5th Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)
7 September 2024 – Jakarta, Indonesia
Volume 3, Nomor 2, September 2024 - ISSN 2962-8628 (*online*)

PENGAMANAN FILE BERBASIS DESKTOP DENGAN ALGORITMA AES-256 DAN STEGANOGRAFI LSB DI PT SINARMAS SEKURITAS

Muhamad Fadli Bahtiar[1*], Joko Christian Chandra[2]

[1,2]Teknik In*formati*ka, Fakultas Teknologi In*form*asi, Universitas Budi Luhur, Jakarta, Indonesia

E-mail: [1*]2011510365@student.budiluhur.ac.id, [2]joko.christian@budiluhur.ac.id
(* : corresponding author)

Abstrak - Keamanan data merupakan aspek kritis bagi perusahaan yang bergerak di bidang keuangan, seperti PT Sinarmas Sekuritas. Dalam lingkungan digital yang semakin kompleks, risiko pencurian dan penyalahgunaan data terus meningkat, menuntut perlindungan yang lebih kuat dan efektif. Permasalahan utama yang dihadapi adalah bagaimana mengamankan file konfigurasi yang sensitif dari potensi akses tidak sah ketika dikirimkan kepada auditor atau pihak ketiga. Rumusan masalah yang muncul mencakup cara implementasi algoritma enkripsi AES-256 untuk mengamankan file serta bagaimana metode Least Significant Bit (LSB) dapat digunakan untuk menyembunyikan file teks ke dalam gambar secara aman. Penelitian ini bertujuan untuk mengembangkan aplikasi desktop berbasis Java yang menggabungkan algoritma AES-256 dengan metode steganografi LSB, guna mengenkripsi dan menyembunyikan file konfigurasi. Metode penelitian yang digunakan adalah pendekatan *waterfall*, yang melibatkan tahap-tahap mulai dari analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian aplikasi. Hasil penelitian menunjukkan bahwa aplikasi yang dikembangkan mampu mengenkripsi dan menyembunyikan data dengan efektif, serta dapat memulihkan file asli tanpa kehilangan data saat proses *decode* dilakukan. Kontribusi penelitian ini adalah menyediakan solusi keamanan data praktis yang meningkatkan perlindungan in*form*asi sensitif di lingkungan kerja yang kritis, seperti di PT Sinarmas Sekuritas, dengan memanfaatkan kombinasi teknik kriptografi dan steganografi. Penelitian ini juga memberikan referensi untuk pengembangan lebih lanjut dalam bidang keamanan data.

Kata Kunci: Keamanan Data, Enkripsi Data, Perlindungan File, Integritas Data, *Platform* Desktop, Aplikasi Java.

DESKTOP BASED FILE SECURITY WITH AES-256 ALGORITHM AND LSB STEGANOGRAPHY AT PT SINARMAS SEKURITAS

Abstract *- Data security is a critical aspect for companies engaged in the financial sector, such as PT Sinarmas Sekuritas. In an increasingly complex digital environment, the risk of data theft and misuse continues to increase, demanding stronger and more effective protection. The main problem faced is how to secure sensitive configuration files from potential unauthorized access when sent to auditors or third parties. The formulation of the problems that arise include how to implement the AES-256 encryption algorithm to secure files and how the Least Significant Bit (LSB) method can be used to hide text files in images securely. This study aims to develop a Java-based desktop application that combines the AES-256 algorithm with the LSB steganography method, to encrypt and hide configuration files. The research method used is the waterfall approach, which involves stages ranging from needs analysis, system design, implementation, to application testing. The results of the study show that the developed application is able to encrypt and hide data effectively, and can recover the original file without losing data during the decode process. The contribution of this research is to provide a practical data security solution that enhances the protection of sensitive information in critical work environments, such as at PT Sinarmas Sekuritas, by utilizing a combination of cryptography and steganography techniques. This research also provides references for further development in the field of data security.*

Keywords: *Data Security, Data Encryption, File Protection, Data Integrity, Desktop Platform, Java Applications.*

1. PENDAHULUAN

PT Sinarmas Sekuritas merupakan perusahaan yang bergerak di dalam sektor keuangan dalam layanan sekuritas dan investasi. Tim IT Infra mengirim file kepada auditor atau pihak ke-tiga masih menggunakan file sharing yang mana rentan terhadap terhadap terjadinya pencurian data dan dibaca oleh pihak lain, yang dapat menyebabkan kerugian terhadap pemilik data tersebut yang kebanyakan data penting baik milik perusahaan maupun customer. Sebagai perusahaan yang berfokus pada sekuritas dan investasi, PT Sinarmas Sekuritas perlu menerapkan kriptografi dan steganografi dalam pengembangan sistem-sistem mereka untuk memastikan kerahasiaan dan integritas data yang mereka tangani. Yang membedakan dari penelitian sebelumnya yaitu penelitian ini menggunakan kombinasi teknik kriptografi dan steganografi, penelitian sebelumnya hanya fokus pada satu teknik keamanan, baik itu kriptografi atau steganografi. Sehingga kombinasi tersebut memberikan lapisan keamanan tambahan yang penting dalam situasi di mana data sangat sensitif.

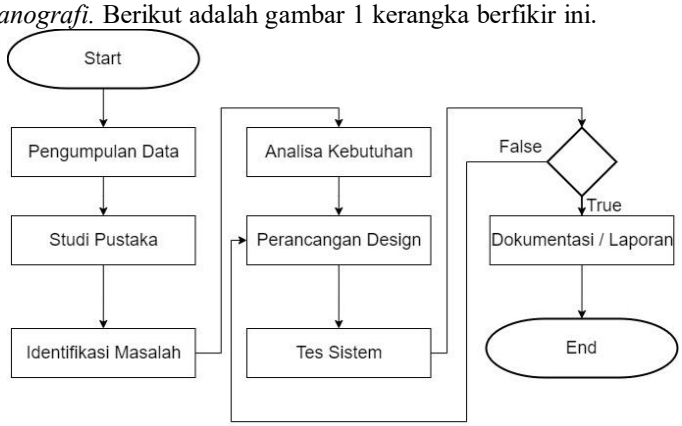
Tujuan penelitian ini adalah untuk mengamankan file dengan Algoritma AES-256 dan Metode LSB untuk mengenkripsi file backup konfigurasi yang berbentuk *.txt supaya tidak dibaca oleh pihak yang tidak bertanggung

jawab. Manfaat penelitian dilihat dari beberapa sisi yaitu Keamanan Data: Implementasi kriptografi dengan algoritma AES-256 dan Metode LSB dapat memberikan tingkat keamanan yang tinggi untuk data yang diolah oleh PT Sinarmas Sekuritas; Pengembangan Aplikasi: Memberikan panduan dalam mengembangkan aplikasi keamanan data berbasis java desktop dengan menggunakan teknik enkripsi dan steganografi; dan Penelitian di Bidang Keamanan Data: menambah referensi mengenai kombinasi algoritma enkripsi dan steganografi dalam pengamanan data.

2. METODE PENELITIAN

2.1. Kerangka Proses Pemikiran

Kerangka berpikir berikut merupakan serangkaian bagan-bagan yang menggambarkan alur dari proses pembuatan aplikasi *Steganografi.* Berikut adalah gambar 1 kerangka berfikir ini.



Gambar 1. Kerangka Proses Pemikiran

a. Pengumpulan Data adalah mengumpulkan data dari sebuah pokok permasalahan dari topik yang diangkat oleh peneliti, yaitu dengan wawancara.
b. Studi Pustaka, setelah data yang dibutuhkan terkumpul, langkah berikutnya yaitu mencari in*form*asi dan fakta yang real melalui studi pustaka.
c. Identifikasi Masalah, Dari data yang terkumpul, langkah berikutnya adalah mengidentifikasi permasalahannya yang relevan dengan batasannya.
d. Analisis Kebutuhan, Berdasarkan hasil identifikasi masalah, langkah selanjutnya adalah melakukan analisis kebutuhan untuk mendukung dalam perancangan aplikasi Steganografi berdasarkan tinjauan pustaka, ini mencakup kebutuhan materi aplilkasi Steganografi, teori perancangan sistem atau aplikasi interaktif serta template atau *platform* dimana perancangan akan dilakukan.
e. Perancangan Desain, Pada tahap ini, perancangan mencakup tampilan tatap muka pengguna yang mudah digunakan dengan prinsip interaksi manusia dengan komputer, serta struktur menu, dan tombol yang relevan.
f. Tes Sistem, Pada tahap ini, sistem akan di tes untuk memvalidasi hasil penelitian dan perancangan dapat disesuaikan atau di revisi untuk mencapai hasil yang diinginkan. Pada tes sistem ini akan dicoba menggunakan metode *black box* untuk memastikan kualitas sistem.
g. Dokumentasi / Pembuatan Laporan, Tahap ini melibatkan dokumentasi atau pembuatan laporan yang menyajikan hasil penelitian dari awal hingga akhir, serta implementasi dalam bentuk penelitian.

2.2. Analisis Kebutuhan Data

Dalam penelitian ini, data yang digunakan berasal dari backup konfigurasi yang diperoleh dari bagian IT Infra di PT Sinarmas Sekuritas. Data ini mencakup rincian backup konfigurasi dari perangkat firewall, router, dan switch.

2.3. Pengujian Aplikasi

Pengujian aplikasi Steganografi yang dikembangkan untuk PT Sinarmas Sekuritas dilakukan menggunakan metode *black box*. Metode ini fokus pada pengujian fungsi sistem tanpa memperhatikan detail internal perangkat lunak. Data yang digunakan dalam pengujian ini adalah data backup konfigurasi, yang bertujuan untuk memastikan bahwa aplikasi berfungsi sesuai dengan spesifikasinya.

2.4. Metode Penelitian

Pengembangan aplikasi Steganografi ini mengikuti metode *waterfall*, yang mencakup beberapa tahap untuk mengumpulkan data yang dibutuhkan, yaitu:

a. Wawancara (*Interview*), Pertanyaan diajukan secara langsung kepada Bapak Anju, yang bertugas di bagian IT Infra PT Sinarmas Sekuritas, untuk memperoleh in*form*asi yang diperlukan.
b. Pengamatan (Observasi), Pengamatan langsung dilakukan terhadap aktivitas yang dijalankan oleh Bapak Anju di bagian IT Infra.
c. Penelitian Kepustakaan, Studi literatur dilakukan dengan mempelajari berbagai sumber pustaka, termasuk buku, jurnal, dan penelitian ilmiah terkait aplikasi Steganografi data.

2.5. Analisis Data

Setelah data terkumpul, dilakukan analisis untuk memproses in*form*asi yang diperlukan dalam penyusunan laporan ini. Langkah-langkah yang diambil meliputi:
a. Memilih komponen-komponen apa saja yang digunakan dalam aplikasi *Steganografi.*
b. Mengaplikasikan data penelitian ke aplikasi *Steganografi.*

2.6. Analisis Masalah

PT Sinarmas Sekuritas adalah perusahaan yang bergerak di sektor keuangan, khususnya dalam layanan sekuritas dan investasi. Perusahaan ini menyimpan sejumlah data penting, termasuk backup konfigurasi yang dikelola oleh tim IT Network internal. Yang hanya boleh diketahui oleh pihak-pihak tertentu saja. Data tersebut merupakan data mengenai *IP Address, Username, Policy, dll.* Data-data tersebut jika diketahui oleh pihak yang tidak bertanggung jawab maka dapat digunakan untuk hal yang merugikan perusahaan PT Sinarmas Sekuritas. Dimana in*form*asi yang bersifat rahasia di enkrip menggunakan algoritma AES-256 lalu disisipkan kedalam bit terakhir pada gambar (*Least Significant Bit*). Sehingga pihak yang tidak diinginkan melihat sebuah gambar tanpa mengetahui didalamnya terdapat sebuah in*form*asi yang bersifat rahasia.

2.7. Landasan Teori

Algoritma Advanced Encryption Standard (AES) dengan panjang kunci 256 bit (AES-256) merupakan salah satu solusi yang sangat handal untuk mengamankan data. [1] AES-256 dikenal karena kemampuannya dalam mengenkripsi data dengan tingkat keamanan yang tinggi, menjadikannya sangat sulit untuk dipecahkan oleh pihak yang tidak berwenang. Selain enkripsi, metode steganografi juga menawarkan lapisan tambahan dalam perlindungan data. Steganografi adalah teknik menyembunyikan in*form*asi dalam media lain sedemikian rupa sehingga keberadaan in*form*asi tersebut tidak terdeteksi oleh pihak yang tidak berwenang. Metode *Least Significant Bit* (LSB) adalah salah satu teknik steganografi yang efektif, di mana data dapat disisipkan ke dalam file gambar tanpa mengubah kualitas visual gambar secara signifikan. [2]

Pengembangan aplikasi berbasis Java desktop yang mengimplementasikan kombinasi AES-256 dan LSB ini bertujuan untuk memberikan solusi praktis bagi PT Sinarmas Sekuritas dalam mengamankan data sensitif mereka. Dengan memanfaatkan *platform* Java desktop, aplikasi ini diharapkan dapat dengan mudah diintegrasikan ke dalam sistem yang sudah ada, serta memberikan antarmuka yang *user*-friendly bagi pengguna di perusahaan.

Steganografi adalah teknik untuk menyembunyikan in*form*asi, diambil dari kata Yunani "steganos" yang berarti tersembunyi, dan "graphein" yang berarti menulis. Teknik ini memungkinkan data disembunyikan dalam berbagai *format*, seperti teks, gambar, audio, atau video. [3] Dalam konteks digital, steganografi sering digunakan di negara-negara dengan sensor in*form*asi yang ketat atau di mana penggunaan enkripsi dilarang. Dengan steganografi, in*form*asi rahasia dapat disembunyikan dengan cara yang sulit dideteksi.[4]

Di dalam teknologi in*form*asi, steganografi adalah teknik untuk menyembunyikan pesan rahasia dalam media lain seperti gambar, audio, atau video, sehingga pihak lain tidak menyadari adanya pesan tersembunyi tersebut. [5] Dalam steganografi, berbagai jenis berkas dapat digunakan, tetapi yang paling efektif adalah berkas dengan tingkat redundansi tinggi. Redundansi mengacu pada jumlah bit tambahan dalam suatu objek yang memberikan akurasi lebih tinggi daripada yang diperlukan untuk fungsinya. Bit-bit tambahan ini dapat dimodifikasi tanpa menyebabkan perubahan yang signifikan, sehingga manipulasi tersebut sulit terdeteksi. [6]

Metode ini didasarkan pada sistem bilangan biner, yaitu angka 0 dan 1, yang erat kaitannya dengan ukuran 1 bit dan 1 *byte*. Secara umum, 1 *byte* terdiri dari 8 bit, di mana bit pada posisi paling kanan disebut sebagai bit pada posisi LSB (*Least Significant Bit*). [7] Bit pertama dikenal sebagai MSB (*Most Significant Bit*), sementara bit terakhir disebut LSB. Bit LSB ini sering dipilih untuk diganti, karena perubahan pada bit ini hanya mengakibatkan pergeseran nilai *byte* sedikit lebih tinggi atau lebih rendah dari nilai awalnya. [8]

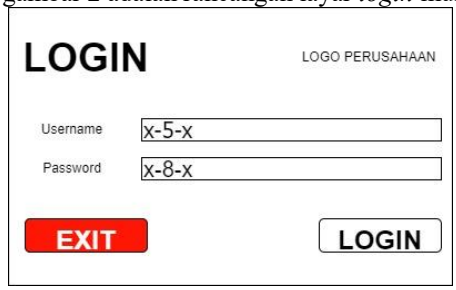
Dalam kriptografi, terdapat berbagai algoritma, salah satunya adalah Algoritma *Advanced Encryption Standard* (AES). AES telah diakui sebagai standar algoritma kriptografi modern yang dipublikasikan oleh NIST dan hingga saat ini belum ada yang mampu memecahkannya. Berdasarkan hal tersebut, aplikasi kriptografi ini dirancang dengan menerapkan algoritma AES-256 untuk enkripsi dan dekripsi dokumen, dengan tujuan melindungi in*form*asi penting dalam dokumen agar tidak jatuh ke tangan yang tidak berwenang. [9] Algoritma AES dipilih karena kemampuannya mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi, yaitu 128 bit, 192 bit, dan 256 bit. Perbedaan panjang kunci ini memengaruhi jumlah putaran dalam algoritma

AES, sehingga memberikan fleksibilitas dalam tingkat keamanan. Teknik pengamanan file menggunakan kriptografi telah menjadi fokus dalam berbagai penelitian. [10]

3. HASIL DAN PEMBAHASAN

3.1. Rancangan Aplikasi

Rancangan ini selalu diperlukan saat merancang sebuah aplikasi. Rancangan layar adalah langkah pertama dalam menciptakan tampilan dan nuansa aplikasi yang dibuat. Tujuan dari rancangan layar adalah untuk membuat tampilan lebih mudah dibuat. Pada gambar 2 adalah rancangan layar *login* muncul ketika awal aplikasi dijalankan

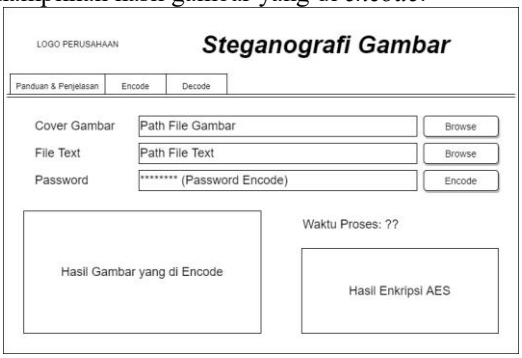


Gambar 2. Rancangan Layar *Login*

Pada gambar 3 adalah rancangan ketika *user* sukses melakukan *login*

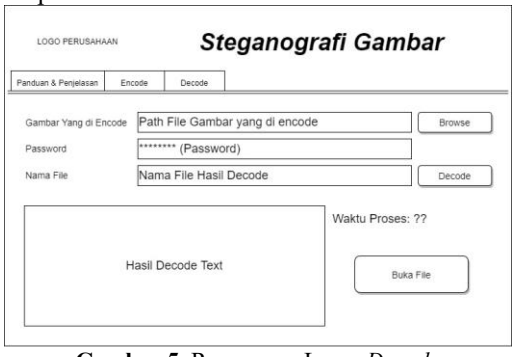
Gambar 3. Rancangan Layar Panduan & Penjelasan

Pada gambar 4 adalah rancangan menu *encode* pada menu *encode* terdapat *form* yang harus di isi oleh *user* dan pada menu *encode* akan menampilkan hasil gambar yang di *encode*.



Gambar 4. Rancangan Layar *Encode*

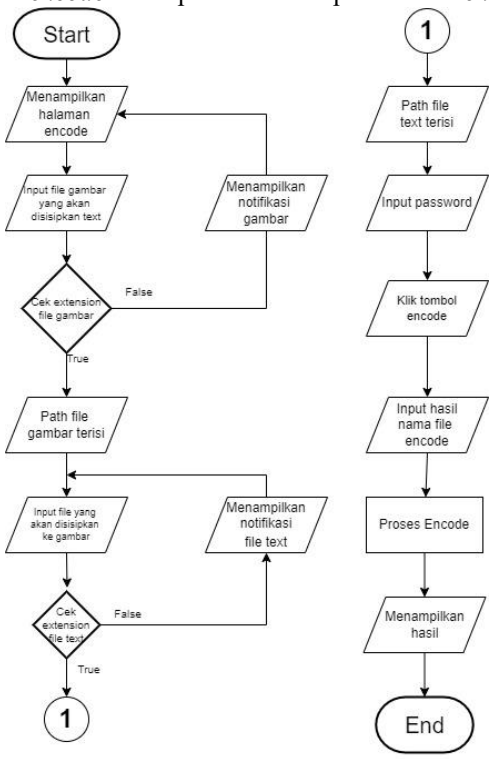
Pada gambar 5 adalah rancangan menu *decode* pada menu *decode* terdapat *form* yang harus di isi oleh *user* dan pada menu *decode* akan menampilkan hasil *decode*.



Gambar 5. Rancangan Layar *Decode*

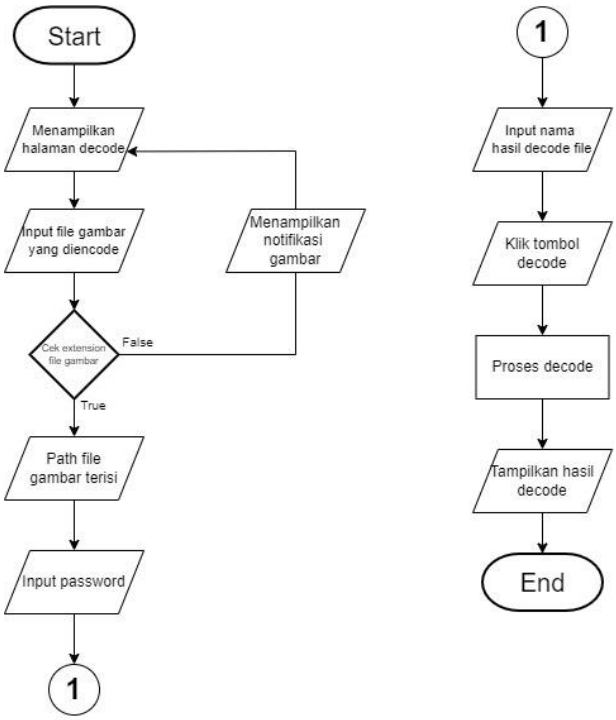
3.2. *Flowchart* Aplikasi

Flowchart Gambar 6 merupakan alur proses dari menu *encode*. dimana *user* menginput gambar yang ingin disisipkan file, file yang ingin disisipkan ke gambar, dan *password encode*. setelah itu *user* mengklik tombol *encode*, dan mengisi hasil nama file *encode*. lalu aplikasi menampilkan hasil *encode*.



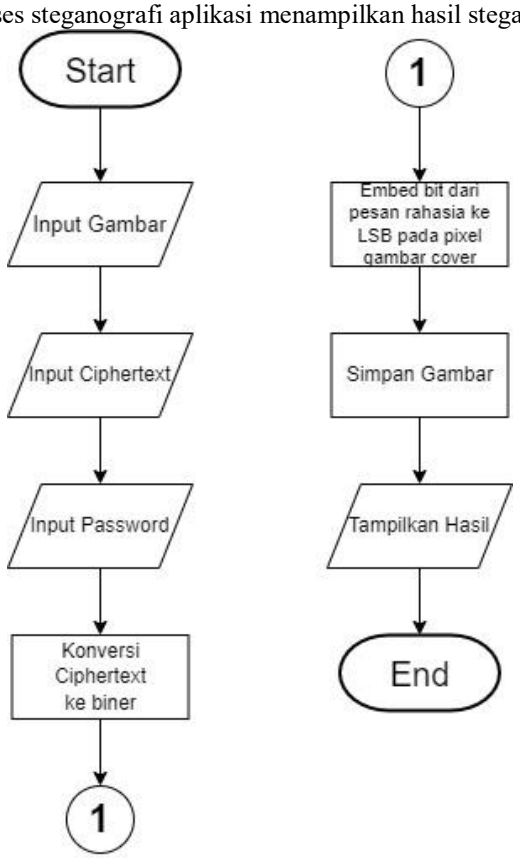
Gambar 6. *Flowchart* Menu *Encode*

Flowchart Gambar 7 merupakan alur proses dari menu *decode*. dimana *user* menginput *form* yaitu gambar yang di*encode*, *password*, dan mengisi hasil nama file *decode*. Setelah itu klik tombol *decode*, lalu aplikasi menampilkan hasil *decode*.



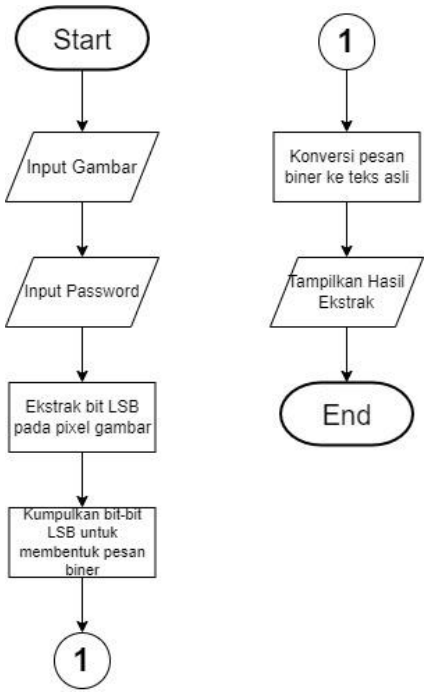
Gambar 7. *Flowchart* Menu *Decode*

Pada *flowchart* Gamabr 8 merupakan alur proses steganografi metode LSB. Dimana setelah *user* meng-klik tombol *encode*, maka proses enkripsi AES-256 terlebih dahulu baru dilakukan setelah itu baru memulai proses steganografi. Setelai selesai proses steganografi aplikasi menampilkan hasil steganografi tersebut.



Gambar 8. *Flowchart* Proses Steganografi Metode LSB

Flowchart Gamabr 9 merupakan alur proses *extract* steganografi metode LSB. Dimana setelah *user* meng-klik tombol *decode*, maka proses *extract* terlebih dahulu baru setelah itu baru memulai proses dekripsi AES-256. Setelai selesai proses *extract* steganografi aplikasi menampilkan hasil *extract* steganografi tersebut.



Gambar 9. *Flowchart* Proses *Extract* Steganografi Metode LSB

3.3. Data Masukan

Setelah seluruh kebutuhan, baik perangkat lunak maupun perangkat keras, terpenuhi, langkah selanjutnya adalah menguji aplikasi yang telah dikembangkan. Pada tahap ini, dijelaskan proses pengujian *Encode* dan *Decode* file. Pengujian ini bertujuan untuk membandingkan hasil file setelah dilakukan proses *encoding*.

3.3.1. Proses *Encode* dan *Decode*

Untuk melakukan proses *encode* maka *user* dapat memilih menu *encode*. *Browse* file gambar yang ingin di disisipkan file txt, lalu klik tombol *Browse* pilih file txt yang ingin disisipkan file ke gambar, masukkan *password* untuk *encode*, lalu klik tombol *Encode*, tunggu hingga proses enkrip AES-256 selesai, lalu masukan nama file klik tombol ok, tunggu hingga proses *encode* ke gambar selesai.

Supaya file yang sudah di*encode* dapat meng*extract* file yang disisipkan ke gambar, maka harus melakukan proses *decode*. Untuk melakukan proses *decode*, maka harus membuka menu *decode*, lalu lakukan seperti tahapan melakukan proses *encode*.

3.3.2. Tabel Pengujian

Dalam pengujian ini, membahas perbandingan proses *encode* dan *decode* pada gambar dengan ekstensi file .jpg, .png, dan .bmp. Pengujian mencakup ukuran awal file yang di-*encode*, panjang *password* yang digunakan, durasi proses *encoding*, durasi proses *decoding*, serta hasil yang dicapai dalam proses *encoding* dan *decoding*.

Berikut Tabel 1 adalah hasil pengujian dari proses *encode.*

Tabel 1. Hasil Pengujian Proses *Encode*

Input File Gambar	Input File Text	*Password*	Ukuran File Txt Input	Ukuran File Gambar Input	Waktu *Encode* (Detik)	Ukuran Hasil *Encode* Gambar	Output File
Google.jpg	SMS-TGR.txt	P@ssw0rd@21	5 KB	176 KB	7,41	996 KB	*encode*1.png
crypto.png	SMS-M2.txt	P@ssw0rd@22	5 KB	25 KB	3,00	156 KB	*encode*2.png
Kripto.bmp	SMS-KLP-GDG.txt	P@ssw0rd@23	5 KB	16 KB	3,34	74 KB	*encode*3.png

Berikut Tabel 2 adalah hasil pengujian dari proses *decode.*

Tabel 2. Hasil Pengujian Proses *Decode*

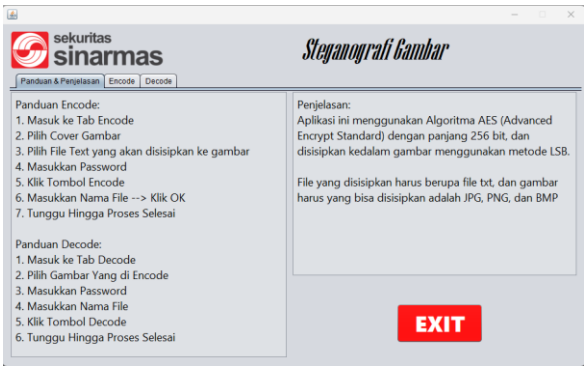
Input File	*Password*	Ukuran File Input	Waktu *Decode* (Detik)	Ukuran Hasil *Decode*	Output File
*encode*1.png	P@ssw0rd@21	996 KB	0,62	5 KB	*decode*1.txt
*encode*2.png	P@ssw0rd@22	156 KB	0,88	5 KB	*decode*2.txt
*encode*3.png	P@ssw0rd@23	74 KB	0,92	5 KB	*decode*3.txt

3.4. Langkah Pengujian

Pengujian ini selalu diperlukan saat membuat sebuah aplikasi. Pengujian sendiri dilakukan mulai dari awal hingga selesai dijalankan. Tujuan dari pengujian adalah untuk mengetahui kekurangan dari aplikasi. Pada gambar 10 adalah tampilan layar *login* muncul ketika awal aplikasi dijalankan.

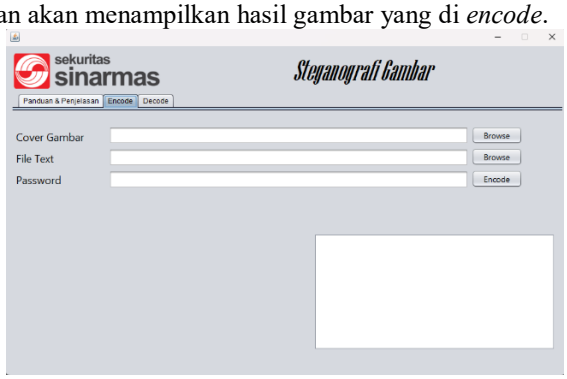
Gambar 10. Tampilan Layar *Login*

Pada gambar 11 adalah tampilan layar panduan dan penjelasan muncul ketika sudah berhasil *login*.



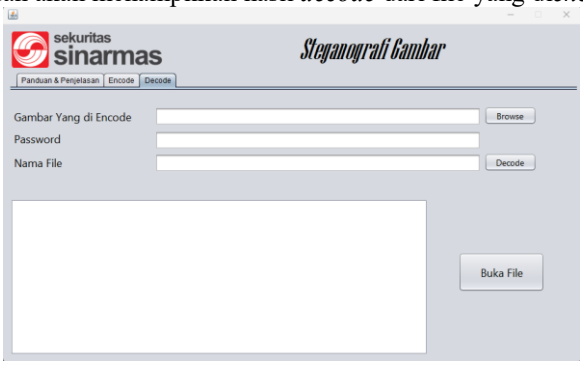
Gambar 11. Tampilan Layar Panduan & Penjelasan

Pada gambar 12 adalah tampilan layar menu *encode* pada menu *encode* terdapat *form* yang harus di isi oleh *user* dan pada menu *encode* dan akan menampilkan hasil gambar yang di *encode*.



Gambar 12. Tampilan Layar *Encode*

Pada gambar 13 adalah tampilan layar menu *decode* pada menu *decode* terdapat *form* yang harus di isi oleh *user* dan pada menu *decode* dan akan menampilkan hasil *decode* dari file yang di*encode*.



Gambar 13. Halaman *Decode*

3.5. Pengujian *Black Box*

Pengujian dilakukan menggunakan *black box* testing bertujuan untuk melihatkan software tentang cara cara beroperasinya, apakah penginputan dan pengeluaran sudah berjalan sesuai dengan yang diharapkan. Berikut Tabel 3 adalah rencana untuk pengujian *Black Box* yang dilakukan:

Tabel 3. Rencana Pengujian *Black Box*

Kelas Uji	Butir Uji	Jenis Pengujian
Encode	Menggabungkan file text dengan gambar	*Black Box*
Decode	*Extract* pesan pada gambar yang di *encode*	*Black Box*

Pada tabel pengujian *encode* sendiri dilakukan untuk memastikan bahwa aplikasi sudah sesuai dengan kebutuhan sistem. Berikut Tabel 4 adalah contoh kasus yang diujikan meliputi:

Tabel 4. Pengujian *Encode*

Hasil Uji (Data Normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih Gambar	Data berjenis .jpg, .png, dan .bmp, yang dapat masuk	Data hanya berjenis .jpg, .png, dan .bmp saja yang dapat masuk kedalam *form*	[X] Diterima [ ] Ditolak

Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data berjenis selain .jpg, .png, dan .bmp, tidak dapat di input kedalam *form Browse* Gambar	Tidak dapat masuk ke inputan *form Browse* Gambar	Pada *form* sudah divalidasi	[X] Diterima [ ] Ditolak

Pada tabel 5 pengujian *decode* berfokus pada kesesuaian dan ketidaksesuaian aplikasi dengan kebutuhan sistem mencakup contoh kasus yang diujikan meliputi:

Tabel 5. Pengujian *Decode*

Hasil Uji (Data Normal)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Gambar yang di *encode* berjenis .png	Data berjenis .png yang dapat masuk	File dengan jenis .png yang dapat masuk kedalam *form decode*	[X] Diterima [ ] Ditolak
Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data berjenis selain .png tidak dapat diinput kedalam *form browse decode*	Tidak dapat masuk ke inputan *form decode*	Pada *form* sudah divalidasi	[X] Diterima [ ] Ditolak

3.6. Hasil dan Pembahasan Uji Coba Aplikasi

Berdasarkan hasil pengujian tersebut aplikasi pada proses *encode* dan *decode* yang telah dilakukan, maka dapat disimpulkan bahwa pengiriman file menggunakan file gambar yang sudah di *encode* dan file tersebut terhapus setelah disisipkan ke gambar. Hasil file *decode* tidak berubah setelah proses *decode*, pada proses *decode* memerlukan gambar yang sudah di*encode* dan *password* untuk mengakses file yang sudah di*encode*. pada aplikasi tersebut ditemukan beberapa kelebihan maupun kekurangan, diantaranya yaitu:

a. Kelebihan Aplikasi
   1. File yang di*encode* tidak dapat dibaca sebelum di*decode* terlebih dahulu.
   2. File yang di*encode* tidak mengalami perubahan atau kerusakan, sehingga tetap konsisten dengan file aslinya.

b. Kekurangan Aplikasi
   1. Semakin besar ukuran file, semakin lama waktu yang dibutuhkan untuk proses *encoding*.
   2. Kecepatan proses *encoding* masih belum optimal.
   3. Aplikasi saat ini hanya mendukung penyisipan file dengan *format* .txt.
   4. Gambar yang bisa disipkan file txt hanya .bmp, .png, dan .jpg.
   5. Dibutuhkan *user* management untuk melihat *activity user* pemakai aplikasi.

4. KESIMPULAN

Penelitian ini bertujuan untuk mengembangkan aplikasi keamanan data berbasis desktop dengan menggabungkan algoritma enkripsi AES-256 dan metode steganografi Least Significant Bit (LSB). Aplikasi ini dirancang untuk mengamankan file konfigurasi sensitif di PT Sinarmas Sekuritas, sebuah perusahaan di sektor keuangan, agar terhindar dari akses tidak sah saat dikirimkan kepada auditor atau pihak ketiga. Metodologi yang digunakan adalah pendekatan *waterfall*, yang mencakup analisis kebutuhan, perancangan, implementasi, dan pengujian aplikasi. Hasil penelitian menunjukkan bahwa aplikasi ini mampu mengenkripsi dan menyembunyikan data secara efektif, serta dapat memulihkan file asli tanpa kehilangan data selama proses *decode*. Kesimpulannya, aplikasi ini menawarkan solusi praktis untuk meningkatkan perlindungan data sensitif dengan kombinasi teknik kriptografi dan steganografi, sekaligus memberikan referensi untuk pengembangan lebih lanjut dalam bidang keamanan data.

Adapun saran yang dapat berguna untuk membuat aplikasi ini dapat bekerja lebih baik lagi antara lain: Kedepannya untuk *login* mengunakan *user* management jadi tidak hanya ada satu *user* saja; Proses *encode* file-file yang berukuran besar dapat dilakukan kompresi file terlebih dahulu, oleh karena itu dikembangkan algoritma kompresi pada aplikasi ini. Supaya proses *encode* dapat berjalan lebih cepat dan hasil *encode* file gambar ukurannya tidak terlalu besar; dan Kedepannya setelah ada *user* management, admin bisa mengetahui aktivitas *user* yang membuka aplikasi tanpa sepengetahuan admin. Seperti *log activity*, fitur tersebut dapat digunakan untuk mengetahui siapa saya yang sudah *login* dan melakukan proses *encode* dan *decode*.

DAFTAR PUSTAKA

[1] A. Pariddudin dan F. Syauqi, "Penerapan Algoritma AES pada QR CODE untuk Keamanan Verifikasi Tiket," Teknois : Jurnal Ilmiah Teknologi In*form*asi dan Sains, vol. 10, no. 2, hlm. 43–52, 2020, doi: 10.36350/jbs.v10i2.87.

[2] F. Rizki Permana, R. Fadillah Setiyanto, dan A. Rafi Fauzi, "Image Steganography Dengan Menggunakan Metode Lsb Pada Python," Jurnal Pendidikan Teknologi In*form*asi, hlm. 1–7, 2023.

[3] B. Siddiqui dan S. Goswami, "A Survey on Image Steganography using LSB Algorithm," SSRN Electronic Journal, hlm. 345–349, 2023, doi: 10.2139/ssrn.4671618.

[4] R. Munir, Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: In*format*ika, 2004. [Daring]. Tersedia pada: https://opac.perpusnas.go.id/DetailOpac.aspx?id=332204

[5] S. Bruce, Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C (cloth). 1996. doi: 10.1109/ISOCC.2015.7401749.

[6] T. Morkel, J. H. P. Eloff, dan M. S. Olivier, "An Overview of Steganography," Advances in Computers, vol. 83, hlm. 51–107, 2011, doi: 10.1016/B978-0-12-385510-7.00002-3.

[7] R. Islamadina, B. Baihaqi, dan M. Sulistriadi, "Analisa Steganografi untuk Citra Bewarna (RGB) Menggunakan Metode Less Significant Bit (LSB)," Jurnal Nasional Komputasi dan Teknologi In*form*asi (JNKTI), vol. 2, no. 1, hlm. 55, 2019, doi: 10.32672/jnkti.v2i1.1058.

[8] A. W. Laksono, S. Suhada, dan A. Zakaria, "Implementasi Metode Least Significant Bit (Lsb) Dalam Teknik Steganografi Pada Citra Digital Menggunakan Matlab," vol. 4, no. 1, 2024.

[9] B. E. Widodo dan A. S. Purnomo, "Implementasi Advanced Encryption Standard Pada Enkripsi Dan Dekripsi Dokumen Rahasia Ditintelkam Polda Diy," Jurnal Teknik In*format*ika (Jutif), vol. 1, no. 2, hlm. 69–77, 2020, doi: 10.20884/1.jutif.2020.1.2.21.

[10] M. Azhari, D. I. Mulyana, F. J. Perwitosari, dan F. Ali, "Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES)," Jurnal Pendidikan Sains dan Komputer, vol. 2, no. 01, hlm. 163–171, 2022, doi: 10.47709/jpsk.v2i01.1390.