

Implementasi Algoritma Redundant Pattern Encoding Untuk Menyisipkan Pesan Teks Ke Dalam Gambar dan Algoritma Fibonacci Codes Untuk Kompresi

Frayogi Aditiya*

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

Email: ^{1,*}frayogi.aditiya@gmail.com

Email Penulis Korespondensi: frayogi.aditiya@gmail.com

Abstrak—Informasi merupakan kumpulan data yang berupa teks yang telah disatukan yang bersifat publik atau rahasia. Data rahasia merupakan data yang berisi tentang sesuatu atau yang tidak untuk dipublikasikan. Salah satu cara untuk mengamankan data tersebut adalah dengan memanfaatkan teknik *Steganografi*. Hal ini memberikan manfaat yang luar biasa untuk menjaga kerahasiaan data. Di dalam steganografi terdapat beberapa metode salah satunya adalah *Redundant Pattern Encoding*. Namun dari proses tersebut, objek yang akan dijadikan sebagai wadah untuk menampung data tersebut akan mengalami perubahan ukuran data. Hal ini dapat menimbulkan kecurigaan banyak pihak yang ingin mengakses data tersebut. Penelitian ini menguraikan bagaimana prosedur yang dilakukan untuk mengompresi objek yang dijadikan sebagai wadah untuk menyisipkan pesan. Proses tersebut menggunakan metode *Fibonacci Codes* yang merupakan salah satu metode kompresi data yang simpel dan sangat sederhana. Hasil dari penelitian ini akan menghasilkan tingkat keamanan data yang lebih kuat.

Kata Kunci: Steganografi; Algoritma Redundant Pattern Encoding; Kompresi; Citra; Fibonacci Codes

Abstract—Information is a collection of data in the form of unified text that is public or confidential. Confidential data is data that contains something or that is not for publication. One way to secure the data is to use *Steganography* techniques. This provides tremendous benefits for maintaining data confidentiality. In steganography there are several methods, one of which is *Redundant Pattern Encoding*. However, from this process, the object that will be used as a container to accommodate the data will experience a change in the size of the data. This can raise the suspicion of many parties who want to access the data. This study describes how the procedure is carried out to compress objects that are used as containers to insert messages. The process uses the *Fibonacci Codes* method which is a simple and very simple data compression method. The results of this research will produce a stronger level of data security.

Keywords: Steganografi; Redundant Pattern Encoding Algorithm; Compression; Image; Fibonacci Codes

1. PENDAHULUAN

Penyalahgunaan informasi merupakan salah satu dampak negatif dari perkembangan teknologi yang berkembang pesat saat ini, oleh karena itu perlu dilakukan pengamanan informasi agar tidak dapat diakses oleh orang-orang yang tidak bertanggung jawab. Informasi-informasi tersebut dapat berupa data pribadi yang tidak dibuat untuk dipublikasikan, data perusahaan penting dan berbagai informasi lain yang bersifat rahasia. Salah satu cara yang dapat digunakan untuk mengamankan informasi tersebut adalah dengan menggunakan metode kriptografi. Steganografi merupakan pengamanan data atau suatu file yang dilakukan dengan cara menyisipkannya ke dalam sebuah wadah atau objek, agar tidak terlihat. Steganografi memiliki teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi misalnya kerahasiaan dan integritas data, serta otentikasi. Kuat lemahnya metode steganografi tidak terletak dari hasil *Enkripsi* atau *Ciphertext*, melainkan terletak pada kunci yang digunakan, oleh sebab itu kunci merupakan jantung dari pertahanan data tersebut agar tidak dapat diakses atau dibobol oleh orang-orang yang tidak bertanggung jawab[1]. Salah satu yang termasuk dalam steganografi adalah Algoritma *Redundant Pattern Encoding*.

Algoritma *Redundant Pattern Encoding* merupakan salah satu algoritma penyisipan pesan pada teknik digital yang biasanya menggunakan media gambar sebagai *cover* dari pesan yang akan disembunyikan[2]. Namun pengamanan menggunakan metode ini dapat dibobol atau dihack oleh pihak lain. Hal ini disebabkan oleh perubahan ukuran objek atau wadah yang digunakan. Jika hal terus berlangsung, maka data yang hendak diamankan akan dapat diketahui dengan gampang oleh peretas dan mengambil data di dalamnya. Salah satu cara yang dapat dilakukan untuk mengatasi kelemahan algoritma *Redundant Pattern Encoding* di atas adalah dengan melakukan kompresi pada objek yang akan dijadikan sebagai wadah untuk menyembunyikan data rahasia. Kompresi citra adalah proses untuk meminimalisasi jumlah bit yang mempresentasikan suatu citra sehingga ukuran data citra menjadi lebih kecil dengan tujuan untuk mengurangi redundansi dari data-data yang terhadap dalam citra sehingga dapat disimpan atau ditransmisikan secara efisien. Salah satu algoritma kompresi yang dapat digunakan adalah algoritma *Fibonacci Codes*.

Algoritma *Fibonacci Codes* merupakan algoritma yang diciptakan oleh Leonardo Pisano Fibonacci yang merupakan seorang ahli matematika Italia. Dia dianggap sebagai Matematikawan terbesar dari abad Pertengahan, Dia berperan penting dalam menghidupkan kembali matematika kuno. *Fibonacci Codes* adalah salah satu objek matematika terkenal. Bilangan *Fibonacci* didefinisikan sebagai barisan bilangan yang suku-sukunya merupakan penjumlahan 2 suku sebelumnya. Bilangan *Fibonacci* dapat ditunjukkan sebagai barisan bilangan[3].

Penelitian ini menguraikan bagaimana prosedur yang dilakukan untuk mengkombinasi dua algoritma, yang dimana setelah pesan yang mau diamankan disisipkan ke dalam sebuah citra, maka tahap selanjutnya adalah mengompres ukuran

gambar tersebut. artinya ukuran gambar yang berubah setelah pesan disisipkan dapat dikompres lagi menjadi lebih kecil. Hal ini tidak akan menimbulkan kecurigaan setiap orang dan menganggap bahwa gambar tersebut adalah gambar biasa.

Penelitian sebelumnya yang dilakukan oleh E. N. T. Guruh M Arindra Pratama (2015), Bahwa metode *Redundant Pattern Encoding* sangat membantu dalam mengamankan data dengan cara menyisipkan pesan tersebut ke dalam sebuah objek sehingga tidak dapat diketahui oleh orang lain[1]. Pengujian yang dilakukan pada penelitian tersebut adalah pesan yang akan diamankan akan dikonversi ke dalam bilangan ASCII yang kemudian dikonversi lagi ke dalam bilangan biner. Untuk wadah yang dijadikan sebagai objek yang mana pada penelitian tersebut menggunakan gambar dengan format jpg, akan dicari nilai *pixel*-nya yang kemudian dikonversi ke dalam bilangan biner dan akan diproses berdasarkan algoritma *Redundant Pattern Encoding*.

Hal ini juga diungkapkan oleh Sannawira dkk (2016), pada penelitian tentang penyisipan pesan ke dalam citra berwarna menggunakan *Redundant Pattern Encoding*[2]. Hasil dari penelitiannya adalah *Redundant Pattern Encoding* sangat cocok untuk mengamankan data rahasia. Penelitian ini juga hampir sama dengan penelitian yang dilakukan oleh E. N. T. Guruh M Arindra Pratama (2015), tetapi objek atau wadah untuk menyembunyikan pesan menggunakan gambar berwarna.

Kemudian penelitian terdahulu oleh Stefanus Yerian Elandha (2016), pada penelitian tentang pengamanan data menggunakan kombinasi algoritma *Redundant Pattern Encoding* dan Algoritma *Advanced Encryption Standard* mengatakan bahwa, kombinasi antara kedua algoritma menghasilkan pengamanan yang lebih kuat[4]. Pada penelitian ini, dilakukan proses kombinasi dua algoritma secara bertahap. Tahapan pertama adalah melakukan proses enkripsi dan dekripsi pesan yang akan diamankan menggunakan Algoritma *Advanced Encryption Standard*. Setelah di dapat hasil enkripsi, maka pesan dari hasil enkripsi tersebut akan disembunyikan ke dalam sebuah gambar menggunakan *Redundant Pattern Encoding*.

Penelitian sebelumnya yang dilakukan oleh Kiki Farida (2017), tentang kompresi data file citra menggunakan algoritma *Fibonacci Codes* menghasilkan data file citra dengan ukuran yang lebih kecil dari yang sebelumnya[3]. Pada penelitian ini, file citra yang ukurannya terlalu besar akan dikompresi terlebih dahulu dicari nilai *pixel*-nya, kemudian dilakukan proses *Fibonacci Codes*.

Kemudian penelitian penggunaan algoritma *Fibonacci Codes* juga diperjelas dalam penelitian yang dilakukan oleh Bobby Ramadhan (2021), bahwa algoritma *Fibonacci Codes* juga cocok digunakan untuk kompresi file pdf[5]. Cara kerjanya hampir sama dengan penelitian Kiki Farida (2017), hanya saja objek yang digunakan adalah file pdf.

Pada penelitian yang dilakukan ini, berbeda dengan penelitian sebelumnya. Pada penelitian ini menggunakan algoritma *Redundant Pattern Encoding*, yang mana algoritma ini merupakan salah satu algoritma pengamanan data yang termasuk steganografi. Sedangkan algoritma kedua yang digunakan adalah algoritma *Fibonacci Code* yang merupakan bagian dari kompresi. Kedua algoritma ini akan dikombinasi, sehingga menghasilkan tingkat keamanan yang lebih kuat.

2. METODOLOGI PENELITIAN

2.1 Steganografi

Steganografi adalah seni menyembunyikan pesan di dalam media digital sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu pesan di dalam media tersebut[4]. Kata steganografi (*steganography*) berasal dari bahasa Yunani *steganos* yang artinya “tersembunyi/terselebung” dan *graphein* “menulis” sehingga kurang lebih artinya “menulis (tulisan) terselubung”. Steganografi membutuhkan wadah penampung (*cover*) dan data yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya citra, suara, teks maupun video. Data yang disembunyikan juga dapat berupa citra, suara, teks, atau video. Steganografi merupakan suatu ilmu atau seni dalam menyembunyikan informasi dengan memasukkan informasi tersebut ke dalam pesan lain. Dengan demikian keberadaan informasi tersebut tidak diketahui oleh orang lain. Tujuan dari steganografi adalah menyembunyikan keberadaan pesan dan dapat dianggap sebagai pelengkap dari kriptografi yang bertujuan untuk menyembunyikan isi pesan. Berbeda dengan kriptografi, dalam steganografi pesan disembunyikan sedemikian rupa sehingga pihak lain tidak dapat mengetahui adanya pesan rahasia. Pesan rahasia tidak diubah menjadi karakter aneh seperti halnya kriptografi. Pesan tersebut hanya disembunyikan ke dalam suatu media berupa gambar, teks, musik, atau media digital lainnya dan terlihat seperti pesan biasa. Pada masa kini, steganografi lebih banyak digunakan pada data digital dengan media teks, gambar, audio, dan video. Ada dua buah proses dalam steganografi yakni proses penyisipan pesan dan proses ekstraksi pesan. Proses penyisipan pesan membutuhkan masukan media penyisipan, pesan yang akan disisipkan, dan kunci. Keluaran dari proses penyisipan ini adalah media yang telah berisi pesan. Proses ekstraksi pesan membutuhkan masukan media yang telah berisi pesan. Keluaran dari proses ekstraksi pesan adalah pesan yang telah disisipkan. Aplikasi ini sebelumnya pernah ada namun di buat dengan metode yang berbeda dengan aplikasi yang saya buat saat ini.

2.2 Algoritma *Redundant Pattern Encoding*

Algoritma *Redundant Pattern Encoding* merupakan salah satu teknik pengamanan data yang dapat digunakan dengan cara menyisipkan sebuah pesan yang akan di amankan ke dalam sebuah objek yang merupakan wadah tempat penampungan atau pengamanan yang selanjutnya akan dijadikan sebagai *noise*[4]. Kelebihan dari algoritma ini adalah bertahan terhadap *cropping*, dan kerugiannya adalah tidak dapat menggambar pesan yang lebih besar[9]. Cara kerja metode ini adalah, menyisipkan pesan yang telah diubah atau dikonversi dalam bentuk biner yang kemudian dimasukkan

ke dalam *pixel* yang telah dikonversi juga dalam bentuk biner pada posisi biner yang paling kecil yang terletak di posisi paling kanan. Berikut adalah langkah-langkah penyisipan pesan menggunakan algoritma *Redundant Pattern Encoding* :

1. Pesan yang hendak disisipkan diubah ke dalam bentuk ASCII yang kemudian dikonversi menjadi bilangan biner.
2. Wadah atau objek yang dijadikan sebagai tempat untuk menyisipkan pesan berupa gambar yang kemudian dicari nilai *pixel* dari gambar tersebut dan mengkonversinya ke dalam bilangan biner.
3. Setiap bilangan 2 bilangan biner pesan disisipkan ke dalam *pixel* citra tersebut dimulai dari *most significant bit* (MSB) atau dimulai dari bilangan biner bagian kiri pesan yang kemudian dimasukkan ke dalam bilangan biner *pixel* gambar yang berada di posisi kanan.

2.3 Kompresi

Kompresi adalah proses pengubahan sekumpulan data menjadi suatu bentuk kode atau simbol untuk menghemat tempat penyimpanan dan waktu[11]. Ada terdapat dua jenis pemampatan data, yaitu pemampatan berkehilangan (*lossy compression*) dan pemampatan tanpa kehilangan.

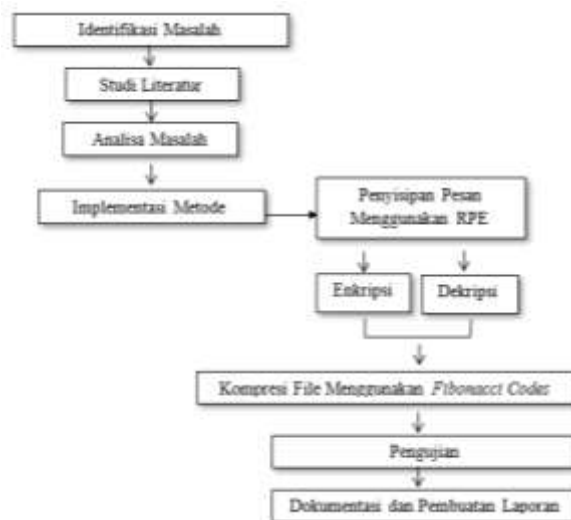
2.4 Fibonacci Codes

Fibonacci Codes merupakan universal code yang dapat mengkodekan integer positif menjadi biner dalam bentuk *codeword*. Semua token akan diakhiri dengan angka “11” dan tidak akan mengandung angka “11” sebelum akhir *codeword*-nya[11]. Kode *fibonacci* dikhususkan untuk mengkodekan bilangan bulat sehingga kadang kala disebut juga dengan representasi *integer fibonacci*. Untuk melakukan pengkodean suatu bilangan bulat n dengan *fibonacci codes* dapat dilakukan dengan tahapan sebagai berikut :

1. Carilah suatu 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” pada $i = 2$, dalam kode fibonacci yang akan dibentuk.
4. Ulangi langkah 2, tukar nilai n dengan sisa pengurangan nilai n dengan f sampai sisa pengurangan nilai n dengan F adalah 0.
5. Tambahkan angka “1” pada posisi paling kanan kode fibonacci yang akan dibentuk.

2.5 Kerangka Kerja Penelitian

Kerangka kerja penelitian merupakan suatu cara atau langkah-langkah yang terstruktur yang dilakukan oleh peneliti dalam rangka untuk mengumpulkan informasi atau data dan melakukan investigasi terhadap data yang telah didapat. Penelitian ini menguraikan tahap-tahap yang dilakukan dalam penyisipan pesan ke dalam gambar menggunakan *Redundant Pattern Encoding* dan proses kompresi menggunakan *Fibonacci Codes*. Secara Umum, Kerangka yang dilakukan dalam penelitian ini digambarkan pada bagan diagram di bawah ini :



Gambar 1. Kerangka Kerja Penelitian

3. HASIL DAN PEMBAHASAN

Steganografi merupakan salah satu teknik pengamanan data yang cara kerjanya adalah dengan menyisipkannya ke dalam sebuah objek atau wadah artinya data yang rahasia yang akan diamankan akan disisipkan ke dalam sebuah wadah atau objek yang telah disiapkan. Teknik pengamanan ini dapat digunakan untuk mengamankan data. selain itu teknik pengamanan ini semakin berkembang dan digunakan oleh banyak orang karena tingkat keamanan yang dihasilkan lebih

optimal. Hal ini disebabkan oleh wadah atau penampung data rahasia yang digunakan tidak menimbulkan kecurigaan banyak orang. Salah satu teknik steganografi yang dapat digunakan adalah *Redundant Pattern Encoding*. Metode ini sangat baik digunakan untuk mengamankan data, dimana media yang digunakan untuk menyembunyikan data adalah berupa gambar. Kelebihan dari metode ini adalah tahan terdapa *cropping* dan kelemahannya adalah ukuran file yang disisipkan sangat terbatas dan ukuran objek yang akan dijadikan sebagai wadah mengalami peningkatan.

Oleh sebab itu, solusi yang dapat digunakan untuk menyelesaikan permasalahan tersebut adalah dengan pengkombinasian metode. Dalam penelitian ini, metode *Redundant Pattern Encoding* akan dikombinasikan dengan metode *Fibonacci Codes* yang merupakan bagian dari kompresi. Dengan pengkombinasian tersebut, maka tingkat keamanan yang dihasilkan lebih optimal, sehingga tidak menimbulkan kecurigaan.

3.1 Penerapan Metode *Redundant Pattern Encoding*

Dalam pengkombinasian kedua metode, ada beberapa tahapan yang akan dilakukan. Untuk lebih jelasnya. Berikut adalah tahapan penerapan metode :

1. Memilih sampel data yang akan disisipkan. Sampel data yang digunakan dalam penelitian ini adalah *password* E-mail peneliti yang kemudian dikonversi dalam bentuk ASCII. Berikut adalah sampel data yang digunakan dalam penelitian ini :

Tabel 1. Sampel Data

Karakter	Nilai ASCII	Nilai Biner
F	70	01000110
R	82	01010010
A	65	01000001
Y	89	01011001
O	79	01011111
G	71	01000111
I	72	01001000

Kemudian menyiapkan citra berwarna yang akan dijadikan sebagai wadah untuk menyisipkan pesan. Berikut adalah citra berwarna yang akan dijadikan sebagai wadah untuk menyisipkan pesan :



Gambar 2. Citra Berwarna

Setelah itu mengkonversi gambar di atas menjadi nilai-nilai *pixel*. Berikut adalah nilai *pixel* dari gambar di atas :

Tabel 2. Nilai *Pixel* Citra Berwarna

	1	2	3	4	5	6	7	8	9	10	...	500
1	85	84	86	84	86	87	89	87	86	88	...	88
2	85	85	83	81	79	79	83	85	85	87	...	79
3	86	83	90	96	103	87	77	92	84	79	...	102
4	86	87	89	78	94	86	89	105	97	98	...	87
5	86	88	79	90	96	103	89	89	79	94	...	93
6	80	85	93	85	87	82	94	105	112	91	...	79
8	80	83	89	89	79	89	107	105	97	98	...	108
8	82	83	85	83	81	79	85	90	96	103	...	112
9	81	79	96	93	89	89	83	85	83	81	...	89
10	80	82	83	87	89	90	82	103	98	108	...	107
...
500	89	80	84	89	191	102	101	100	108	101	...	152

2. Proses penyisipan

Pada tahapan ini akan dilakukan proses penyisipan dengan mengganti nilai biner dari setiap *pixel*. Artinya pada proses ini nilai *pixel* akan dikonversi ke dalam bilangan biner. Pesan yang akan disisipkan dalam penelitian ini adalah berjumlah 7 karakter dimana setiap karakter terdiri dari 8 bit bilangan biner yang berarti ada 56 bit bilangan biner yang akan disisipkan ke dalam citra. Jumlah *pixel* yang akan disisipkan oleh pesan adalah berjumlah 56 *Pixel*. *Pixel* yang 56 ini akan dikonversi ke dalam bilangan biner. Berikut adalah nilai *pixel* yang akan dikonversikan :

85 = 01010101	85 = 01010101	86 = 01010110
84 = 01010100	85 = 01010101	83 = 01010011
86 = 01010110	83 = 01010011	90 = 01011010
84 = 01010100	81 = 01010001	96 = 01100000
86 = 01010110	79 = 01001111	103 = 01100111
87 = 01010111	79 = 01001111	87 = 01010111
89 = 01011001	83 = 01010011	77 = 01001101
87 = 01010111	85 = 01010111	92 = 01011100
86 = 01010110	85 = 01010111	84 = 01010100
88 = 01011000	87 = 01010111	79 = 01001111
86 = 01010110	86 = 01010110	80 = 01010000
87 = 01010111	88 = 01011000	85 = 01010101
89 = 01011001	79 = 01001111	93 = 01011101
78 = 01001110	90 = 01011010	85 = 01010101
94 = 01011110	96 = 01100000	87 = 01010111
86 = 01010110	103 = 01100111	82 = 01010010
89 = 01011001	89 = 01011001	
105 = 01101001	89 = 01011001	
97 = 01100001	79 = 01001111	
98 = 01100010	94 = 01011110	

Deretan bilangan biner pesan yang akan disisipkan adalah 01000110 01010010 01000001 01011001 01011111 01000111 01001000. Proses penyisipan dilakukan dengan cara mengganti nilai terakhir sebelah kanan bilangan biner *pixel* secara berurutan. Berikut adalah proses penyisipan pesan ke dalam nilai *pixel* :

01010100 = 84	01010100 = 84	01010110 = 86
01010101 = 85	01010101 = 85	01010010 = 82
01010110 = 86	01010010 = 82	01011010 = 92
01010100 = 84	01010000 = 80	01100001 = 99
01010110 = 86	01001111 = 79	01100110 = 102
01010111 = 87	01001110 = 78	01010111 = 87
01011001 = 89	01010010 = 82	01001100 = 76
01010110 = 86	01010111 = 87	01011101 = 93
01010110 = 86	01010110 = 86	01010101 = 85
01011001 = 89	01010110 = 86	01001110 = 78
01010111 = 87	01010110 = 86	01010000 = 80
01010111 = 87	01011001 = 89	01010100 = 84
01011001 = 89	01001110 = 78	01011101 = 93
01001110 = 78	01011010 = 90	01010100 = 84
01011111 = 95	01100001 = 97	01010110 = 86
01010110 = 86	01100110 = 102	01010010 = 82
01011000 = 88	01011000 = 88	
01101000 = 104	01011001 = 89	
01100000 = 96	01001110 = 78	
01100011 = 99	01011111 = 95	

Berikut adalah hasil perubahan nilai *pixel* setelah pesan disisipkan :

Tabel 3. Nilai *Pixel* Terbaru

	1	2	3	4	5	6	7	8	9	10	...	500
1	84	85	86	84	86	87	89	86	86	89	...	88
2	84	85	82	80	79	78	82	87	86	86	...	79
3	86	82	92	99	102	87	76	93	85	78	...	102
4	87	87	89	78	95	86	88	104	96	99	...	87
5	86	89	78	90	96	102	88	89	78	95	...	93

	1	2	3	4	5	6	7	8	9	10	...	500
6	80	84	93	84	86	82	94	105	112	91	...	79
8	80	83	89	89	79	89	107	105	97	98	...	108
8	82	83	85	83	81	79	85	90	96	103	...	112
9	81	79	96	93	89	89	83	85	83	81	...	89
10	80	82	83	87	89	90	82	103	98	108	...	107
...
500	89	80	84	89	191	102	101	100	108	101	...	152

3.2 Penerapan Metode *Fibonacci Codes*

Proses kompresi dilakukan dengan menggunakan metode *Fibonacci Codes*. Proses kompresi dilakukan pada nilai *pixel* terbaru yang telah disisipkan pesan. Berikut adalah proses kompresi citra yang telah disisipkan pesan :

1. Setiap nilai *pixel* citra dikonversi ke dalam bilangan biner yang diurutkan dari nilai *pixel* yang banyak muncul atau frekuensi nilai terbesar nilai *pixel* sama. Sampel yang digunakan adalah bagian *pixel* yang telah disisipkan pesan. Berikut adalah urutan *pixel* yang banyak muncul :

Tabel 4. Nilai Frekuensi *Pixel*

Piksel	Biner	Bit	Frekuensi	Bit*Frekuensi
86	01010110	8	11	88
84	01010100	8	6	56
89	01011001	8	4	32
85	01010101	8	3	24
87	01010111	8	3	24
83	01010011	8	3	24
78	01001110	8	3	24
88	01011000	8	2	16
80	01010000	8	2	16
79	01001111	8	2	16
96	01100000	8	2	16
82	01010010	8	1	8
91	01011011	8	1	8
97	01100001	8	1	8
102	01100110	8	1	8
76	01001100	8	1	8
92	01011100	8	1	8
94	01011110	8	1	8
105	01101001	8	1	8
99	01100011	8	1	8
90	01011010	8	1	8
103	01100111	8	1	8
95	01011111	8	1	8
93	01011101	8	1	8

Dari tabel di atas diperoleh total jumlah bit adalah 432. Jumlah nilai bit inilah yang akan di kompres agar lebih kecil. Tahapan selanjutnya adalah melakukan proses kompresi dengan ketentuan pada proses *Fibonacci Codes*. Berikut adalah tabel perhitungannya :

Tabel 5. Nilai Frekuensi

Piksel	Code Fibonacci	Bit	Frekuensi	Bit*Frekuensi
86	11	2	11	22
84	011	3	6	18
89	0011	4	4	16
85	1011	4	3	12
87	00011	5	3	15
83	10011	5	3	15
78	01011	5	3	15
88	000011	6	2	12
80	100011	6	2	12
79	010011	6	2	12
96	101011	6	2	12
82	0000011	7	1	7
91	1000011	7	1	7

Piksel	Code Fibonacci	Bit	Frekuensi	Bit*Frekuensi
97	0100011	7	1	7
102	0010011	7	1	7
76	1010011	7	1	7
92	10100011	8	1	8
94	10000011	8	1	8
105	00100011	8	1	8
99	00010011	8	1	8
90	01010011	8	1	8
103	01010011	8	1	8
95	01100011	8	1	8
93	10000011	8	1	8

Jadi total bit yang didapatkan setelah dikompresi adalah 260 bit yang sebelumnya adalah 432. Artinya kompresi telah berhasil dilakukan

Kemudian menghitung hasil kompresi *Fibonacci codes*

Ratio of compression (Rc) = ukuran lama : ukuran baru = 432 : 260 = 1.6

Compression ratio = (ukuran data baru : ukuran data lama)*100% = (260:432)*100% = 60,18%

Space saving = ((432-260)/432)*100% = 39,81%

Tahapan dekompresi merupakan tahapan untuk mengembalikan data yang telah dikompresi artinya kebalikan dari kompresi. Caranya adalah dengan melakukan penambahan angka 1 pada setiap bit yang dikompresi yang kemudian dilakukan pembalikan. Berikut adalah proses dekompresi :

Tabel 6. Hasil Dekompresi

Piksel	Biner	Bit	Frekuensi	Bit*Frekuensi
86	01010110	8	11	88
84	01010100	8	6	56
89	01011001	8	4	32
85	01010101	8	3	24
87	01010111	8	3	24
83	01010011	8	3	24
78	01001110	8	3	24
88	01011000	8	2	16
80	01010000	8	2	16
79	01001111	8	2	16
96	01100000	8	2	16
82	01010010	8	1	8
91	01011011	8	1	8
97	01100001	8	1	8
102	01100110	8	1	8
76	01001100	8	1	8
92	01011100	8	1	8
94	01011110	8	1	8
105	01101001	8	1	8
99	01100011	8	1	8
90	01011010	8	1	8
103	01100111	8	1	8
95	01011111	8	1	8
93	01011101	8	1	8

2. Ekstrak pesan

Proses ekstrak pesan dilakukan dengan cara mengambil nilai biner *pixel* akhir dengan menggantikannya dengan nol (0). Berikut adalah hasil ekstrak pesan :

01010100 = 84	01010100 = 84	01010110 = 86
01010101 = 85	01010101 = 85	01010010 = 82
01010110 = 86	01010010 = 82	01011010 = 92
01010100 = 84	01010000 = 80	01100000 = 98
01010110 = 86	01001111 = 79	01100110 = 102
01010111 = 87	01001110 = 78	01010111 = 87
01011001 = 89	01010010 = 82	01001100 = 76
01010110 = 86	01010111 = 87	01011101 = 93

01010110 = 86	01010110 = 86	01010101 = 85
01011001 = 89	01010110 = 86	01001110 = 78
01010111 = 87	01010110 = 86	01010000 = 80
01010111 = 87	01011001 = 89	01010100 = 84
01011001 = 89	01001110 = 78	01011101 = 93
01001110 = 78	01011010 = 90	01010100 = 84
01011111 = 95	01100001 = 97	01010110 = 86
01010110 = 86	01100110 = 102	01010010 = 82
01011000 = 88	01011000 = 88	
01101000 = 104	01011001 = 89	
01100000 = 96	01001110 = 78	
01100011 = 99	01011111 = 95	

Hasil dari proses ekstrak pesan adalah 01000110 01010010 01000001 01011001 01011111 01000111 01001000. Untuk lebih jelasnya, berikut adalah tabel pesan yang sudah diekstrak :

Tabel 7. Hasil Ekstrak Pesan

Nilai Biner	Nilai ASCII	Karakter
01000110	70	F
01010010	82	R
01000001	65	A
01011001	89	Y
01011111	79	O
01000111	71	G
01001000	72	I

4. KESIMPULAN

Kesimpulan yang dapat diambil dari hasil penelitian analisa dan perancangan pengamanan data berdasarkan algoritma *Redundant Pattern Encoding* untuk menyisipkan pesan teks dan *Fibonacci Codes* untuk kompresi, yaitu Prosedur pengamanan data berdasarkan algoritma *Redundant Pattern Encoding* adalah menyisipkan tiap nilai biner pesan yang telah dikonversi ke dalam nilai biner *pixel* diakhir nilai biner *pixel* secara berurutan, sehingga menghasilkan nilai *pixel* terbaru yang tidak jauh beda dengan sebelumnya. Proses kombinasi metode *Redundant Pattern Encoding* dan *Fibonacci Codes* sangat sederhana, dimana setelah proses penyisipan dilakukan, maka tahapan selanjutnya adalah tahapan kompresi menggunakan algoritma *Fibonacci Codes*. Sebab setelah proses penyisipan, ukuran data menjadi lebih besar dari sebelumnya.

REFERENCES

- [1] E. N. T. Guruh M Arindra Pratama, "Metode Catalan Number dan Double Columnar," *J. Teknol. Yogyakarta*, vol. 4, no. 5, pp. 31–40, 2015.
- [2] R. F. Sannawira and A. S. Purnomo, "Penyisipan Citra Pesan Ke Dalam Citra Berwarna Menggunakan Metode Least Significant Bit dan Redundant Pattern Encoding," vol. 1, no. 1, pp. 39–46, 2016.
- [3] K. F. Simbolon, "Perbandingan Kinerja Algoritma Fibonacci Codes Dan Algoritma Reverse Unary Codes Dalam Kompresi Citra," *Jurnal Ilmu Komput. Dan Inf.*, vol. 1, 2017.
- [4] M. K. Stefanus Yerian Elandha, Magdalena A. Ineke Pakereng, "Perancangan dan Implementasi Steganografi Menggunakan Metode Redundant Pattern Encoding dengan Algoritma AES (Advanced Encryption Standard) Artikel Ilmiah Perancangan dan Implementasi Steganografi Menggunakan Metode Redundant Pattern Encoding dengan Alg," *Kumpul. J. TI*, vol. 1, no. 1, 2016.
- [5] B. Ramdhan, "Implementasi Kombinasi Fibonacci Codes Dan Levenstein Codes Untuk Kompresi File PDF," *INTI*, vol. 8, no. 2, pp. 67–71, 2021.
- [6] A. H. K. Fresly Nandar Pabokory, Indah Fitri Astuti, "Implementasi Kriptografi pengamanan Data pada Pesan Text Isi File Dokumen Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard," *J. Inform. Mulawarman*, vol. 10, no. 1, pp. 1–10, 2015.
- [7] A. M. Hasibuan, "Rancang Bangun Aplikasi Keamanan Data Menggunakan Metode AES pada Smartphone," *Media Inf. Anal. dan Sist.*, vol. 2, no. 1, pp. 29–35, 2017.
- [8] A. Suroso, "Implementasi Keamanan Data dengan Algoritma Simetri Rijndael Menggunakan VB.NET 2008," *COMPETITIVE*, vol. 11, no. 2, pp. 96–97, 2016.
- [9] E. F. Nugraha, I. T. Bandung, and J. G. Bandung, "Meningkatkan Kapasitas Pesan yang disisipkan dengan Metode Redundant Pattern Encoding," *J. Inst. Teknol. Bandung*, vol. 1, no. 11, 2011.
- [10] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: ANDI, 2010.
- [11] A. Gulo, "IMPLEMENTASI ALGORITMA KRIPTOGAFI RABIN DAN FIBONACCI CODES DALAM PENGAMANAN DAN KOMPRESI FILE CITRA," *J. USU*, vol. 1, no. 1, 2018.
- [12] Ihsan and D. P. Utomo, "Analisis Perbandingan Algoritma Even-Rodeh Code Dan Algoritma Subexponential Code Untuk Kompresi File Teks," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 4, no. 1, 2020.

- [13] Lamsah and D. P. Utomo, "Penerapan Algoritma Stout Codes Untuk Kompresi Record Pada Databade Di Aplikasi Kumpulan Novel," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.
- [14] S. R. Saragih and D. P. Utomo, "Penarapan Algoritma Prefix Code Dalam Kompresi Data Teks," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.