sementara itu untuk penyimpanan citra sendiri memerlukan *storage* yang cukup besar sehingga membuat *storage* cepat penuh, saat pertukaran data informasi juga menguras waktu dan *bandwidth*.

Secara umum citra ialah suatu gambar, foto ataupun tampilan dua dimensi lainnya yang menggambarkan suatu visualisasi objek.Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya. Sumber cahaya menerangi objek, objek kembali memantulkan sebagian dari cahaya tersebut.Pantulan ini kemudian ditangkap oleh alat-alat optik misalnya mata manusia, *scanner*, kamera, dan lain sebagainya[1]. Citra sebagai alat keluaran dari sistem perekaman dapat bersifat, optik berupa foto, analog berupa sinyal video seperti gambar pada monitor televisi, dan digital yang dapat langsung disimpan pada suatu pita magnetik [2].

Teknik kompresi yang ada sekarang memungkinkan citra dikompresi, sehingga ukurannya menjadi jauh lebih kecil dari pada ukuran asli. Sayangnya, rasio yang dihasilkan kompresi citra sangat rendah karena banyaknya aplikasi yang memerlukan kompresi tanpa cacat, seperti pada aplikasi radiografi, kompresi citra hasil diagnose medis atau gambar satelit, dimana kehilangan gambar sekecil apapun akan menyebabkan hasil yang tak diharapkan.

Permasalahan dimana ukuran citra yang dihasilkan besar sehingga memerlukan media penyimpanan yang sangat besar, dan munculnya permasalahan lain saat pertukaran data dan informasi yang semakin menguras waktu dan *bandwidth*. Maka dari itu muncul jugalah solusi untuk masalah di atas, yaitu dengan cara kompresi data. Kompresi data adalah proses mengubah aliran data *input* (sumber aliran atau data mentah asli) menjadi aliran data lain (*output*, aliran terkompresi) yang memiliki ukuran lebih kecil.

Graphics Interchange Format (GIF) adalah sebuah format gambar yang sejak dulu dikenal sebagai format gambar yang bisa bergerak, ada animasinya walaupun singkat. Berbeda dengan *JPG*, *BMP*, atau *PNG* yang cenderung diam. *GIF* merupakan format grafis untuk penyimpanan gambar warna digital yang terdiri dari 256 warna dan animasi pendek yang paling sering digunakan untuk keperluan desain website jika dibandingkan dengan format gambar lainnya. *GIF* juga baik untuk menampilkan gambar dengan komposisi yang tidak menggunakan terlalu banyak warna seperti gambar kartun[3]. Pada awalnya gambar *GIF* dibuat menggunakan kumpulan dari beberapa gambar yang digerakkan secara statis layaknya sebuah video pada periode waktu tertentu Hasilnya adalah gambar *GIF* tersebut akan terlihat seolah-olah bergerak atau memiliki animasi.*GIF* adalah salah satu format citra yang paling sering ditemui di dunia digital.Hal ini, karena ukuran dari format ini relative lebih kecil. Misalnya, berkas dengan format *GIF* dapat lebih kecil ukurannya dibandingkan dengan format *JPG* [4].

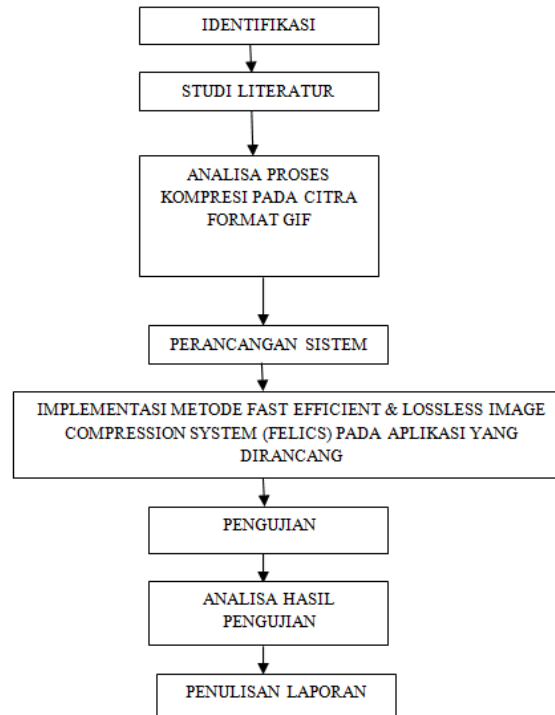
Oleh karena itu, pada penelitian ini metode yang digunakan untuk kompresi data gambar adalah Metode *FELICS*.Metode ini adalah salah satu metode kompresi yang dapat digunakan untuk mengurangi kapasitas sebuah gambar. Metode *FELICS* merupakan singkatan dari *Fast EfficientCompression & Lossless Image System*, adalah

algoritma kompresi gambar *lossless* yang berkinerja 5 kali lebih cepat daripada codec JPEG *lossless* asli dan menghasilkan rasio kompresi dengan nilai sama[5]. Dari beberapa penjabaran di atas, penulis telah mengumpulkan beberapa jurnal dari berbagai sumber yang berkaitan dengan permasalahan yang sedang penulis bahas. Sesuai dengan penjelasan latar belakang masalah di atas maka penulis menyusun penelitian dengan judul “ Implementasi Metode Fast Efficient & Lossless Image Compression System (FELICS) Pada Kompresi Citra Format GIF ”.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Adapun Tahapan Penelitian yang akan dilakukan dapat dilihat sebagai berikut :



Gambar 1. Tahapan Penelitian

2.2 Kompresi Citra

Kompresi citra adalah proses pemampatan citra yang bertujuan untuk mengurangi duplikasi data pada citra sehingga memori yang digunakan untuk mempresentasikan citra menjadi lebih sedikit daripada representasi citra semula[6]. Pada saat ini kompresi data sangat penting dikarenakan dapat menghemat penyimpanan data, menghemat waktu pengiriman data serta memperkecil kebutuhan *bandwidth*. Teknik kompresi bisa dilakukan terhadap data teks/biner, gambar (*JPEG, PNG, TIFF*), audio (*MP3, AAC, RMA, WMA*), dan video (*MPEG, H261, H263*).

Maka dari itu istilah lain kompresi sering disebut juga *source coding, data compression, bandwidth compression*, dan *signal compression*. Terdapat banyak metode untuk kompresi data. Metode-metode tersebut lahir dari ide yang berbeda-beda, cocok untuk berbagai tipe data, dan menghasilkan *output* yang berbeda. Namun, prinsip dasar yang menjadi dasar tiap metode adalah sama, yaitu mengkompres data dengan *redundancy* dari data asli. Berdasarkan kebutuhan datanya, jenis-jenis kompresi dibagi menjadi atas dua bagian, yaitu:

a. Kompresi *Loseless*

Kompresi *Loseless* merupakan jenis kompresi yang *reversible*, yakni data yang sudah terkena kompresi dimungkinkan untuk membentuk kembali tepat sama dengan data awal sebelum terkena proses kompresi. Jenis kompresi ini sering diterapkan dalam file teks.

b. Kompresi *Lossy*

Kompresi *Lossy* merupakan jenis kompresi *irreversible*, yaitu yang menghasilkan data yang berbeda apabila dilakukan proses dekompresi. Data hasil dekompresinya mengalami kehilangan beberapa data yang tidak sesuai dengan data awal sebelum terkena proses kompresi, jenis kompresi ini sering diterapkan dalam file gambar, suara, maupun video[7].

2.3 Metode FELICS

Metode FELICS merupakan singkatan dari *Fast Efficient & Loseless Image Compression System*. Ini adalah sebuah metode kompresi yang dirancang untuk gambargayscale (skala abu-abu). Prinsip FELICS adalah membuat kode piksel

cache dengan kode ukuran variable berdasarkan nilai dari dua piksel tetangganya yang terlihat sebelumnya. Metode FELICS menyajikan sistem yang lebih sederhana untuk kompresi gambar lossless itu berjalan dengan sangat cepat dan hanya kehilangan efisiensi kompresi yang minimal.

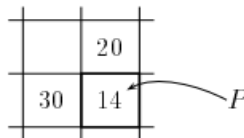
FELICS ditemukan oleh Paul G. Howard dan Jeffrey S. Vitter dari Departemen Ilmu Komputer di Brown University di Providence, Rhode Island, AS, dan pertama kali dipresentasikan pada Konferensi Kompresi Data IEEE 1993 di Snowbird, Utah. Dan telah diimplementasikan dalam perangkat keras dan digunakan sebagai bagian dari HiRISE di *Mars Reconnaissance Orbiter*.

Untuk menyandikan gambar, langkah yang dilakukan pertama mengeluarkan dua piksel pertama tana pengkodean, lalu ulangi langkah-langkah berikut:

- Memilih piksel P berikutnya dan menemukan tetangga terdekatnya N1 dan N2.
- Menghitung $L = \min(N1; N2)$, $H = \max(N1; N2)$, dan $\Delta = H - L$
 - If $L \leq P \leq H$, menggunakan satu bit untuk menyandikan *IN-RANGE*; lalu kita menggunakan kode biner yang disesuaikan untuk mengkodekan $P - L$ di $[0, \Delta]$.
 - If $P < L$, menggunakan satu bit untuk menyandikan *OUT-OF-RANGE*, dan satu bit untuk menyandikan *BELOW-RANGE*. Kemudian menggunakan kode Golomb-Rice untuk mengkodekan integer non-negatif $L - P - 1$.
 - If $P > H$, menggunakan satu bit untuk menyandikan *OUT-OF-RANGE*, dan satu bit untuk menyandikan *ABOVE-RANGE*. Kemudian menggunakan kode *Golomb-Rice* untuk mengkodekan integer non-negatif $P - H - 1$.

Algoritma *decoding* melibatkan hanya membalikkan langkah 3, *decoding* keputusan *in-range/ out of-range* dan di atas-range/ di bawah-range, bercabang sesuai, dan menyesuaikan angka yang diterjemahkan untuk menyandikan nilai P.

Dalam contoh berikut ini menggunakan 1 untuk menyandikan *IN-RANGE*, 0 untuk menyandikan *OUT-OF-RANGE*, 0 untuk menyandikan *BELOW-RANGE*, dan 1 untuk menyandikan *ABOVE-RANGE*.

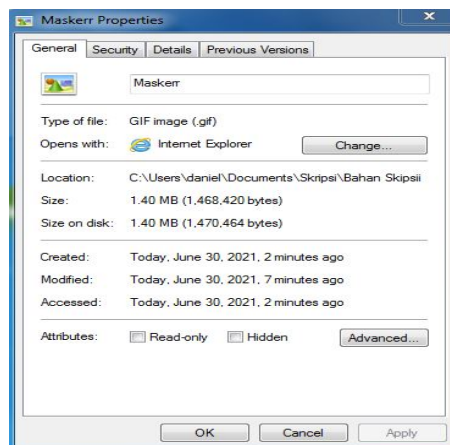


Misalkan kita ingin mengkodekan intensitas piksel P pada gambar contoh piksel di atas; nilainya adalah 14. Tetangga terdekat memiliki intensitas $N1 = 30$ dan $N2 = 20$, jadi $L = 20$ dan $H = 30$. Karena $P < L$, maka menghasilkan 0 untuk *OUT-OF-RANGE* dan 0 lainnya untuk *BELOW-RANGE*. Konteks untuk piksel P adalah perbedaan $= H - L = 10$. Sebagai contoh, kita ibaratkan bahwa kita memerkirakan parameter pengkodean *Rice Code* untuk konteks $\Delta = 10$ menjadi $k_{10} = 2$. Karena *Rice Code* digunakan untuk menyandikan bilangan bulat *non-negatif*, menyandikan $n = 0$ untuk nilai di luar kisaran (dalam hal ini $P = 19$ akan dikodekan sebagai $n = 0$), $n = 1$ untuk nilai yang selanjutnya, dan seterusnya. Nilai $P = 14$ dikodekan sebagai $n = 5$. Untuk mengkodekan dengan *Rice Code* dengan $k = 2$, kita membagi $n = 5$ menjadi dua bagian, 1 01, bagian *low-order* memiliki $k = 2$ bit. Kemudian output 1 bagian *high-order* (*unary*), memberikan 10, dan k bit *low-order*, 01 secara langsung. Jadi output untuk piksel ini adalah 001001.

3. HASIL DAN PEMBAHASAN

3.1 Sampel Data

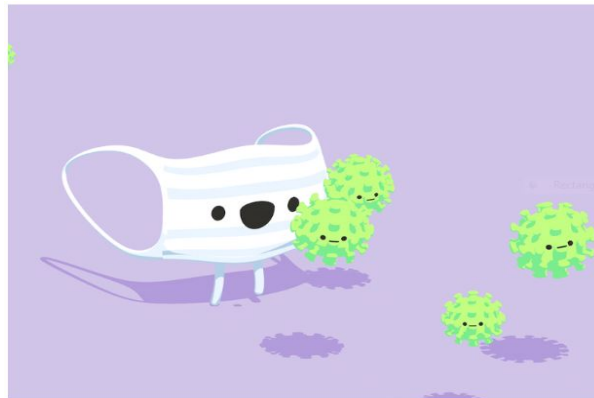
Sampel data yang digunakan dalam penelitian ini adalah citra dengan format GIF. Adapun spesifikasi citra yang digunakan dalam penelitian ini, adalah sebagai berikut:



Gambar 2. Spesifikasi Citra File Maskerr.gif

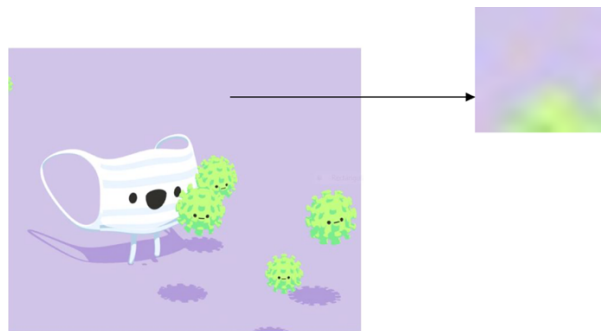
3.2 Analisa Perancangan Metode

Untuk kepentingan hitungan manual, maka penulis mengambil satu objek ,yang kemudian objek tersebut akan diperkecil untuk dijadikan sebagai sampel. Adapun objek citra file GIF tersebut adalah:



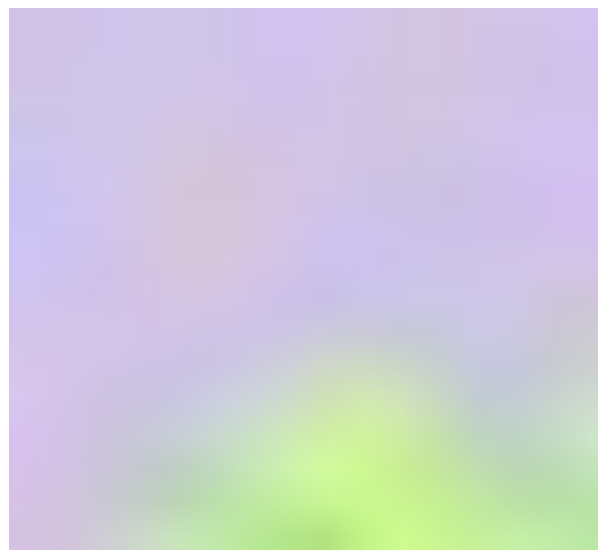
Gambar 3. Objek Citra File GIF

Untuk memudahkan proses penerapan metode FELICS, penulis memperkecil bagian citra Maskerr.gif yang akan dijadikan sebagai sampel. Berikut tampilan citra sampel, dan penulis menggunakan aplikasi Matlab untuk mengambil bagian kecil citra.



Gambar 4. Pengambilan citra sampel dengan resolusi piksel 16 x 16

Selanjutnya terdapat gambar citra sampel dengan resolusi 16 x 16 pixel



Gambar 5. Citra sampel dengan resolusi 16 x 16 piksel

Dari citra sampel dengan piksel 16 x 16 dilakukan proses pengambilan nilai citra *Grayscale*. Berikut adalah hasil nilai citra *Grayscale*:

Tabel 1. Nilai *Grayscale* 16 x 16

203	203	204	204	204	204	204	204	204	204	204	204	204	204	204	205	205
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

204	204	204	204	205	205	204	204	204	204	204	204	204	203	203	203
204	204	204	204	205	205	204	203	204	204	204	204	204	204	203	203
204	205	205	205	205	204	204	204	204	204	204	205	205	205	205	205
204	205	206	206	204	204	204	205	204	204	203	203	203	204	205	206
203	205	206	206	205	204	205	205	204	203	202	201	201	202	203	204
204	204	205	205	205	206	205	205	204	203	203	202	202	202	203	204
205	204	203	204	206	207	206	204	203	203	204	204	204	205	205	205
206	204	203	204	205	205	204	202	200	204	205	204	206	206	203	202
205	205	206	205	204	203	204	204	205	204	202	203	205	204	203	205
207	206	204	203	202	203	206	209	212	212	209	205	204	202	203	207
208	205	202	201	204	208	212	214	216	221	218	209	202	203	208	214
205	204	203	203	206	211	216	220	222	223	219	210	204	204	210	219
204	204	202	201	201	207	217	224	229	224	217	212	208	203	205	213
204	204	203	202	203	207	213	217	221	223	221	217	213	208	206	207
203	203	207	212	216	215	209	204	201	218	226	221	218	219	216	210

Setelah mendapatkan nilai *Grayscale*, selanjutnya melakukan kompresi dengan menggunakan metode FELICS.

- a. Membuat tabel frekuensi kemungkinan kemunculan nilai piksel.

Tabel 2. Frekuensi Kemunculan Nilai Piksel

No	Piksel L s/d H	Frekuensi	Nilai Biner	Jumlah Biner	Jumlah Bit * Frek
1.	200	1	11001000	8	8
2.	201	6	11001001	8	48
3.	202	15	11001010	8	120
4.	203	37	11001011	8	296
5.	204	83	11001100	8	664
6.	205	44	11001101	8	352
7.	206	17	11001110	8	136
8.	207	8	11001111	8	64
9.	208	5	11010000	8	40
10.	209	4	11010001	8	32
11.	210	3	11010010	8	24
12.	211	1	11010011	8	8
13.	212	4	11010100	8	32
14.	213	2	11010110	8	16
15.	214	1	11010111	8	8
16.	215	1	11011000	8	8
17.	216	3	11011001	8	24
18.	217	3	11011010	8	24
19.	218	3	11011010	8	24
20.	219	3	11011011	8	24
21.	220	1	11011100	8	8
22.	221	4	11011101	8	32
23.	222	1	11011110	8	8
24.	223	2	11011111	8	16
25.	224	2	11100000	8	16
26.	226	1	11100010	8	8
27.	229	1	11100101	8	8
Total Jumlah Bit x Frekuensi					2048

Frekuensi nilai kemunculan nilai piksel pada tabel di atas memiliki nilai total jumlah bit yang dikalikan dengan frekuensi kemunculan nilai piksel senilai 2048 bit.

- b. Membuat Tabel Kompresi FELICS

Membuat tabel kompresi FELICS mengacu pada proses 3 *decision* tata letak piksel dengan ketentuan sebagai berikut:

Tabel 3. Penentuan posisi Piksel yang akan di *Encoding*

Posisi Nilai Piksel	Biner Code Awal (Biner +)	Encode
$L \leq P \leq H$	00	$N = P - L$
$P < L$	10	$N = L - P - 1$

Posisi Nilai Pixel	Biner Code Awal (Biner +)	Encode
$P > H$	11	$N = P - H - 1$

Selanjutnya menentukan nilai N_1 dan N_2 sebagai nilai piksel terdekat terhadap nilai piksel (P), kemudian menentukan nilai *High* (H) dan *Low* (L) dari nilai " N_1 ; N_2 " yang telah ditentukan. Berikut tabel *codeword*

Tabel 4. Hasil Kompresi Metode FELICS

No.	Piksel (P)	Frek Uensi	N_1 ; N_2	H	L	Posisi Piksel	Rumus Encode	Hasil (Des)
1.	200	1	202;203	202	203	$P < L$	$L - P - 1$	2
2.	201	6	202;203	202	203	$P < L$	$L - P - 1$	1
3.	202	15	205;204	205	204	$P < L$	$L - P - 1$	1
4.	203	37	204;205	204	205	$P < L$	$L - P - 1$	1
5.	204	83	204;205	204	205	$P < L$	$L - P - 1$	0
6.	205	44	208;206	208	206	$P < L$	$L - P - 1$	0
7.	206	17	205;203	205	203	$P > H$	$N = P - H - 1$	0
8.	207	8	201;211	201	211	$P < L$	$L - P - 1$	3
9.	208	5	204;203	204	203	$P > H$	$N = P - H - 1$	3
10.	209	4	206;204	206	204	$P > H$	$N = P - H - 1$	2
11.	210	3	219;209	219	209	$L \leq P \leq H$	$P - L$	1
12.	211	1	206;208	206	208	$P > H$	$N = P - H - 1$	4
13.	212	4	208;206	208	206	$P > H$	$N = P - H - 1$	3
14.	213	2	207;217	207	217	$P < L$	$L - P - 1$	3
15.	214	1	212;209	212	209	$P > H$	$N = P - H - 1$	1
16.	215	1	216;207	216	207	$L \leq P \leq H$	$P - L$	8
17.	216	3	211;212	211	212	$P > H$	$N = P - H - 1$	4
18.	217	3	207;216	207	216	$P > H$	$N = P - H - 1$	9
19.	218	3	221;209	221	209	$L \leq P \leq H$	$P - L$	3
20.	219	3	223;218	223	218	$L \leq P \leq H$	$P - L$	1
21.	220	1	216;214	216	214	$P > H$	$N = P - H - 1$	3
22.	221	4	217;229	217	229	$P > H$	$N = P - H - 1$	3
23.	222	1	220;216	220	216	$P > H$	$N = P - H - 1$	1
24.	223	2	221;224	221	224	$P > H$	$N = P - H - 1$	1
25.	224	2	229;223	229	223	$L \leq P \leq H$	$P - L$	1
26.	226	1	218;221	218	221	$P > H$	$N = P - H - 1$	7
27.	229	1	224;222	224	222	$P > H$	$N = P - H - 1$	4

Berikut tabel lanjutan proses kompresi serta hasil penerapan metode FELICS, ditampilkan pada tabel di bawah ini.

Tabel 5. Hasil Kompresi Dalam Bentuk Biner

No	Piksel	Frekuensi	Hasil (Dec)	Biner (B)	Biner+ (B+)	Biner+ ;Biner	Jlh Biner	Frekuensi * jlh Bit
1.	200	1	2	10	10	1010	4	4
2.	201	6	1	01	10	1001	4	24
3.	202	15	1	01	10	1001	4	60
4.	203	37	1	01	10	1001	4	148
5.	204	83	0	00	10	1000	4	332
6.	205	44	0	00	10	1000	4	176
7.	206	17	0	00	11	1100	4	68
8.	207	8	3	11	00	0011	4	32
9.	208	5	3	11	11	1111	4	20
10.	209	4	2	10	11	1110	4	16
11.	210	3	1	01	00	0001	4	12
12.	211	1	4	100	11	11100	5	5
13.	212	4	3	11	11	1111	4	16
14.	213	2	3	11	10	1011	4	8
15.	214	1	1	01	11	1101	4	4
16.	215	1	8	1000	00	001000	6	48
17.	216	3	4	100	11	11100	5	20
18.	217	3	9	1001	11	111001	6	18

No	Piksel	Frekuensi	Hasil (Dec)	Biner (B)	Biner+ (B+)	Biner+ ;Biner	Jlh Biner	Frekuensi * jlh Bit
19.	218	3	3	11	00	0011	4	12
20.	219	3	1	01	00	0001	4	12
21.	220	1	3	11	11	1111	4	4
22.	221	4	3	11	11	1111	4	16
23.	222	1	1	01	11	1101	4	4
24.	223	2	1	01	11	1101	4	8
25.	224	2	1	01	00	0101	4	8
26.	226	1	7	111	11	11111	5	5
27.	229	1	4	100	11	11100	5	5
Total Jumlah Bit								1083

Dari hasil penerapan metode FELICS di atas memiliki total jumlah bit senilai 1083 bit.

c. Menghitung Kinerja Kompresi

Ukuran citra .gif sebelum dikompresi adalah 2048 piksel. Setelah dilakukan proses pemampatan dengan menggunakan metode FELICS maka dihasilkan nilai piksel sebesar 1083. Sehingga, dapat dihitung analisis rasio kompresinya sebagai berikut:

$$Rc = \frac{\text{Ukuran File Asli}}{\text{Ukuran File Terkompresi}}$$

$$= \frac{2048}{1083}$$

$$= 1,89$$

$$CR = \frac{\text{Ukuran File Terkompresi}}{\text{Ukuran File Asli}} \times 100\%$$

$$= \frac{1083}{2048} \times 100\%$$

$$= 0,52 \times 100\%$$

$$= 52\%$$

$$SS = 100\% - CR$$

$$SS = 100\% - 52\%$$

$$SS = 48\%$$

Berdasarkan nilai rasio kompresi yang dihasilkan nilai efektivitas pemampatan metode FELICS senilai 52%. Sementara untuk nilai *Space Saving* (SS) metode FELICS senilai 48%.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka hasil akhir dari penelitian ini dapat diambil beberapa kesimpulan. Adapun kesimpulan dari penelitian ini bahwa Prosedur kompresi pada citra format .gif dengan menggunakan metode *Fast Efficient & Lossless Image Compression System* (FELICS) telah berhasil dilakukan dan proses kompresi berjalan sesuai dengan teknik kompresi. Penerapan metode *Fast Efficient & Lossless Image Compression System* (FELICS) telah dilakukan dan terbukti bahwa suatu citra dengan format .gif yang memiliki ukuran besar dapat dikompresi dan ukurannya menjadi lebih kecil.

REFERENCES

- [1] P. Digital, U. I. N. Sunan, and K. Yogyakarta, "BEBERAPA APLIKASI DARI PENGOLAHAN CITRA DIGITAL Shofwatul 'Uyun Abstrak," 2008, [Online]. Available: <http://digilib.uin-suka.ac.id/id/eprint/358>.
- [2] A. A. Murni, *Data Processing*. Jakarta: Elex Media Komputindo, 1992.
- [3] A. Yulius, C. Adipianto, F. Teknologi, I. Universitas, and W. Dharma, "PEMANFAATAN LIBRARY FFMPEG UNTUK PERANCANGAN APLIKASI KONVERSI FILE VIDEO KE FORMAT GIF BERBASIS ANDROID," vol. 7, no. 2, pp. 72–82.
- [4] U. Aplikasi, "Analisa Kualitas Citra pada Steganografi," *Program*, pp. 1–9, 2012.
- [5] P. G. Howard and J. S. Vitter, "Fast progressive lossless image compression," *Proc. SPIE*, vol. 2186, p. 98, 1994, [Online]. Available: <http://link.aip.org/link/?PSISDG/2186/98/1>.
- [6] F. A. Sianturi, "Kompresi File Citra Digital Dengan Arithmetic Coding," *J. Tek. Inform. Unika St. Thomas*, vol. 03, no. 1, pp. 45–51, 2018, [Online]. Available: <http://ejournal.ust.ac.id/index.php/JTIUST/article/view/245/263>.
- [7] C. P. Nugraha, R. G. Santosa, and L. Chrisantyo A.A., "Perbandingan Metode Lz77, Metode Huffman Dan Metode Deflate Terhadap Kompresi Data Teks," *J. Inform.*, vol. 10, no. 2, 2015, doi: 10.21460/inf.2014.102.327.
- [8] D. Salomon, G. Motta, and D. Bryant, *Data Compression*. London: USA, 2007.
- [9] F. Savira *et al.*, "Analisa Perbandingan Algoritma DCT, Haar Wavelet, Huffman dan LZW pada Kompresi Citra Digital

Menggunakan Matlab R2013a Rahmad,” *J. Chem. Inf. Model.*, vol. 21, no. 2, pp. 1689–1699, 2017, [Online]. Available: [https://www.oecd.org/dac/accountable-effective-institutions/Governance Notebook 2.6 Smoke.pdf](https://www.oecd.org/dac/accountable-effective-institutions/Governance%20Notebook%202.6%20Smoke.pdf).

- [10] E. Sujatmiko, R. R. , Isnanto, and E. ,Handoyo, “Pemilihan Algoritma Optimal untuk Kompresi Data Citra Iris Mata Manusia,” *Tek. Elektro*, vol. 2, no. 13, p. 1, 2017.
- [11] (Univ. Pend. Wiryawan, et al, “Steganografi Berdasarkan Metode Least Significant Bit (LSB),” *J. Ilmu Komput. Indones.*, no. 1, pp. 34–40, 2019.
- [12] E. S. Wijaya, H. Jamaludin, and D. Watermarking, “Analisis pengujian keamanan watermarking menggunakan metode dct watermarking safety test analysis using dct method,” pp. 21–36, 2021.