

Penerapan Algoritma Fibonacci Codes Dalam Aplikasi Kompresi File Citra Digital

Rizkah Fadillah

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma Medan, Indonesia
Email: rizkahfadillah52@gmail.com

Abstrak—Perkembangan teknologi sangat mempengaruhi penyampaian informasi, dimana sebelumnya pengiriman data hanya dapat dilakukan dengan ukuran file asli sehingga data yang disajikan terbatas dan jika pengiriman data dan informasi dalam jumlah banyak maka diperlukan ruang penyimpanan yang besar sehingga proses menjadi lambat. Misalnya pada suatu file gambar atau citra format JPG. Semakin baik kualitas gambar yang dihasilkan maka semakin besar ukuran hexa yang dibutuhkan untuk mengirim gambar tersebut, sehingga berdampak pada kecepatan dan ruang penyimpanan. Solusi dari permasalahan tersebut adalah mengecilkan ukuran file dengan teknik kompresi. Algoritma kompresi yang digunakan mengkompres file gambar pada penelitian ini adalah fibonacci codes. Selain mudah memahami proses encoding dan proses decoding atau proses dekompresi, fibonacci code juga sangat simple dan efektif untuk mengkompres file gambar. Teknik yang dilakukan dengan menghilangkan beberapa bits pada file gambar tersebut, Ada dua tipe data yang di kompres, yaitu lossy dan lossless. Pada penelitian ini digunakan tipe lossless dimana file yang telah di kompres bisa di kembalikan seperti semula tanpa merubah. Dalam penerapan algoritma fibonacci codes, proses mengompresi file citra dinilai sangat efektif karena dapat menghasilkan nilai dari Compression Ratio (CR) sebesar 62,5% dan nilai Space Saving (Ss) sebesar 37,5% sehingga file gambar setelah di kompresi berukuran 1.805 KB atau 1,76 MB berkurang jauh lebih kecil dari ukuran semula 2,82 MB.

Kata Kunci: Kompresi; Fibonacci Code; citra Digital

Abstract—The development of technology greatly affects the delivery of information, where previously the transmission of data could only be done with the size of the original file so that the data presented was limited and if the transmission of data and information in large quantities was needed a large storage space so that the process became slow. For example, in an image file or image in JPG format. The better the quality of the resulting image, the larger the hexa size needed to send the image, thus impacting speed and storage space. The solution to the problem is to shrink the file size with compression techniques. The compression algorithm used to compress image files in this study is Fibonacci codes. In addition to easily understanding the encoding process and the decoding process or decompression process, Fibonacci code is also very simple and effective for compressing image files. The technique done by removing some bits in the image file, there are two types of data that are compressed, namely lossy and lossless. In this study, a lossless type was used where the compressed file can be purchased as before without changing it. In the application of the Fibonacci codes, the process of compressing image files is considered very effective because it can produce a value of Compression Ratio (CR) of 62.5% and a Space Saving (Ss) value of 37.5% so that the image file after compression measuring 1,805 KB or 1.76 MB is much smaller than the original size of 2.82 MB.

Keywords: Compression; Fibonacci Code; Digital imagery

1. PENDAHULUAN

Perkembangan teknologi sangat mempengaruhi penyampaian informasi, dimana sebelumnya pengiriman data hanya dapat dilakukan dengan ukuran file asli sehingga data yang disajikan terbatas dan jika pengiriman data dan informasi dalam jumlah banyak maka diperlukan ruang penyimpanan yang besar sehingga proses menjadi lambat. Misalnya pada suatu file gambar atau citra dimana semakin baik kualitas gambar yang dihasilkan maka semakin besar ukuran hexa yang dibutuhkan untuk mengirim gambar tersebut, sehingga berdampak pada ukuran file yang harus disimpan pada media penyimpanan.

File gambar berformat JPEG (Joint Photographic Experts Group) merupakan jenis file gambar yang paling populer dan lebih sering digunakan dibanding format lainnya akan tetapi jika di buka maka berkurang kualitas gambar dikarenakan termasuk file terkompresi sehingga digunakan file gambar yang berformat JPG [1].

Pada penelitian ini, permasalahan yang dialami adalah pada saat seseorang diterima atau dinyatakan lulus sebagai mahasiswa baru, pada saat melakukan pendaftaran ulang melampirkan berkas berupa gambar yang berformat .jpg, ukuran file tersebut terlalu besar sehingga file tidak dapat di muat dan proses input file gagal. Solusi untuk masalah diatas yaitu dengan dilakukannya proses kompresi. Saat ini ada beberapa aplikasi kompresi yang dapat digunakan dengan hasil kinerja yang berbeda, beberapa memberikan hasil terbaik karena ukuran file yang dikompresi lebih kecil dari data aslinya. Hal ini terjadi karena dipengaruhi oleh algoritma kompresi yang digunakan, yaitu dengan mengurangi jumlah bit yang sama dari data aslinya. Dalam kompresi data terdapat banyak algoritma di dalamnya yang dapat mengkompresi file gambar dan memiliki hasil yang berbeda-beda, namun kesemuanya didasarkan pada prinsip yang sama yaitu mengecilkan ukuran. Algoritma yang digunakan untuk mengkompres file gambar dalam penelitian ini adalah Fibonacci Codes.

Leonardo Pisano Fibonacci adalah seorang matematikawan Italia, Dia dianggap sebagai matematikawan terbesar Abad Pertengahan, Dia berperan penting dalam menghidupkan kembali matematika kuno. Dalam bukunya, Liber Abaci, ia memperkenalkan Eropa pada notasi Hindu-Arab untuk bilangan tersebut. Bilangan Fibonacci didefinisikan sebagai barisan bilangan yang suku-sukunya merupakan jumlah dari 2 suku sebelumnya. Angka Fibonacci dapat direpresentasikan sebagai urutan angka: 0,1,1,2,3,5,8,13,21,34,55,89,144,... [2] [3].

Berdasarkan penelitian yang dilakukan oleh Dita Anggraini, pada tahun 2021 dengan judul “Penerapan Algoritma Fibonacci Code Pada File Transfer Berbasis Android” menunjukkan Penerapan algoritma Fibonacci Code untuk mengkompres file yang akan dikirim berhasil membuat ukuran file menjadi lebih kecil dengan rasio 48% sehingga dapat memberikan proses transmisi yang cepat dan membantu mengurangi penggunaan paket data internet. Aplikasi transfer file dirancang dan dibangun menggunakan tool Eclipsejuno dan menerapkan algoritma Fibonacci Code sehingga dapat memperkecil ukuran file yang akan dikirim [4].

Pada tahun 2020, seorang penelitian bernama Putri Fitria melakukan Kompres file gambar menggunakan algoritma Rice Codes. algoritma tersebut dapat memperkecil ukuran data suatu file gambar sehingga penyimpanan menjadi lebih kecil dan dapat menghemat ruang penyimpanan [5].

Penelitian yang dilakukan oleh M. Ade Syahputra dan Putri Ramadhani pada tahun 2020 dengan judul “Implementasi Metode Fibonacci Dalam Kompresi Iklan Pada Website PT. Amanah Wisata Travel” bertujuan untuk memperkecil ukuran file pada website PT. Kepercayaan perjalanan dalam bentuk gambar. Jika file yang ditransfer cukup besar, perlu dilakukan kompresi file agar ukuran file diperkecil menggunakan fibonacci codes [6].

Penelitian terdahulu dilakukan oleh Ibrahim Hasan, dkk. pada tahun 2021 tentang Analisis Parameter Kompresi Algoritma Elias Omega Code dan Fibonacci Code Pada File Digital. Hasil akhir kompresi File berekstensi *.EB dan *.FB berisi informasi karakter dan String Bit dari hasil kompresi yang dapat didekompresi, hasil dekompresi file teks digital asli yang disimpan dengan ekstensi file *.txt. Dalam pengujian sistem ini digunakan sampel yaitu string yang terdiri dari satu jenis karakter yang disimpan dalam file teks digital dengan ekstensi *.txt. Dari proses kompresi pada string karakter file digital, algoritma Fibonacci Code lebih unggul dalam hal rasio kompresi rata-rata 1,72, rasio kompresi rata-rata 57% dan penghematan ruang dengan rata-rata 42% [7].

Berdasarkan beberapa penelitian terdahulu diatas, dapat disimpulkan bahwa algoritma fibonacci code sangat efektif digunakan untuk kompresi file berupa teks atau pun gambar, akan tetapi jenis kompresi yang dilakukan yaitu lossy compression untuk gambar. Penulis tertarik mengkompresi file gambar tersebut dengan teknik lossless compression. Dalam penelitian ini teknik kompresi dilakukan agar ukuran file menjadi lebih kecil sehingga file gambar tersebut dapat di input dan tidak membuat ruang penyimpanan menjadi penuh. Proses kompresi dilakukan oleh sistem pada saat mengupload file gambar dan untuk melihat kembali atau menampilkan file gambar yang terupload perlu dilakukan proses dekompresi oleh sistem.

2. METODOLOGI PENELITIAN

2.1 Kerangka Kerja Penelitian



Gambar 1. Kerangka Kerja Penelitian

1. Identifikasi Masalah dilakukan agar dapat memperkirakan dan menguraikan apa yang menjadi sumber masalah kapasitas penyimpanan sehingga membuat ruang penyimpanan penuh serta lambatnya proses pengiriman file gambar.
2. Kajian Pustaka dilakukan untuk memahami objek yang akan diteliti dan metode apa aja yang digunakan melalui berbagai referensi seperti buku, artikel atau jurnal, serta sumber bacaan lainnya.
3. Analisa Proses Kompresi Menerapkan metode Fibonacci Code dalam menyelesaikan masalah yang ada, tahapan analisa ini dilakukan untuk mengetahui proses pengkompresian file gambar menggunakan algoritma Fibonacci Code yang bertujuan untuk memperkecil ukuran data, dimana penerapan algoritma Fibonacci Code ditujukan untuk menghitung codeword dari setiap n yang ada.
4. Analisa Proses Dekompresi Menerapkan metode Fibonacci Code untuk mengembalikan file gambar yang telah dikompresi (diperkecil) menjadi kebentuk file asli sebelum dikompresi menggunakan algoritma Fibonacci Code.
5. Perancangan Sistem Aplikasi menggunakan data yang telah dianalisis ke dalam bentuk yang sederhana, mudah, dan dapat dimengerti oleh pemakai (user).

6. Pengujian, pada tahap ini dilakukan pengujian sistem aplikasi menerapkan algoritma fibonacci code.
7. Laporan Penelitian, pada tahap ini penulis membuat hasil dari penelitian yang sudah penulis lakukan dalam sebuah laporan penelitian.

2.2 Kompresi

Kompresi berarti pengecilan ukuran, suatu aplikasi kompresi informasi atau data yang dilakukan pada citra digital untuk memperkecil bit pada citra sehingga dapat disimpan atau dikirimkan secara efektif dan efisien disebut kompresi citra. Pada umumnya proses kompresi itu dilakukan pada komputer atau mesin lainnya. Kompresi dibagi menjadi 2 bagian yaitu Lossy data-compression proses kompresi yang bermasalah masih diperbolehkan selama tidak merubah pola data atau proses kompresi hanya sebagian data saja, file yang dikompresi berupa gambar dan suara. Kompresi data lossless, merupakan proses kompresi dimana data sebelum dan sesudah di kompresi harus sama dan tidak diperbolehkan ada data yang hilang [8][9] [10].

Dalam mengkompresi data dapat menggunakan beberapa algoritma sebagai metode penyelesaian sebuah studi kasus. Ada berbagai factor yang digunakan dalam menentukan kualitas pengkompresian yaitu:

1. Ratio of Compression (RC)

Ratio of Compression (RC) ialah nilai perbandingan antara ukuran bit data sebelum dikompresi dengan ukuran bit data yang telah dikompresi. Secara sistematis dapat dituliskan sebagai berikut:

$$RC = \frac{\text{Ukuran data sebelum di kompresi}}{\text{Ukuraan data setelah di kompreis}} \dots\dots\dots (1)$$

2. Compression Ratio (CR)

Compression Ratio (CR) ialah persentase perbandingan antara data yang sudah dikompresi dengan data yang belum dikompresi. Secara sistematis dapat dituliskan sebagai berikut:

$$CR = \frac{\text{Ukuran data setelah di kompresi}}{\text{Ukuraan data sebelum di kompreis}} * 100\% \dots\dots\dots (2)$$

3. Redudancy (Rd)

4. Redudancy ialah hasil penilaian nilai rasio dengan hasil nilai rasio kompresi. Secara sistematis dapat dituliskan sebagai berikut :

$$Rd = \frac{\text{File sebelum di kompresi}-\text{File setelah di kompresi}}{\text{Ukuraan file sebelum di kompresi}} * 100\% \dots\dots\dots (3)$$

5. Space Saving (Ss) adalah selisih antara data yang belum dikompresi dengan besar data yang sudah dikompresi

$$S_s = 100\% - C_R \dots\dots\dots (4)$$

2.3 Gambar (Citra)

Sesuatu yang dapat mewakili suatu objek, memiliki warna dan bentuk yang sama dengan aslinya disebut sebagai Gambar. Suatu objek yang di gambarkan atau yang di representasikan adalah citra. Citra merupakan keluaran suatu sistem perekam data analog dan digital yang bersifat optik seperti foto yang dapat langsung disimpan, ditampilkan dan membuat gambar digital berwarna dengan menggabungkan tiga warna dasar seperti merah, hijau, dan biru. Citra digital adalah gambar bidang dua dimensi yang diperoleh dari sinyal-sinyal kontinu pada monitor televisi (analog) yang dapat di proses oleh komputer [11][12].

2.4 Algoritma Fibonacci

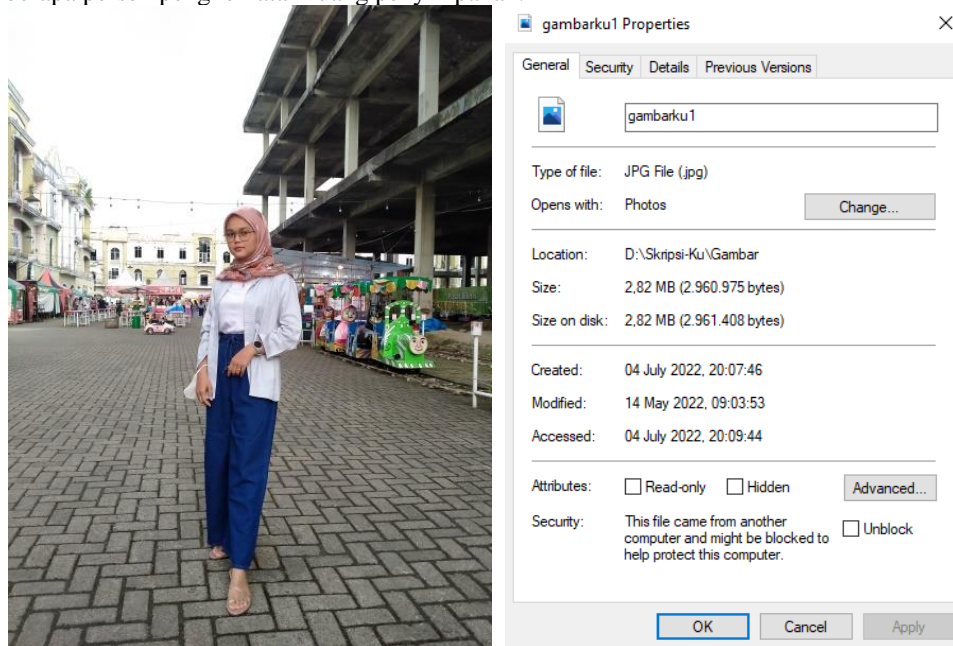
Kode Fibonacci adalah kode length variabel yang berkode pendek jika bilangan bulat-nya terkecil. Kode diakhiri dengan dua bit angka satu atau membubuhkan angka “1” di kanan dari Fibonacci codes sementara dan untuk bit yang ditentukan diperoleh dari jumlah nilai fibonacci yang telah disesuaikan (tidak termasuk bit terakhir) merupakan proses akhir atau proses encoding, Sedangkan menghilangkan angka “1” bagian kanan fibonacci codes tersebut merupakan proses decoding [4][2].

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. Temukan bilangan Fibonacci f terbesar yang lebih kecil atau sama dengan n, kurangkan nilai n dengan 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

3. HASIL DAN PEMBAHASAN

Adapun yang dilakukan dalam penelitian ini adalah mengkompres file citra menerapkan fibonacci codes dengan beberapa sampel data. Sampel yang digunakan berupa bilangan hexadesimal yang akan diproses dari pencarian code fibonacci sementara hingga penentuan codeword. Setelah itu diperoleh nilai bit terkompres, dengan proses pengelompokan bit string maka diperoleh karakter yang akan terbentuk setelah di kompres. Menentukan seberapa besar kinerja proses kompresi, maka dapat dilihat melalui Ratio of Compression, Compression Ratio, Redudancy dan Space Saving berapa persen penghematan ruang penyimpanan.



Gambar 2. Sampel File Gambar

Pada gambar 2 sampel file gambar yang berformat .jpg dengan dengan jenis citra warna dan ukuran awal 2,82 MB di ubah ke nilai hexadesimal menggunakan aplikasi atau software HxD. Berikut 25 nilai Hexadesimal yang digunakan sebagai sampel untuk proses kompresi dan dekompresi :

A2 80 0A 28 A2 80 0A 28 A2 80 0C 53 48 C7 EB 4E
C6 3A 52 1C 73 F4 A0 06 8E

Gambar 3. Sampel Nilai Hexadecimal

3.1 Penerapan Algoritma Fibonacci

Dalam proses kompresi sebelumnya disusun dahulu nilai sampel data berdasarkan frekuensi kemunculan, dimulai dari nilai Hexadesimal dengan frekuensi yang terbesar hingga frekuensi terkecil. Berikut tabel 1 Frekuensi data.

Tabel 1. Frekuensi data

N	Nilai Hexa	Biner	Frekuensi	Bit	Frek x Bit
1	A2	10100010	3	8	24
2	80	10000000	3	8	24
3	0A	00001010	2	8	16
4	28	00101000	2	8	16
5	0C	00001100	1	8	8
6	53	01010011	1	8	8
7	48	01001000	1	8	8
8	C7	11000111	1	8	8
9	EB	11101011	1	8	8
10	4E	01001110	1	8	8
11	C6	11000110	1	8	8
12	3A	00111010	1	8	8
13	52	01010010	1	8	8

N	Nilai Hexa	Biner	Frekuensi	Bit	Frek x Bit
14	1C	00011100	1	8	8
15	73	01110011	1	8	8
16	F4	11110100	1	8	8
17	A0	10100000	1	8	8
18	06	00000110	1	8	8
19	8E	10001110	1	8	8
Total bit sebelum kompresi					200

Berikut pembentukan kode fibonacci sementara berdasarkan nilai n dari tabel 1 yang telah diurutkan berdasarkan frekuensi terbesar hingga frekuensi terkecil dengan perolehan total bit sebelum kompresi berdasarkan perkalian frekuensi dengan bit. Nilai bit diperoleh berdasarkan nilai biner yang diubah dari nilai hexadesimal.

N = 1

1. Bilangan bulat positif $n = 1$
2. Temukan bilangan Fibonacci f yang nilainya lebih kecil atau sama dengan n , kurangkan nilai n dengan f dan catat sisa pengurangan nilai n dengan $f = 1 - 1 = 0$
3. Jika bilangan yang dikurangkan adalah bilangan yang terdapat dalam deret Fibonacci $F(i)$, tambahkan angka "1" dalam kode Fibonacci sementara.

Tabel 2. Pembentukan Kode Fibonacci Sementara Nilai 1

Urutan Bilangan Fibonacci	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)	F(7)	F(8)	F(9)	F(10)
Bilangan Fibonacci	0	1	1	2	3	5	8	13	21	34	55
Kode Fibonacci Sementara	-	-	1								

4. Ulangi langkah ke-2, hingga sisa pengurangan bernilai "0". Sisa pengurangan bernilai 0 sehingga tidak perlu melakukan langkah ke-2. Untuk nilai n berikutnya, tambahkan angka "1" pada fibonacci sementara jika bilangan yang dikurangkan terdapat dalam deret bilangan fibonacci dan tambahkan angka "0" untuk setiap yang tidak terdapat dalam bilangan fibonacci. Dari tabel diatas terbentuk kode Fibonacci sementara untuk angka 1 adalah "1".
5. Tambahkan angka "1" pada posisi paling kanan kode Fibonacci yang telah di bentuk, maka kode fibonacci angka 1 adalah "11". Untuk n berikutnya lakukan sesuai langkah diatas.

Berikut tabel *Codeword* Fibonacci setelah mencari nilai kode fibonacci sementara :

Tabel 3. Urutan *Codeword* fibonacci

N	<i>Codeword</i> Fibonacci
1	11
2	011
3	0011
4	1011
5	00011
6	10011
7	01011
8	000011
9	100011
10	010011
11	001011
12	101011
13	0000011
14	1000011
15	0100011
16	0010011
17	1010011
18	0001011

N	Codeword Fibonacci
19	1001011

Urutan code word fibonacci sementara berdasarkan tabel diatas akan digunakan sebagai inti dari proses pengkompresian dimana sebelumnya untuk memperoleh total bit dilakukan berdasarkan bit dari nilai biner, sedangkan pada proses kompresi ini total bit diperoleh berdasarkan nilai codeword dengan cara mengalikan nilai codeword dengan frekuensi. Berikut tabel 4 proses encoding dalam penerapan fibonacci codes.

Tabel 4. Proses *Encoding*

N	Nilai Hexa	Frek	Pengurangan nilai n	Kode Fibonacci Sementara	Codeword	Bit	Frek x Bit
1	A2	3	1-1	1	11	2	6
2	80	3	2-2	01	011	3	9
3	0A	2	3-3	001	0011	4	8
4	28	2	4-3-1	101	1011	4	8
5	0C	1	5-5	0001	00011	5	5
6	53	1	6-5-1	1001	10011	5	5
7	48	1	7-5-2	0101	01011	5	5
8	C7	1	8-8	00001	000011	6	6
9	EB	1	9-8-1	10001	100011	6	6
10	4E	1	10-8-2	01001	010011	6	6
11	C6	1	11-8-3	00101	001011	6	6
12	3A	1	12-8-3-1	10101	101011	6	6
13	52	1	13-13	000001	0000011	7	7
14	1C	1	14-13-1	100001	1000011	7	7
15	73	1	15-13-2	010001	0100011	7	7
16	F4	1	16-13-3	001001	0010011	7	7
17	A0	1	17-13-3-1	101001	1010011	7	7
18	06	1	18-13-5	000101	0001011	7	7
19	8E	1	19-13-5-1	100101	1001011	7	7
Total bit setelah di kompresi							125

Berdasarkan tabel 4, total bit string setelah di kompres berjumlah 125 bit. Berikut susunan nilai hexa decimal sesuai sebelum dikompresi A2, 80, 0A, 28, A2, 80, 0A, 28, A2, 80, 0C, 53, 48, C7, EB, 4E, C6, 3A, 52, 1C, 73, F4, A0, 06, 8E.

String bit yang dihasilkan nilai Hexadesimal sesuai Codeword 11 011 0011 1011 11 011 0011 1011 11 011 00011 10011 01011 000011 100011 010011 001011 101011 0000011 1000011 0100011 0010011 1010011 0001011 1001011

Selanjutnya, untuk memperoleh karakter baru maka hitung dahulu padding dan flagging. Sebelum menghitung padding dan flagging, tentukan dulu berapa nilai n. agar memperoleh nilai n maka string bit tersebut dibagi per 8 bit, Karena string bit tersebut tidak habis dibagi, maka bisa menggunakan mod.

$125 \text{ Mod } 8 = 5$, Sisa tersebut dinyatakan dengan nilai N.

$N = 5$.

$\text{Padding} = 7 - N + "1"$

$\text{Flagging} = 9 - N$

$\text{Padding} = 7 - 5 + "1"$

$\text{Flagging} = 9 - 5$

$\text{Padding} = 2 \text{ (Jumlah 0)} + "1"$

$\text{Flagging} = 4$, biner dari 4

$\text{Padding} = 001$

$\text{Flagging} = 00000100$

Setelah diperoleh padding "001" dan flagging "00000100" Maka dapat ditambahkan Padding dan Flagging di akhir string bit:

1101100111011101100111011101100011100110101100001110001101001100101110101100000111000011010011001001110100110001011100101100100000100. Total bit menjadi 136 bit.

Setelah penambahan padding dan flagging maka ada penambahan jumlah bit juga, selanjutnya bagi String bit tersebut menjadi per 8 bit agar dapat di mempermudah pembentukan nilai decimal:

11011001 11011110 11001110 11110110 00111001 10101100 00111000 11010011 00101110 10110000 01110000 11010001 10010011 10100110 00101110 01011001 00000100

Ubah string bit diatas yang telah di bagi menjadi per 8 bit ke bentuk decimal dan karakter:

11011001 = 217 = Û

11011110 = 222 = Þ

11001110 = 206 = Î

11110110 = 246 = ö

00111001 = 57 = 9

10101100 = 172 = ˆ

00111000 = 56 = 8
 11010011 = 211 = Ó
 00101110 = 46 = .
 10110000 = 176 = °
 01110000 = 112 = p
 11010001 = 209 = Ñ
 10010011 = 147 = “
 10100110 = 166 = ÷
 00101110 = 46 = .
 01011001 = 89 = Y
 00000100 = 4

Hasil dari proses kompresi menghasilkan karakter berikut : ÜPÎö9-8Ó.°pÑ“÷.Y. Setelah dilakukan proses kompresi tersebut, berikut penentuan kualitas pengkompresian :

1. *Ratio of Compression (RC)*

$$RC = \frac{\text{Ukuran data sebelum di kompresi}}{\text{Ukuraan data setelah di kompreis}} \rightarrow RC = \frac{200}{125} = 1,6$$

2. *Compression Ratio (CR)*

$$CR = \frac{\text{Ukuran data setelah di kompresi}}{\text{Ukuraan data sebelum di kompreis}} * 100\%$$

$$CR = \frac{125}{200} * 100\%$$

$$CR = 62,5\%$$

3. *Space Saving (Ss)*

$$S_S = 100\% - C_R$$

$$S_S = 100\% - 62,5\%$$

$$S_S = 37,5\%$$

Berdasarkan kualitas pengkompresian, ukuran file gambar awal adalah 2,82 MB dengan Compression Ratio sebesar 62,5% dapat menghemat 37,5% ukuran file tersebut terkompres sehingga ukuran file gambar tersebut setelah di kompres berukuran 1805 KB atau 1,76 MB.

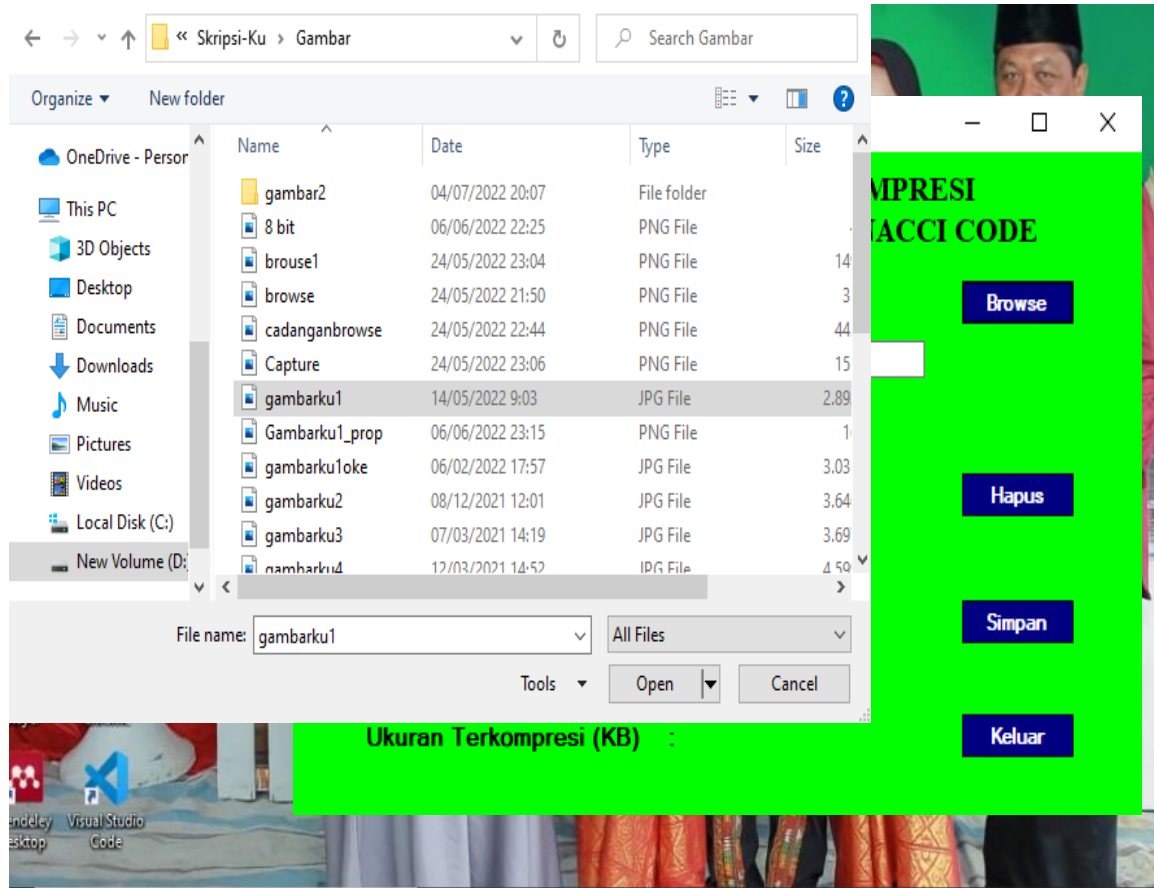
3.2 Tampilan Sistem

1. Tampilan Awal Form Kompresi dan Dekompresi

Gambar 4. Tampilan Form Awal

Berdasarkan gambar 4, terlihat bahwa tampilan awal pada saat dijalkannya aplikasi kompresi formnya masih kosong, belum ada inputan apapun.

2. Tampilan “Browse” Kompresi sebelum input



Gambar 5. Tampilan Browse kompresi sebelum input

Berdasarkan gambar 5, terlihat bahwa tampilan form kompresi dan dekompresi pada saat klik button “Browse” maka akan di arahkan ke bagian penyimpanan directory laptop. User bisa memilih file gambar yang akan di kompresi dengan format gambar sesuai ketentuan yaitu .jpg.

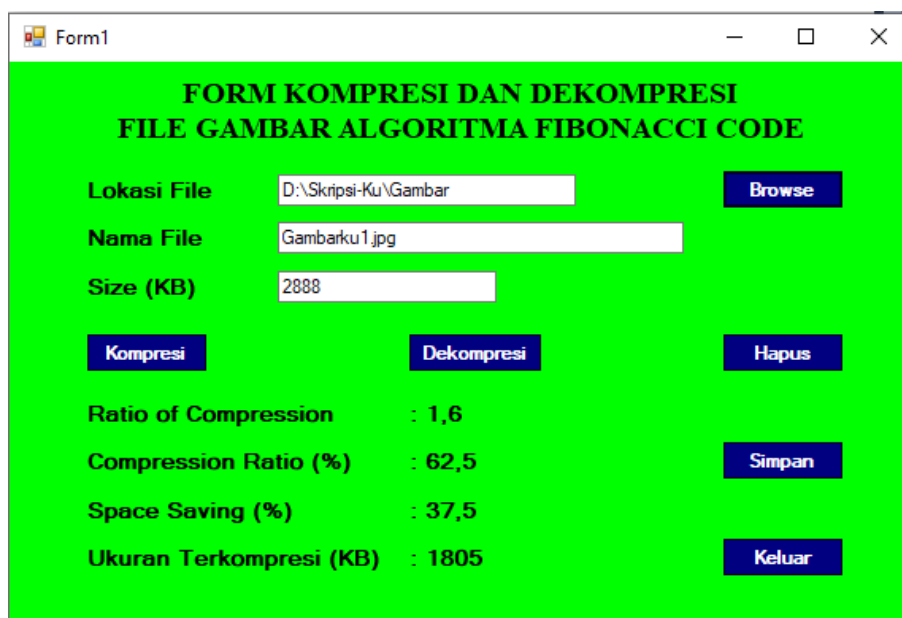
3. Tampilan Setelah Input Browse



Gambar 6. Tampilan setelah input

Berdasarkan gambar 6, terlihat bahwa tampilan form kompresi dan dekompresi setelah file gambar dimasukkan (input), maka terlihat lokasi penyimpanan file, nama file, dan ukuran file tersebut.

4. Tampilan output Kompresi



Gambar 7. Tampilan Output Kompresi

Berdasarkan gambar 7, terlihat bahwa tampilan form kompresi dan dekompresi setelah di klik tombol button “Kompresi” maka akan muncul hasil Ratio of Compression sebesar 1,6, Compression Ratio sebesar 62,5%, Space Saving sebesar 37,5% dan Ukuran file gambar setelah dikompresi menjadi 1805 KB atau 1,76 MB.

4. KESIMPULAN

Kompresi file citra dapat dilakukan dengan mengubah file citra ke dalam bentuk hexadesimal dengan menggunakan aplikasi HxD untuk proses perhitungan kompresi kemudian dapat di terapkan algoritma fibonacci codes. Algoritma fibonacci codes dapat diterapkan dalam mengkompresi file gambar dengan hasil implementasi berupa parameter Ratio of Compression sebesar 1,6, Compression Ratio sebesar 62,5%, Redudancy dan Space Saving sebesar 37,5%. Perancangan aplikasi kompresi file citra menggunakan software microsoft visual studio 2010 dengan hasil pengujian aplikasi kompresi file citra menerapkan algoritma fibonacci codes sangat baik karena menghasilkan ukuran file terkompresi menjadi 1805 KB atau 1,76 MB dari ukuran awal file gambar sebelum terkompresi sebesar 2,82 MB.

REFERENCES

- [1] E. S. Mulyanta, *Pengolahan Digital Image dengan Photoshop CS2*. Yogyakarta, 2006.
- [2] B. Ramadhana, “Implementasi Kombinasi Algoritma Fibonacci Codes Dan Levenstein Codes Untuk Kompresi File Pdf,” vol. 8, no. 2, pp. 67–71, 2021.
- [3] M. Elveny, R. Syah, I. Jaya, and I. Affandi, “Implementation of Linear Congruential Generator (LCG) Algorithm, Most Significant Bit (MSB) and Fibonacci Code in Compression and Security Messages Using Images,” *J. Phys. Conf. Ser.*, vol. 1566, no. 1, 2020, doi: 10.1088/1742-6596/1566/1/012015.
- [4] D. Angraini, “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>
- [5] P. Fitria, “Penerapan Algoritma Rice Codes Pada Aplikasi Kompresi File Gambar,” *J. Comput. Syst. Informatics*, vol. 1, no. 3, pp. 158–165, 2020.
- [6] 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.
- [7] 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.
- [8] B. W. Transform and A. S. Harahap, “Analisis Dan Implementasi Kompresi File Citra Menggunakan Algoritma,” vol. 1, no. 1, pp. 6–12, 2021.
- [9] E. Prayoga and K. M. Suryaningrum, “IMPLEMENTASI ALGORITMA HUFFMAN DAN RUN LENGTH ENCODING PADA APLIKASI KOMPRESI BERBASIS WEB,” *J. Ilm. Teknol. Infomasi Terap.*, vol. 4, no. 2, pp. 92–101, 2018, doi: 10.33197/jitter.vol4.iss2.2018.154.
- [10] T. P. Sari, S. D. Nasution, and R. K. Hondro, “Penerapan Algoritma Levenstein Pada Aplikasi Kompresi File Mp3,” *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 2, no. 1, pp. 439–443, 2018, doi: 10.30865/komik.v2i1.946.
- [11] B. D. Raharja and P. Harsadi, “Implementasi Kompresi Citra Digital Dengan Mengatur Kualitas Citra Digital,” *J. Ilm. SINUS*, vol. 16, no. 2, pp. 71–77, 2018, doi: 10.30646/sinus.v16i2.363.
- [12] N. Z. Munantri, H. Sofyan, and M. Y. Florestiyanto, “Aplikasi Pengolahan Citra Digital Untuk Identifikasi Umur Pohon,” *Telematika*, vol. 16, no. 2, p. 97, 2020, doi: 10.31315/telematika.v16i2.3183.