

Implementasi Algoritma Gronsfeld Cipher dan Steganografi End Of File Untuk Pengamanan Data

Kadek Vincky Sedana^{a1}, I Komang Ari Mogi^{a2}, Ida Bagus Gede Dwidasmara^{a3}, I Gede Arta Wibawa^{a4},
Cokorda Rai Adi Pramatha^{a5}, Luh Gede Astuti^{a6}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana,
Badung, Bali, Indonesia

¹vinckysedana@gmail.com

²arimogi@unud.ac.id

³dwidasmara@unud.ac.id

⁴gede.arta@unud.ac.id

⁵cokorda@unud.ac.id

⁶lg.astuti@unud.ac.id

Abstract

Information is a collection of data or facts that are organized or processed in a certain way so that they have meaning for both recipients and users. Data and information are very important things to keep their security or confidentiality so that unauthorized parties cannot find out the data or information. This Data Security Application was developed using the Gronsfeld Cipher algorithm and the End Of File Steganography method. This desktop-based application was created with the aim of increasing data security on confidential information so that unauthorized parties cannot find out the contents of the data. This application is expected to be a solution to prevent the theft of important information by unauthorized parties. From the test results, the developed application can receive input, process input and produce output as expected. Testing the quality of the embedded image gets an average PSNR value of 75,543 dB which can be said to be good because an image can be said to be good if it has a PSNR value above 40 dB, this also shows that the developed application can protect the data embedded in the digital image.

Keywords: *Cryptography, Steganography, Gronsfeld Cipher, End Of File, Data Security*

1. Pendahuluan

Pesatnya perkembangan teknologi informasi mempengaruhi semua aspek kehidupan. Salah satunya yaitu dalam pengamanan informasi yang bersifat rahasia. Data atau informasi adalah hal yang sangat penting untuk dijaga keamanan atau kerahasiannya agar pihak lain yang tidak berkepentingan tidak bisa mengetahui data maupun informasi tersebut. Salah satu cara untuk mengamankan data adalah menggunakan teknik kriptografi yaitu ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi [1]. Implementasinya dapat menggunakan banyak metode enkripsi dari enkripsi klasik hingga enkripsi modern.

Gronsfeld Cipher adalah salah satu algoritma kriptografi yang dapat digunakan. *Gronsfeld Cipher* adalah algoritma enkripsi yang menggunakan kunci numerik dan tabel dalam proses enkripsi dan dekripsi, jika teks yang akan dienkripsi atau didekripsi lebih panjang dari kunci yang digunakan, kunci yang digunakan akan diulang dari kiri. Teks yang telah disandikan akan menjadi teracak dan tidak memiliki makna, namun penggunaan teknik kriptografi belum cukup untuk melindungi data, karena teks terenkripsi masih ditampilkan walaupun dalam bentuk simbol yang tidak beraturan [2]. Jika teks yang telah disandikan didapatkan oleh orang lain tentu akan menjadi sebuah ancaman karena teks yang telah disandikan tersebut dapat dianalisa oleh orang yang mengerti tentang kriptografi atau bahkan hanya sekedar mengacak pesan tersebut. Oleh karena itu, diperlukan teknik lain yang dapat digabungkan dengan teknik kriptografi.

Steganografi adalah salah satu teknik yang dapat digabungkan dengan kriptografi, ini adalah teknik menyembunyikan data rahasia pada suatu media sehingga keberadaan data rahasia tidak dapat dideteksi oleh orang lain [3]. Cara ini digunakan untuk menyembunyikan teks yang telah disandikan agar tidak menimbulkan kecurigaan terhadap orang lain, karena secara visual tidak ada yang berubah

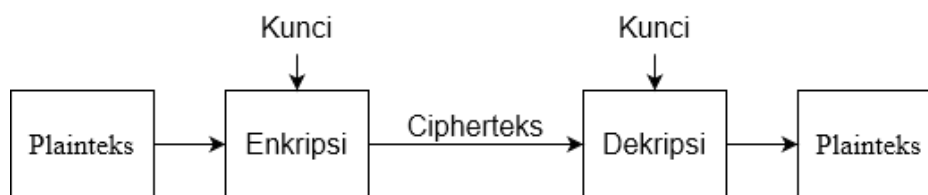
dari media yang ditumpanginya. Salah satu metode steganografi yang dapat digunakan adalah metode *End Of File* (EOF). Metode *End Of File* menyisipkan data terenkripsi ke dalam nilai akhir file gambar, yang hanya memperbesar ukuran file dan memiliki garis tambahan di akhir file gambar sehingga kualitasnya tidak banyak berubah dari citra aslinya.

Berdasarkan permasalahan tersebut, penulis bermaksud untuk mengembangkan sebuah aplikasi yang dapat mengamankan data dengan menggabungkan teknik kriptografi menggunakan algoritma *Gronsfeld Cipher* dan teknik steganografi *End Of File* dengan cara informasi atau pesan rahasia dalam bentuk teks dienkripsi terlebih dahulu menggunakan teknik kriptografi, kemudian file yang telah dienkripsi tersebut akan disisipkan ke dalam sebuah gambar dengan menggunakan teknik steganografi. Aplikasi ini dikembangkan untuk menjadi solusi meningkatkan keamanan data pada informasi-informasi rahasia sehingga pihak yang tidak berkepentingan tidak dapat mengetahui isi dari data yang disimpan.

2. Metode Penelitian

2.1 Kriptografi Algoritma *Gronsfeld Cipher*

Kriptografi adalah ilmu yang mempelajari teknik matematika yang berkaitan dengan aspek keamanan informasi seperti kerahasiaan, integritas data dan otentikasi [3]. Kriptografi terdiri dari dua proses utama: yaitu enkripsi dan dekripsi. Dalam kriptografi, pesan asli disebut *plaintext*, sedangkan pesan yang telah dienkripsi atau disandikan disebut *ciphertext*.



Gambar 1. Skema Proses Enkripsi dan Dekripsi

Pada Gambar 1 ditunjukkan skema dari proses enkripsi dan dekripsi, di mana pada proses enkripsi, plaintext akan dienkripsi menggunakan kunci yang dimasukkan dan akan mendapatkan ciphertext, begitu juga pada proses dekripsi, ciphertext yang dimiliki akan didekripsi menggunakan kunci yang sesuai dan akan menghasilkan plaintext atau pesan asli.

Gronsfeld Cipher adalah algoritma kriptografi yang menggunakan suatu kunci numerik dan tabel dalam proses enkripsi dan dekripsi, jika teks yang akan dienkripsi atau dekripsi lebih panjang dari kunci yang digunakan maka kunci yang digunakan akan diulang dari kiri, kunci yang digunakan juga biasanya cukup pendek misalnya 1324. Caranya adalah dengan mengubah huruf menjadi bilangan desimal, maka plaintext hanya terdiri dari susunan angka, bukan huruf. Kemudian enkripsi menggunakan prinsip yang sama dengan Algoritma *Vigenère* yaitu menggunakan tabel yang hanya berukuran 10x10. Rumus enkripsi *Gronsfeld Cipher*:

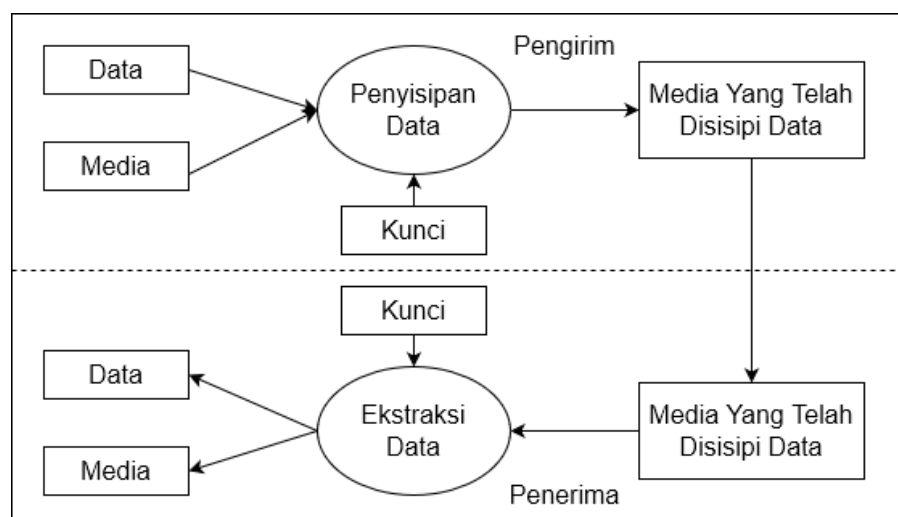
$$C_i = (P_i + K_i) \bmod 26 \quad (1)$$

Rumus dekripsi *Gronsfeld Cipher*:

$$P_i = (C_i - K_i) \bmod 26 \quad (2)$$

2.2 Steganografi Metode *End Of File*

Steganografi adalah suatu metode penyembunyian suatu data atau informasi ke dalam sebuah media atau data digital lain yang tidak diduga oleh orang pada umumnya sehingga orang yang melihatnya tidak curiga [4]. Dalam steganografi, informasi atau pesan yang akan disembunyikan disebut dengan *hidtext* atau *hidden object*, lalu media yang akan digunakan sebagai penampung pesan disebut *covertext* atau *cover object*, lalu media yang telah disisipkan pesan disebut *stegotext* atau *stego object* dan yang terakhir kunci yang digunakan untuk menyisipkan atau mengekstrak pesan disebut *stegokey*



Gambar 2. Skema Proses Steganografi

Pada Gambar 2 ditunjukkan skema proses penyisipan dan ekstraksi. Di mana pada proses penyisipan, data akan disisipkan ke dalam suatu media menggunakan kunci dan akan menghasilkan sebuah *stego-file*. Sedangkan pada proses ekstraksi, *stego-file* yang dimiliki akan diekstraksi menggunakan kunci yang sesuai dan akan menghasilkan data dan media asli sebelum penyisipan.

Metode *End Of File* (EOF) merupakan salah satu teknik yang digunakan dalam steganografi. Teknik ini digunakan dengan cara menambahkan data atau pesan rahasia pada akhir *file* citra. Dalam teknik EOF, data yang disisipkan di akhir diberi tanda khusus sebagai pengenal *start* dari data tersebut dan pengenal akhir dari data tersebut. Dengan metode ini, tidak ada batasan jumlah pesan yang dapat disisipkan. Metode EOF juga tidak mengubah isi *file* yang disisipkan. Ini adalah salah satu keunggulan metode EOF dibandingkan metode lain, karena disisipkan di akhir *file*, pesan yang disisipkan tidak bersinggungan dengan isi *file* yang ditumpangi. Tentu saja, jika ada keunggulan, ada juga kelemahannya, metode EOF ini mengubah ukuran *file* sesuai dengan ukuran pesan yang disisipkan.

2.3 Metode Waterfall

Metode *waterfall* merupakan model pengembangan sistem informasi yang sistematis dan berurutan [5]. Pengembangan aplikasi ini akan menggunakan metode *waterfall*. Metode *waterfall* memiliki beberapa tahapan yaitu analisis, desain, implementasi dan pengujian. Tahap analisis akan menentukan kapabilitas yang harus dimiliki oleh sistem untuk memenuhi apa yang dibutuhkan pengguna dalam menjalankan aplikasi ini.

1. Bahasa pemrograman yang digunakan dalam pengembangan system ini adalah Python.
2. Untuk dapat menjalankan sistem ini, diharuskan untuk menginstall beberapa *library* Python yang digunakan dalam sistem ini.
3. Untuk dapat menjalankan sistem ini, diharuskan untuk memiliki file *user interface* yang telah dibuat dan diletakkan pada direktori yang sama.

Pada tahap desain, kebutuhan yang dibutuhkan dalam pembangunan aplikasi ini adalah sebagai berikut.

a. Halaman *Home*

Halaman home ini memiliki 2 pilihan, penyisipan dan ekstraksi. Pengguna akan dibawa ke halaman penyisipan saat pengguna memilih menu penyisipan, dan sebaliknya pengguna akan dibawa ke halaman ekstraksi saat pengguna memilih menu ekstraksi.

b. Halaman Penyisipan

Pada halaman penyisipan ini, *user* akan melakukan proses penyisipan yang di mana *user* diminta untuk memilih file teks berekstensi .txt yang ingin diamankan dan *file* citra digital berekstensi .png yang akan digunakan sebagai media penyisipan. *User* juga diminta untuk memasukkan kunci yang akan digunakan pada proses enkripsi file teks sebelum disisipkan ke

dalam *file* citra digital. User juga diminta untuk memilih tempat penyimpanan untuk *file* citra digital yang telah disisipi pesan rahasia. Setelah semua dirasa sudah benar, *user* dapat mengklik tombol submit dan proses penyisipan akan dilakukan.

c. Halaman Ekstraksi

Pada halaman ekstraksi ini, *user* akan melakukan proses ekstraksi yang di mana *user* diminta untuk memilih *file* citra digital berekstensi .png yang telah disisipi pesan rahasia. *User* juga diminta untuk memasukkan kunci yang akan digunakan pada proses dekripsi setelah pesan di dalam *file* citra digital berhasil diekstraksi. *User* juga diminta untuk memilih tempat penyimpanan untuk *file* teks yang akan diekstraksi. Setelah semua dirasa sudah benar, *user* dapat mengklik tombol submit dan proses ekstraksi akan dilakukan.

3. Hasil dan Pembahasan

Implementasi pembangunan aplikasi dibuat berdasarkan tahap desain yang akan diterjemahkan ke dalam sebuah program.

3.1. Pengkodean

Implementasi sistem ini adalah aplikasi dapat mengenkripsi pesan, menyisipkan pesan ke dalam citra digital, mengekstrak pesan dari dalam citra digital, dan mendekripsi pesan terenkripsi.

a. Tampilan Antarmuka Sistem



Gambar 3. Tampilan Halaman Home

Pada Gambar 3 menunjukkan tampilan halaman *home*. Halaman *home* merupakan tampilan awal dari aplikasi yang dikembangkan. Ada dua menu utama di halaman ini, menu penyisipan yang membawa pengguna ke halaman penyisipan saat ditekan, dan menu ekstraksi yang membawa pengguna ke halaman ekstraksi saat ditekan. Pada halaman ini juga terdapat tombol *exit* yang jika ditekan akan mengarahkan pengguna untuk keluar dari sistem.

Pada Gambar 4 menunjukkan tampilan halaman penyisipan. Halaman penyisipan merupakan tampilan yang akan dilihat pengguna ketika memilih menu penyisipan pada halaman *home*. Pada halaman ini pengguna diminta untuk memasukkan 4 buah masukan yang diperlukan dalam proses penyisipan yaitu *file* teks, citra digital, kunci, dan *path* untuk menyimpan *output* nanti. Pada halaman ini juga terdapat tombol *back* yang dapat digunakan untuk kembali ke halaman *home*.



Gambar 4. Tampilan Halaman Penyisipan



Gambar 5. Tampilan Halaman Ekstraksi

Pada Gambar 5 menunjukkan tampilan halaman ekstraksi. Halaman ekstraksi merupakan tampilan yang akan dilihat pengguna ketika memilih menu ekstraksi pada halaman *home*. Pada halaman ini pengguna diminta untuk memasukkan 3 buah masukan yang diperlukan dalam proses ekstraksi yaitu *stego-image*, kunci dan *path* untuk menyimpan *output* nanti. Pada halaman ini juga terdapat tombol *back* yang dapat digunakan untuk kembali ke halaman *home*.

3.2. Pengujian

Data penelitian yang akan digunakan untuk menguji sistem ini adalah 6 *file* citra digital dan 5 *file* teks. Data tersebut dapat dilihat pada Tabel 1 dan Tabel 2.







Pada Tabel 1, isi dari setiap file teks diambil secara random, yang membedakan adalah jumlah karakter pada setiap filenya, di mana tiap filenya akan ditambahkan karakter sebanyak 700. Dimulai dari file teks 1 memiliki jumlah karakter 700, file teks 2 memiliki jumlah karakter 1400, file teks 3 memiliki jumlah karakter 2100, file teks 4 memiliki jumlah karakter 2800, dan file teks 5 memiliki jumlah karakter 3500.

Pengujian kualitas hasil penyisipan dilakukan untuk mengetahui bagaimana kualitas citra setelah disisipkan pesan rahasia. Pengujian dilakukan dengan menggunakan metode *Peak Signal Noise Ratio* (PSNR) dengan membandingkan citra asli dan citra hasil penyisipan sesuai dengan data penelitian yang telah dijelaskan di atas. Semakin kecil nilai MSE maka semakin mirip citra dengan citra aslinya, di mana setiap piksel berada pada posisi yang sama. Tentunya semakin kecil nilai MSE maka semakin besar nilai PSNR, maka dapat dikatakan semakin besar nilai PSNR dan semakin kecil nilai MSE berarti penyisipan dapat dikatakan berhasil dilakukan dengan baik, dan begitu juga sebaliknya.

Tabel 1. Data Teks

No.	Nama File (.txt)	Panjang Karakter	Ukuran File
1	Teks1	700	699 bytes
2	Teks2	1400	1.36 KB
3	Teks3	2100	2.05 KB
4	Teks4	2800	2.73 KB
5	Teks5	3500	3.41 KB

Tabel 2. Data Citra Digital

No.	Nama File (.png)	Resolusi	Ukuran File	Preview
1	Citra1	1080x720	50 KB	
2	Citra2	760x458	72.6 KB	
3	Citra3	1600x1000	328 KB	
4	Citra4	1920x1080	347 KB	
5	Citra5	2560x1707	505 KB	
6	Citra6	2000x1335	828 KB	

Tabel 3. Hasil Pengujian Kualitas Citra

No	Media Penampung (.png)	Objek Penyisipan (.txt)	MSE	PSNR (dB)
1	Citra1	Teks1	0.001	76.943
		Teks2	0.002	73.924
		Teks3	0.003	72.161
		Teks4	0.005	70.918
		Teks5	0.007	69.917
2	Citra2	Teks1	0.003	73.445
		Teks2	0.006	70.328
		Teks3	0.009	68.589
		Teks4	0.012	67.337
		Teks5	0.015	66.378
3	Citra3	Teks1	0.001	80.173
		Teks2	0.001	77.114
		Teks3	0.002	75.306
		Teks4	0.003	74.071
		Teks5	0.003	73.103
4	Citra4	Teks1	0.001	81.070
		Teks2	0.001	78.025
		Teks3	0.002	76.275
		Teks4	0.002	75.029
		Teks5	0.003	74.057
5	Citra5	Teks1	0.001	84.375
		Teks2	0.001	81.367
		Teks3	0.001	79.579
		Teks4	0.001	78.324
		Teks5	0.001	77.340
6	Citra6	Teks1	0.001	82.414
		Teks2	0.001	79.370
		Teks3	0.001	77.613
		Teks4	0.002	76.369
		Teks5	0.002	75.390

Tabel 3 menunjukkan hasil kualitas penyisipan dengan metode PSNR. Tabel tersebut menampilkan informasi berupa nilai MSE dan PSNR dari perbandingan citra asli dan citra hasil penyisipan.

Nilai PSNR tertinggi dari pengujian ini adalah 84.375 pada Citra5 dan Teks1 sedangkan nilai PSNR terendah dari pengujian ini adalah 66.378 pada Citra2 dan Teks5. Dari pengujian ini didapatkan rata-rata nilai PSNR yaitu 75.543 dB, dan dapat dikatakan citra hasil penyisipan memiliki kualitas yang baik karena citra yang baik adalah citra yang memiliki nilai PSNR di atas 40 dB [6].

4. Kesimpulan

Kualitas hasil penyisipan dengan teknik steganografi metode *End Of File* dapat dikatakan baik. Hal ini dibuktikan dengan nilai PSNR pada kualitas citra hasil penyisipan, di mana nilai rata-rata dari keseluruhan pengujian adalah 75.543 dB. Nilai PSNR yang baik berada di atas 40 dB dan menunjukkan kualitas hasil penyisipan yang baik. Berdasarkan pernyataan sebelumnya, aplikasi ini telah berhasil mencapai tujuan dengan mengamankan pesan teks ke dalam sebuah citra digital dan mengekstraknya kembali tanpa kehilangan data sedikit pun.

Aplikasi ini dapat dikembangkan dengan menambah jenis ekstensi *file* yang dapat diproses agar pengguna tidak perlu mengganti ekstensi filenya ketika ingin menggunakan aplikasi ini. Aplikasi ini juga dapat dikembangkan agar dapat dijalankan pada berbagai sistem operasi, sehingga pengguna dapat menggunakan aplikasi ini dengan lebih mudah.

Daftar Pustaka

- [1] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Florida: CRC Press, 1996.
- [2] D. Darwis, Wamiliana, and A. Junaidi, "Proses Pengamanan Data Menggunakan Kombinasi Metode Kriptografi Data Encryption Standard dan Steganografi End Of File," no. 978, pp. 228–240, 2017.
- [3] R. Munir, *Kriptografi*. Bandung: Informatika, 2006.
- [4] R. Munir, *Matematika Diskrit*. Bandung: Informatika, 2012.
- [5] G. Wiro Sasmito, "Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal," *J. Inform. Pengemb. IT*, vol. 2, no. 1, pp. 6–12, 2017.
- [6] A. Solichin, "Mengukur Kualitas Citra Hasil Steganografi," *Mengukur Kualitas Citra Hasil Steganografi*, no. April, pp. 1–4, 2015.