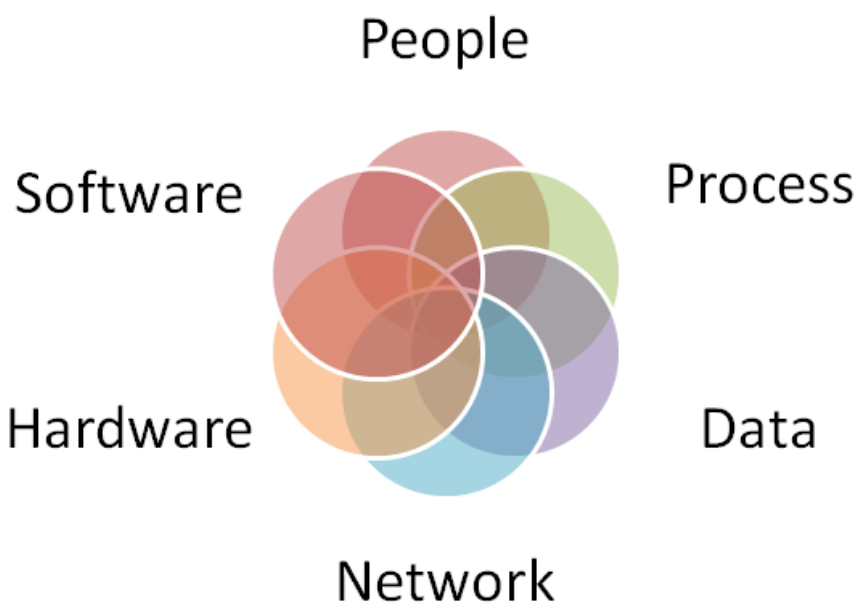
OAJIS
Open Access
Journal of
Information
Systems
is.its.ac.id/pubs/oajis/

ISSN 1979-3979

SISFO

Inspirasi Profesional Sistem Informasi

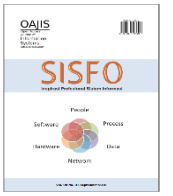


Vol. 08 No. 03 Mei 2019

OAJIS
Open Access
Journal of
Information
Systems
is.its.ac.id/pubs/oajis/

SISFO
Inspirasi Profesional Sistem Informasi

Jurnal Sisfo Vol. 08 No. 03 (2019) i-ii



Sekretariat

Departemen Sistem Informasi – Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember (ITS) – Surabaya
Telp. 031-5999944 Fax. 031-5964965
Email: editor@jurnalsisfo.org
Website: http://jurnalsisfo.org

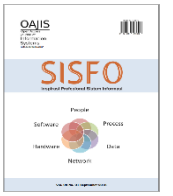
Jurnal SISFO juga dipublikasikan di *Open Access Journal of Information Systems* (OAJIS)

Website: http://is.its.ac.id/pubs/oajis/index.php

OAJIS
Open Access
Journal of
Information
Systems
is.its.ac.id/pubs/oajis/

SISFO
Inspirasi Profesional Sistem Informasi

Jurnal Sisfo Vol. 08 No. 03 (2019) i-ii



Mitra Bestari

Nur Aini Rakhmawati, Ph.D. (Institut Teknologi Sepuluh Nopember)

Rahadian Bisma, M.Kom. ITILF. (Universitas Negeri Surabaya)

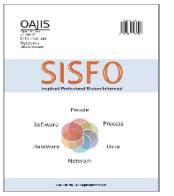
Raras Tyasnurita, S.Kom, M.BA, Ph.D. (Institut Teknologi Sepuluh Nopember)

Satria Fadil Persada, S.Kom, M.BA, Ph.D (Institut Teknologi Sepuluh Nopember)

Sholiq, S.T, M.Kom, M.SA. (Institut Teknologi Sepuluh Nopember)

OAJIS
Open Access
Journal of
Information
Systems
is.its.ac.id/pubs/oajis/

SISFO
Inspirasi Profesional Sistem Informasi
Jurnal Sisfo Vol. 08 No. 03 (2019) iii

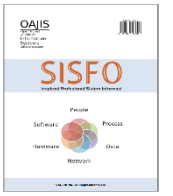


Daftar Isi

Halaman ini sengaja dikosongkan

OAJIS
Open Access Journal of Information Systems
is.its.ac.id/pubs/oajis/

SISFO
Inspirasi Profesional Sistem Informasi
Jurnal SISFO Vol.08 No.03 (2019) 171–194



Integrasi Algoritma Kriptografi *Blowfish* dengan Steganografi LSB untuk Pengamanan Data pada *File* MP3

Bonifacius Vicky Indriyono*

Jurusan Sistem Informasi, Sekolah Tinggi Manajemen Informatika dan Komputer Kadiri

Abstract

Along with the development of information technology, the flow of information dissemination is increasing. This is one reason that information technology users are looking for ways to protect information. In the world of informatics, several ways can be done to maintain information security, including cryptographic techniques and steganography. Cryptography is defined as a technique that learns about how to protect information so that it is safe when sent to interested parties by encoding into forms that cannot be known by unauthorized parties, while steganography is defined as art and science of hiding secret messages so that their existence is unknown by other people. In this study discussed how to secure information using blowfish cryptography and LSB steganography with MP3 media. In addition, software is also produced for the implementation of these two information security techniques. From the results of the implementation it was concluded that in addition to using image media, the process of hiding messages can be done using MP3 media without causing significant changes to the original MP3 file.

Keywords: Cryptography, Blowfish, Steganography, LSB, MP3

Abstrak

Seiring berkembangnya teknologi informasi menyebabkan arus penyebaran informasi semakin meningkat. Hal ini menjadi salah satu alasan pengguna teknologi informasi mencari cara agar informasi terlindungi. Dalam dunia informatika, beberapa cara dapat dilakukan untuk menjaga keamanan informasi, diantaranya dengan teknik kriptografi serta steganografi. Kriptografi diartikan sebagai teknik yang mempelajari tentang cara melindungi informasi agar tetap aman saat dikirimkan ke pihak yang berkepentingan dengan jalan menyandikan ke bentuk yang tidak dapat diketahui maknanya oleh pihak yang tidak berkepentingan, sedangkan steganografi diartikan sebagai seni dan ilmu menyembunyikan pesan rahasia sehingga keberadaannya tidak diketahui oleh orang lain. Dalam penelitian ini dibahas cara melakukan pengamanan informasi menggunakan kriptografi *blowfish* dan steganografi LSB dengan media MP3. Selain itu, dihasilkan pula perangkat lunak untuk implementasi kedua teknik pengamanan informasi tersebut. Dari hasil implementasi disimpulkan bahwa selain menggunakan media gambar, proses penyembunyian pesan dapat dilakukan dengan menggunakan media MP3 tanpa mengakibatkan perubahan yang signifikan pada file MP3 aslinya.

Kata kunci: Kriptografi, Blowfish, Steganografi, LSB, MP3

Histori Artikel: Disubmit 05-03-2019; Direvisi 07-05-2019; Diterima 21-05-2019; Tersedia online 29-05-2019

*Corresponding Author
Email address: bonifaciusvicky@gmail.com (Bonifacius Vicky Indriyono)
https://doi.org/10.24089/j.sisfo.2019.05.003

1. Pendahuluan

Perkembangan teknologi informasi yang semakin pesat dewasa ini, menyebabkan arus pertukaran informasi semakin besar. Tindak kejahatan terhadap akses informasi ini pun semakin berdampak pada kekhawatiran banyak pihak akan keamanan data dan informasi. Hal ini menyebabkan pengguna teknologi informasi berusaha untuk mencari jalan agar informasi terlindungi dari pihak-pihak yang tidak berkepentingan. Dalam dunia informatika, memiliki banyak cara yang dapat dilakukan untuk menjaga keamanan informasi, diantaranya dengan menggunakan teknik kriptografi dan steganografi. Kriptografi dan steganografi sebenarnya memiliki cara kerja yang berbeda namun dua teknik tersebut memiliki korelasi yang erat dalam hal pengamanan data. Kriptografi dapat diartikan sebagai teknik yang mempelajari tentang aspek-aspek dalam keamanan informasi, seperti kerahasian data, keabsahan data, integritas data maupun autentikasi data. Tetapi tidak semua aspek keamanan informasi dapat diselesaikan dengan kriptografi [1]. Dalam proses pengiriman informasi dari satu pihak ke pihak lain, kemungkinan besar kerahasiaan informasi bisa jatuh ke tangan pihak lain yang tidak berhak untuk mengetahui isi informasi tersebut. Untuk menghindari hal tersebut, maka isi informasi dapat di sandikan (enkripsi) ke bentuk yang maknanya sulit bahkan tidak di mengerti oleh orang lain. Teknik kedua adalah dengan steganografi. Steganografi merupakan sebagai teknik yang bisa digunakan untuk menyisipkan pesan khusus melalui media lain sehingga posisi dimana pesan tersebut berada tidak diketahui oleh pihak lain [2]. Untuk melakukan proses penyembunyian pesan ini dibutuhkan dua media yakni media penampung yang akan digunakan sebagai tempat untuk menyembunyikan pesan dan media pesan yang akan disembunyikan isinya. Media penampung ini merupakan komponen yang penting dalam proses steganografi dimana dapat menggunakan gambar (JPEG, PNG, GIF, BMP), suara (WAV, MP3), teks maupun video (AVI, MP4).

Pengamanan data serta informasi dengan menggunakan kombinasi teknik kriptografi dan steganografi ini menjadi hal yang lazim digunakan. Namun tidak jarang pula beberapa dari kombinasi teknik tesebut hanya salah satu teknk saja yang digunakan untuk menjaga kerahasiaan data. Kedua teknik tersebut semuanya dapat dipakai untuk menjaga keamanan inormasi. Perbedaan kedua teknik tersebut adalah, jika pada teknik kriptografi, data yang telah tersandi (*ciphertext*) tetap ditampilkan, sedangkan pada teknik steganografi pesan yang telah tersandi disembunyikan ke dalam media tertentu sehingga keberadaan pesan tersebut tidak terlihat oleh orang lain. Oleh karena itu, pemakaian kombinasi teknik pengamanan data menjadi pilihan agar kerahasiaan data benar-benar terjaga.

Pada penelitian ini, akan dibahas bagaimana melindungi kerahasiaan informasi dengan menggunakan kombinasi teknik kriptografi dan steganografi. Agar fokus pembahasan tidak melebar dan teruji, maka dibuat juga sebuah perangkat lunak yang mampu mengimplementasikan kombinasi teknik kriptografi *blowfish* untuk melakukan proses enkripsi pesan dalam bentuk *file* teks (.txt) dan teknik steganografi *Least Significant Bit* (LSB). Sebagai media penampung pesan, dalam penelitian ini menggunakan *file* MP3. MP3 merupakan format pemampatan audio yang memiliki sifat "menghilangkan". Menghilangkan disini memiliki arti bahwa proses pemampatan akan menghilangkan suara-suara yang keberadaannya kurang atau bahkan tidak signifikan bagi sistem pendengaran manusia. Media *file* MP3 lebih dipilih dibandingkan menggunakan media gambar dengan pertimbangan bahwa manusia lebih suka melakukan hal yang bisa menghibur. Selain itu, penggunaan media MP3 ini bisa mengurangi kecurigaan yang berlebihan dari orang lain. Algoritma kriptografi *blowfish* dipilih dan digunakan dalam penelitian ini karena *blowfish* memiliki kelebihan diantaranya **cepat**, **kompak, sederhana** dan **memiliki tingkat keamanan yang fleksibel** karena memiliki panjang kunci yang beragam, sedangkan untuk teknik steganografi LSB dipilih karena setelah pesan disisipkan ke dalam media penampung, hasil proses tidak akan berdampak pada perubahan ukuran dari media tersebut sehingga hal ini tidak akan menimbulkan kecurigaan bagi orang lain

Menurut Mukhedkar et al. [3], penggunaan algoritma *blowfish* untuk enkripsi dipilih karena *blowfish* memiliki kinerja yang baik dan lebih cepat sedangkan teknik LSB sebagai teknik penyembunyian gambar merupakan teknik yang cepat dan sederhana dalam memanipulasi Bit. Secara umum, prosedur yang dijalankan dalam sistem adalah pertama kali sistem akan melakukan proses enkripsi isi pesan/informasi

sesuai dengan algoritma enkripsi *blowfish*. Setelah isi informasi di enkripsi, kemudian sistem akan melakukan proses penyembunyian isi informasi tersebut ke dalam media audio MP3 yang diawali dengan memilih *file* MP3 terlebih dahulu. Agar isi informasi yang telah dienkripsi dapat terbaca, maka pengguna harus melakukan proses dekripsi pada *file* MP3 terpilih dengan memasukkan kunci dekripsi yang telah ditentukan pada saat proses enkripsi.

Masalah pengamanan data dan informasi merupakan salah satu hal yang wajib diperhatikan oleh pengguna teknologi informasi dengan tujuan agar kerahasiaan pesan benar-benar dapat terjaga. Penelitian ini bertujuan untuk menerapkan enkripsi *blowfish* pada proses enkripsi sebuah *file* yang berisi pesan rahasia. Selain itu, penelitian ini akan mencoba menerapkan kombinasi enkripsi *blowfish* dengan teknik steganografi LSB untuk menyembunyikan *file* enkripsi ke dalam media MP3 dan menganalisa bagaimana pengaruhnya terhadap file MP3 yang telah disisipi pesan baik ukuran, durasi maupun kualitas suaranya. *File* pesan yang digunakan berformat *file* teks (.txt). Aplikasi dibangun menggunakan *compiler* Delphi 2010.

2. Tinjauan Pustaka/Penelitian Sebelumnya

Tinjauan pustaka penelitian sebelumnya ini digunakan untuk mempelajari dan menganalisis hasil dari beberapa penelitian terdahulu yang membahas topik yang identik dengan topik penelitian sekarang serta pustaka-pustaka yang menunjang kegiatan penelitian. Beberapa penelitan tentang penerapan algoritma *blowfish* dan teknik steganografi LSB dengan media MP3 telah dilakukan oleh peneliti sebelumnya diantaranya:

1) Sitinjak, et al. [4] dalam penelitiannya menjelaskan tentang cara menjaga keamanan data dan informasi yang tersimpan dalam file menggunakan prinsip algoritma enkripsi blowfish. Dari penelitian sebelumnya memiliki persamaan dalam hal pemanfaatan teknik dalam algoritma blowfish untuk melakukan enkripsi data, sedangkan perbedaannya terletak pada hasil dari proses enkripsi. Pada penelitian sebelumnya, ekstensi file yang dienkripsi diubah namanya, sedangkan pada penelitian sekarang, file hasil enkripsi tidak diubah nama ekstensinya.
2) Thakur dan Kumar [5] dalam penelitiannya menjelaskan tentang analisis performa dari enkripsi DES, AES dan blowfish. Penelitian ini memiliki kesamaan dengan penelitian sekarang dalam hal penerapan algoritma enkripsi blowfish. Perbedaan yang menonjol adalah jika pada penelitian sebelumnya hasil akhir berupa grafik performa pengujian DES, AES dan blowfish, sedangkan pada penelitian yang dilakukan sekarang hasil akhir bukan berupa grafik performa tetapi lebih pada bagaimana menerapkan hasil ekripsi blowfish dan steganografi dalam media MP3.
3) Lubis, et al. [6] menjelaskan tentang bagaimana membuat perangkat lunak untuk menyisipkan file terenkripsi melalui media MP3. Topik ini sama dengan penelitian yang dilakukan sekarang. Perbedaannya terletak dari proses penyisipan pesannya yang tidak dijelaskan mengenai ekstensi file pesannya dan teknik enkripsi. Pada penelitian yang dilakukan sekarang file yang akan dienkripsi berupa file teks yang berekstensi .txt atau .doc.
4) Wardoyo, et al. [7] menjelaskan bagaimana memanfaatkan algoritma blowfish untuk mengenkripsi file pada android. Penelitian yang dilakukan sekarang memiliki persamaan dalam teknik enkripsi file yang sama-sama menggunakan blowfish. Perbedaannya terletak pada perangkat yang digunakan untuk implementasi enkripsi blowfish. Pada penelitian sekarang diimplementasikan ke dalam perangkat lunak berbasis desktop sedangkan pada penelitian sebelumnya diimplementasikan ke dalam perangkat lunak berbasis Android.
5) Abdullah dan Saputro [8] menjelaskan tentang bagaimana proses penyisipan file terenkripsi ke dalam media MP4 tanpa mengakibatkan perubahan signifikan pada file tersebut. Topik penelitian yang dilakukan peneliti sebelumnya dengan penelitian sekarang memiliki kesamaan pada algoritma enkripsi dan steganografi yang digunakan, sedangkan perbedaannya terletak pada media steganografi yang digunakan dimana pada penelitian sebelumnya menggunakan media video dan pada penelitian sekarang menggunakan media file MP3.

6) Ghorpade dan Talwar [9] memaparkan secara lengkap teori tentang kriptografi dan terutama tentang algoritma blowfish. Perbedaannya, pada penelitian yang dilakukan sekarang menjelaskan tentang penerapan teori algoritma blowfish, sedangkan pada penelitian sebelumnya tidak dijelaskan tentang proses penerapannya.
7) Siburian, et al. [10] memaparkan tentang bagaimana memanfaatkan Audio MP3 dan WAV sebagai media untuk menyisipkan file atau pesan. Penelitian sebelumnya dengan yang dilakukan sekarang memiliki kesamaan pada media yang digunakan, yakni MP3 dan teknik steganografi yang menggunakan LSB, sedangkan perbedaannya terletak pada proses selanjutnya. Pada penelitian sebelumnya, file hanya langsung disisipkan ke dalam MP3 dan WAV, pada penelitian sekarang, sebelum disisipkan, file pesan di enkripsi terlebih dahulu dengan algoritma blowfish.
8) Patel, et al. [11] menjelaskan penggunaan Algoritma ECC (Elliptic Curve Cryptography) dan blowfish pada media smartphone. Perbedaan dengan penelitian sekarang adalah terletak pada implementasinya. Pada penelitian sebelumnya, peneliti mengintegrasikan blowfish dengan algoritma ECC dan diterapkan pada smartphone, sedangkan pada penelitian sekarang hanya menerapkan algoritma blowfish untuk enkripsi data pada aplikasi berbasis desktop.
9) Adynugraha et al. [12] menjelaskan tentang penerapan algoritma blowfish untuk enkripsi data dan teknik LSB untuk penyisipan data. Perbedaan dengan penelitian sekarang adalah dari sisi media penampung data/pesan dan bahasa pemrograman yang digunakan. Pada penelitian sebelumnya media yang digunakan adalah gambar dan diterapkan menggunakan bahasa pemrograman Java, maka dalam penelitian sekarang media yang digunakan berupa file MP3 dan menggunakan bahasa pemrograman Delphi.

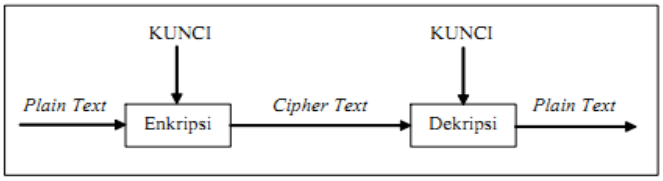
Pustaka-pustaka yang digunakan untuk menunjang pelaksanaan kegiatan penelitian ini adalah sebagai berikut.

2.1 Definisi Kriptografi

Kriptografi berasal dari dua suku kata dalam bahasa Yunani yakni ***kriptos*** dan ***graphia***. Kata *kriptos* memiliki arti *secret* (rahasia) sedangkan kata *graphia* berarti *writing* (tulisan). Jika dilihat dari arti dua suku kata tersebut, kriptografi dapat diartikan sebagai sebuah tulisan rahasia yang maknanya tidak daat diketahui oleh orang lain. Menurut Ariyus [1], kriptografi diartikan sebagai suatu seni dan ilmu untuk menjaga keamanan pesan pada saat pesan tersebut dikirimkan dari satu tempat ke tempat lain. Beberapa istilah-istilah penting yang terkandung dalam kriptografi antara lain [4]:

1) **Pesan**. Merupakan data atau informasi yang artinya dapat dibaca dan dimengerti dengan jelas (*plaintext*), sedangkan pesan yang sudah di enkripsi diistilahkan sebagai pesan tersandi (*chipertext*).
2) **Pengirim**. Pengirim diartikan sebagai orang yang mengirimkan pesan ke orang lain.
3) **Penerima.** Orang yang menerima pesan dari orang lain
4) **Penyadap**. Orang yang berusaha menangkap pesan saat pesan tersebut dikirimkan.
5) **Kriptanalisis**. Ilmu serta seni yang dipakai untuk mengubah teks tersandi menjadi teks jelas tanpa mengetahui kunci yang digunakan. Orang yang berperan ini dinamakan kriptanalis.
6) **Kriptologi.** Sebuah teori mengenai kriptografi dan kriptanalis.
7) **Enkripsi**. Proses penyandian dari teks jelas menjadi teks tersandi.cipherteks.
8) **Dekripsi**. Proses mengembalikan cipherteks menjadi plainteks semula.
9) ***Cipher***. Aturan atau fungsi matematika untuk melakukan proses *enchipering* dan *dechipering*. Kunci (*key*) adalah
10) **Kunci**. Parameter yang digunakan untuk transformasi proses enkripsi dan dekripsi. Kunci disini bisa berupa *string* atau suatu deret angka.

Secara umum, proses kriptografi diperlihatkan separti pada Gambar 1 dibawah ini.



Gambar 1. Proses umum kriptografi

2.2 Definisi Steganografi

Steganografi diartikan sebagai suatu teknik komunikasi rahasia yang dilakukan dengan cara menyisipkan pesan pada media tertentu. Dari asal katanya, steganografi berasal dari kata *steganos* yang berarti tertutup dan *graphia*, yang berarti menulis, sehingga arti steganografi adalah "menulis" (tulisan) terselubung" [13]. Steganografi juga berarti cara berkomunikasi yang dilakukan dengan menyisipkan pesan yang akan diinformasikan ke dalam media lain. Menurut Munir [2], proses steganografi merupakan lanjutan dari proses kriptografi dimana dalam prakteknya merupakan cara menyembunyikan pesan di dalam media lain sehingga pihak ketiga tidak dapat melihat dimana pesan itu berada. Pesan yang tersembunyi tersebut dapat ditampilkan dan dikembalikan lagi persis sama seperti aslinya. Beberapa istilah-istilah penting yang terdapat dalam steganografi diantaranya [6] :

1) ***Data Embedding***: penyembunyian data ke dalam suatu media audio digital setidaknya membutuhkan dua *file*. *P*ertama adalah media audio digital yang asli (belum dimodifikasi) dan yang kedua adalah *file* pesan yang akan disembunyikan.
2) ***Covertext***: merupakan sebuah *file* yang digunakan sebagai media untuk menyembunyikan pesan yang disisipkan.
3) ***Stegotext***: merupakan hasil modifikasi *covertext* dalam beberapa cara sehingga isinya berupa *chipertext*.
4) ***Data Encoding***: merupakan proses menempatkan posisi karakter baik dalam bentuk angka, tanda baca, huruf maupun simbo ke dalam format khusus agar transmisi menjadi efisien.
5) ***Data Decoding***: berlawanan dari makna *data* encoding. *Decoding* merupakan proses mengembalikan data dari bentuk yang disandikan menjadi ke bentuk aslinya.

Untuk menilai algoritma steganografi yang baik saat proses penyembunyian data dapat dilihat dari beberapa unsur antara lain [14]:

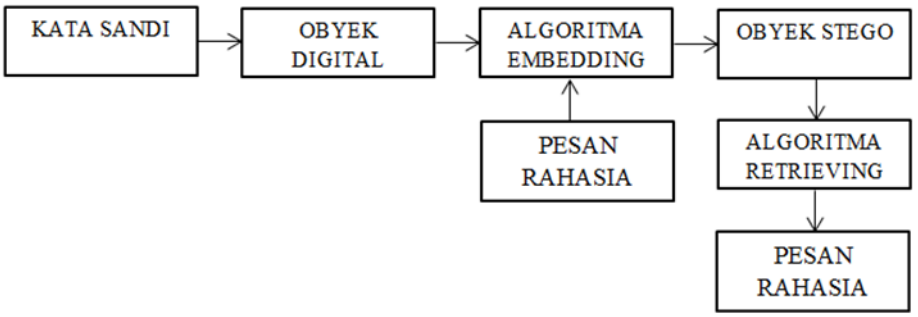
1) ***Impercepbility***: pesan yang disembunyikan ke dalam suatu media penampung, tidak terdeteksi keberadaannya oleh pihak lain.
2) ***Fidelity***: apabila sebuah pesan disisipkan ke dalam media penampung, maka tidak banyak mempengaruhi keadaan dari media tersebut baik dari sisi kualitas suara, ukuran maupun sisi yang lain.
3) ***Recovery***: tujuan dari steganografi adalah penyembunyian pesan sehingga pesan yang disembunyikan dalam sebuah penampung harus dapat ditampilkan kembali keberadaannya.

Tahapan umum dari proses steganografi diperlihatkan separti pada Gambar 2.

2.3 Definisi LSB (*Least Significant Bit*)

Dalam proses penyembunyian pesan, terdapat banyak metode yang bisa digunakan. Salah satunya adalah LSB (*Least Significant Bit*). Metode ini merupakan teknik umum yang banyak digunakan untuk menyembunyikan pesan. LSB merupakan metode steganografi yang sederhana dan banyak digunakan untuk kepentingan penyisipan data ke dalam suatu media digital lain. Selain itu LSB mudah diimplementasikan dalam aplikasi. LSB menggunakan citra digital sebagai *covertext* [15]. Menurut Shinta [16], dalam menyembunyikan data, metode LSB bekerja dengan cara mengganti bit-bit yang tidak memiliki arti di dalam

sebuah *cover* dengan bit-bit pesan rahasia. Didalam susunan bit itu sendiri terdapat 2 jenis bit yakni MSB (*Most Significant Bit*) yang merupakan bit paling berarti dan LSB(*Least Significat Bit*) yang merupakan bit paling kurang berarti.



Gambar 2. Tahapan proses steganografi

2.4 Prinsip Kerja LSB

Prinsip kerja LSB pada dasarnya adalah melakukan penggantian bit yang termasuk dalam bit LSB pada setiap *byte* warna pada sebuah piksel dengan bit-bit pesan rahasia yang akan disisipkan ke dalam media lain. Metode LSB mengganti nilai yang paling kurang signifikan dari jumlah bit yang berada dalam 1*byte file carrier* [8]. Sebagai contoh diketahui sebuah byte dengan nilai "**01000111**" akan dimasukan kedalam LSB seperti berikut ini :

10010101 00001101 11001001 10010110

00001111 11001011 10011111 00010000

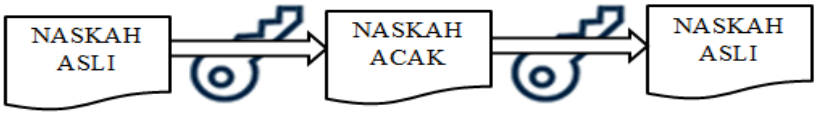
Dari *file carrier* tersebut, maka hasil dari proses LSB modifikasinya adalah sebagai berikut (diperlihatkan pada angka yang dicetak tebal dan garis bawah):

1001010**0** 0000110**1** 1100100**0** 1001011**0**

0000111**0** 1100101**1** 1001111**1** 0001000**1**

2.5 Definisi Enkripsi

Enkripsi merupakan sebuah proses untuk menyandikan sebuah pesan yang maknanya dapat dimengerti (*plain text*) menjadi sebuah pesan yang tidak dimengerti lagi maknanya (*chiper text*). Proses seballiknya dikenal dengan nama dekripsi. Dekripsi ini merupakan proses menjadikan pesan yang tersandi yang tidak dapat dimengerti maknanya (*chiper text*) menjadi sebuah pesan yang dapat dimengerti lagi maknanya (*plain text*). Untuk melakukan proses enkripsi dan deskripsi memerlukan dibutuhkan sebuah kunci tertentu [17]. Secara umum, tahapan dari proses enkripsi dan dekripsi diperlihatkan dalam Gambar 3 berikut.



Gambar 3. Tahapan enkripsi dan dekripsi [18]

2.6 Algoritma Enkripsi Blowfish

Algoritma *blowfish* merupakan salah satu dari sekian banyak algoritma kriptografi simetris selain DES, AES, *Twofish*, *Tripple-DES* maupun IDEA. Kriptografi simetris merupakan algoritma kriptografi yang menggunakan kunci yang sama dalam proses enkripsi maupun dekripsi. *Blowfish* merupakan algoritma kunci simetris *cipher blok* dimana dalam proses penyandiannya berdasarkan pada sekumpulan bit atau *byte* data. Menurut Abdullah [8], *blowfish* merupakan algoritma kriptografi kunci simetris. Dalam proses enkripsi dan dekripsi, *blowfish* menggunakan mekanisme kunci yang sama baik pada proses enkripsi maupun dekripsi yang menggunakan data masukan serta keluaran berupa sekumpulan blok data dengan ukuran 64 bit. Algoritma *blowfish* dirancang oleh seorang *cryptanalyst* bernama Bruce Schneier untuk menggantikan algoritma DES pada tahun 1993.

Menurut Schneier [19], beberapa keunggulan *blowfish* pada saat algoritma ini dirancang antara lain:

1) **Kecepatan**. *Blowfish* memiliki kecepatan mencapai 26 *clock cycle per byte* untuk pemakaian yang optimal.
2) **Kekompakan**. Kinerja algoritma *blowfish* dapat diterapkan pada memori dengan ukuran 5 kb atau kurang.
3) **Sederhana**. Operasi matematika yang digunakan dalam *blowfish* sederhana yakni penambahan, XOR
4) Memiliki **tingkat keamanan yang bervariasi**. Panjang kunci yang digunakan oleh *Blowfish* dapat bervariasi dan bisa sampai sepanjang 448 bit. Masih menurut Schneier [19], algoritma *Blowfish* terdiri atas dua bagian yakni ekspansi kunci dan enkripsi data.

2.7 Definisi MP3

MP3 atau yang biasa disebut dengan MPEG (*Moving Picture Expert Grou*p)-1 audio layer III merupakan salah satu dari beberapa format berkas pengkodean dalam digital audio dan merupakan *file* format kompresi auido. MP3 dapat melakukan pemampatan yang baik sehingga hasil pemampatan tersebut bisa berukuran lebih kecil. Pemampatan audio ke dalam format MP3 memungkinkan aspek-aspek yang tidak signifikan untuk pendengaran manusia dapat dihilangkan[6]. Terdapat dua bagian penting dalam MP3 yakni bagian yang memiliki kegunaan sebagai pengenal bagi MP3 (*header*) dan bagian yang berisi data *file* MP3 sendiri (data audio).

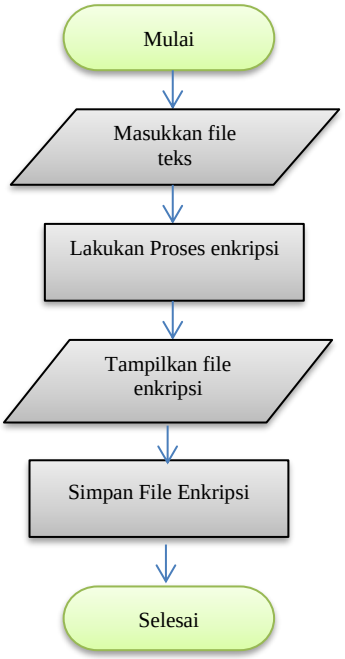
3. Metodologi

3.1 Metode Perancangan Sistem

Perancangan sistem dimulai dari penyusunan alur proses enkripsi, proses penyisipan *file* dan alur proses pembacaan/pengambilan pesan dari *file* MP3.

3.1.1 Alur Proses Enkripsi

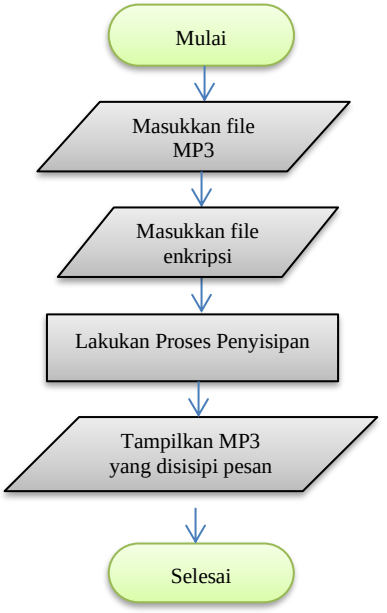
Alur proses enkripsi *file*/data dengan algoritma enkripsi *blowfish* diperlihatkan pada Gambar 4. Proses enkripsi yang diperlihatkan pada Gambar 4 dapat dijelaskan sebagai berikut: Pertama-tama pengguna sistem memasukkan *file* teks (bisa dalam bentuk *file* notepad maupun *file* dari microsoft word). File teks ini diambil dari media penyimpanan baik dari harddisk komputer/laptop maupun media penyimpanan lainnya. Setelah *file* teks ditentukan, selanjutnya dilakukan proses enkripsi *blowfish* dimana untuk melakukan enkripsi ini diperlukan kunci enkripsi yang harus diisikan oleh pengguna. Sebagai catatan, kunci enkripsi ini bisa diisi dalam bentuk angka ataupun alphabet/huruf. File teks yang telah terenkripsi kemudian ditampilkan dan setelah file hasil enkripsi tampil, proses selanjutnya adalah menyimpan file tersebut ke dalam media penyimpanan.



Gambar 4. Alur proses enkripsi *file*

3.1.2 Alur Proses Penyisipan File ke MP3

Proses penyisipan file ke dalam media MP3 diperlihatkan pada Gambar 5. Berdasarkan informasi pada Gambar 5, alur proses steganografi dengan media MP3 dimulai dari pemilihan *file* MP3 yang telah tersimpan baik dari komputer/laptop maupun media penyimpanan yang lain.

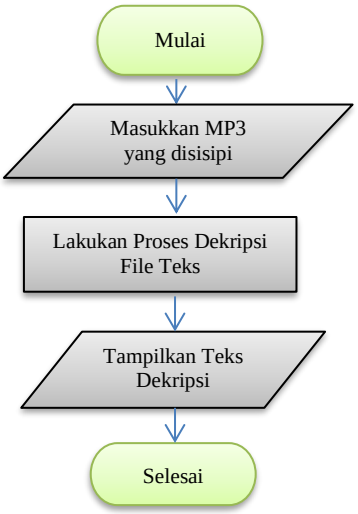


Gambar 5. Alur proses steganografi MP3

Langkah selanjutnya adalah memilih/memasukkan *file* pesan yang telah terenkripsi pada awal proses. Setelah *file* MP3 dan *file* teks telah siap, maka proses berikutnya adalah menyisipkan *file* pesan tersebut ke dalam *file* MP3 menggunakan teknik LSB. Proses berikutnya adalah menampilkan daftar *file* MP3 yang telah berisi pesan teks terenkripsi.

3.1.3 Alur Proses Pengambilan *File* dan Pembacaan *File*

Setelah alur proses penyisipan dibuat, maka berikutnya adalah merancang alur pembacaan kembali *file* yang tersembunyi dalam media MP3. Alur prosesnya diperlihatkan pada Gambar 6. Dari alur yang digambarkan pada Gambar 6, proses pengambilan dan pembacaan pesan dimulai dari proses pemilihan *file* MP3 yang telah berisi pesan enkripsi. Setelah *file* MP3 ditampilkan, selanjutnya dilakukan proses dekripsi. Pada saat melakukan proses dekripsi, pengguna diharuskan memasukkan kunci dekripsi dimana kunci ini harus sama dengan kunci yang ditentukan pada saat enkripsi *file* pesan yang dimasukkan ke media MP3 ini. Kesalahan memasukkan kunci mengakibatkan pesan tidak dapat dikembalikan dalam bentuk *plain text*. Apabila kunci pesan yang dimasukkan sama dengan kunci dekripsi, maka pesan yang berada pada *file* MP3 tersebut akan keluar dalam bentuk pesan asli (*plain text*).



Gambar 6. Alur proses pengambilan dan pembacaan *file*

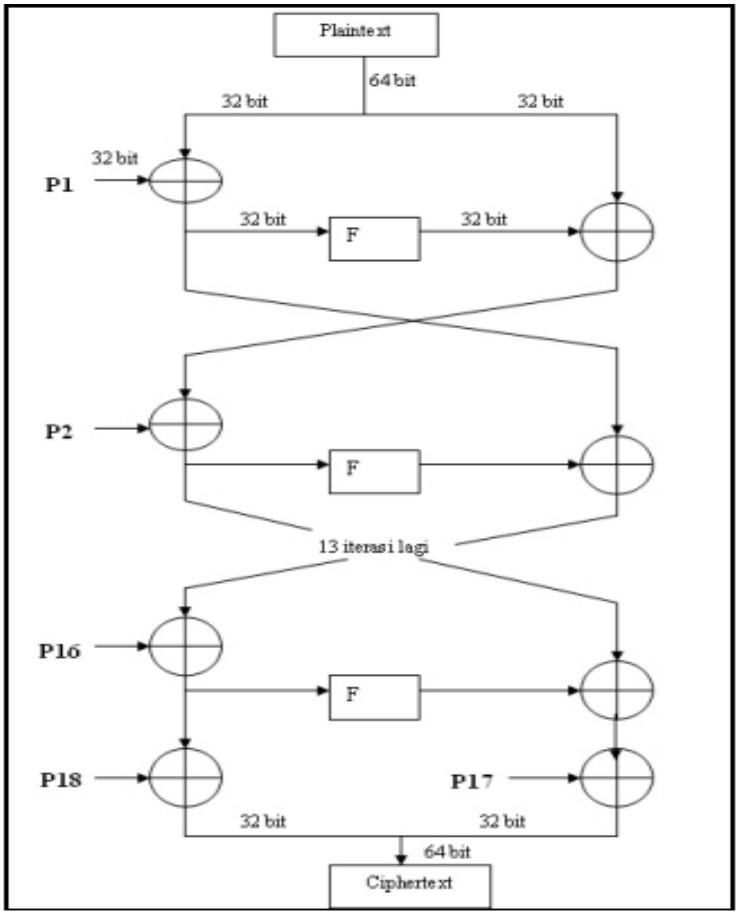
3.2 Metode Analisis Ekspansi Kunci Blowfish

Algortima *blowfish* merupakan salah satu dari algoritma *block chiper* dengan ukuran 64 bit. Pada algoritma ini dibutuhkan 2 hal utama yakni: 1). kunci ekspansi yang digunakan untuk merubah kunci hingga dapat mencapai 448 bit menjadi beberapa subkey sebesar total 4168 *byte*. Total ini terdiri atas P-*array* sebesar 18 x 32 bit dan S-*box* sebesar 4 x 256 x 32 bit, sehingga apabila dijumlahkan menjadi 576 bit untuk P-*array* dan 32768 bit untuk S-*box* = 33344 bit. 2). enkripsi data yang terdiri dari fungsi sederhana yang berasal dari proses *iterasi* sejumlah 16 kali. Operasi yang ada dalam fungsi ini semuanya berupa operasi penambahan dan XOR pada variabel 32 bit. *Blowfish* terdiri dari banyak sub kunci dan semuanya harus dibangkitkan terlebih dahulu sebelum dilakukan proses enkripsi atau dekripsi data.

Untuk memperjelas bagaimana proses ekspansi kunci enkripsi *blowfish*, berikut ini ditunjukkan secara ringkas langkah-langkah umum dalam ekspansi kunci *blowfish*:

1) P-*array* yang pertama diinisialisasi. Setelah itu menginisialisasi 4 S-*box* dengan *string* yang terdiri dari digit-digit heksadesimal secara urut.
2) Pada *array* P1 dilakukan operasi XOR dengan kunci awal 32 bit. Selanjutnya dilakukan operasi yang sama untuk *array* P2 dengan 32 bit kedua dari kunci. Demikian seterusnya dilakukan untuk semua bit dari kunci. Proses dilakukan secara terurut sampai semua P-*array* selesai tdiproses XOR.
3) Melakukan proses enkripsi dari *string* yang semuanya bernilai nol menggunakan algoritma *blowfish* berdasarkan pada urutan langkah proses 1 dan 2 diatas.
4) Mengganti nilai *array* p1 dan *array* p2 dengan hasil proses dari langkah 3.
5) Hasil proses dari langkah 3 selanjutnya di enkripsi dengan algoritma *blowfish* menggunakan modifikasi dari sub-sub kunci.
6) Mengganti nilai *array* p3 dan *array* p4 dengan hasil dari proses pada langkah 5.
7) Lanjutkan seluruh proses penggantian secara berurutan untuk nilai P-*array* dan semua S-*box* menggunakan nilai keluaran *blowfish* yang selalu berubah secara kontinyu.

3.3 Metode Analisis Enkripsi Data Algoritma Blowfish



Gambar 7. Skema enkripsi *blowfish*

Enkripsi data *blowfish* memiliki iterasi fungsi sederhana (jaringan Feistel (F)) sejumlah 16 kali putaran yang terdiri dari masukan 64 elemen data N. Semua operasi dalam enkripsi data ini berupa operasi penambahan dan XOR pada variabel dengan ukuran 32 bit. Dibawah ini di berikan uraian secara ringkas proses enkripsi data *blowfish* :

1) Membagi nilai elemen data *N* menjadi 2 bagian dimana masing-masing bernilai 32 bit yakni : N_L dan N_R.
2) Setelah membagi nilai elemen data N, kemudian melakukan proses berikut : **For Y = 1 to 16; Elemen data N_L=Elemen data N_L XOR *array* P_i ; N_R=F(N_L) XOR N_R ; Tukarlah nilai N_L dan N_R.** Catatan : pada langkah 2 ini terdaoat perhitungan fungsi F. Fungsi F di dituliskan dengan membagi nilai elemen data N_L menjadi 4 bagian (d,e,f,g) yang masing-masing bernilai 8 bit. Bentuk umum dari fungsi F ini ditunjukkan dalam persamaan (1) berikut.
   F(XL) = ((S1,d + S2,e mod 232) XOR S3,f) + S4,g mod 232 (1)
3) Apabila sudah mencapai iterasi yang ke 16, maka tukarlah kembali nilai N_L dan N_R. Pertukaran kembali nilai ini bertujuan untuk membatalkan proses pertukaran yanga terakhir.
4) Setelah melakukan langkah ke 3, maka lakukan proses berikut : N_R=N_R XOR *array* P_{17}; N_L=N_L XOR *array* P_{18}.
5) Untuk mendapatkan hasil *chipertext* maka gabunglah kembali nilai N_L dengan N_R.
6) Selesai

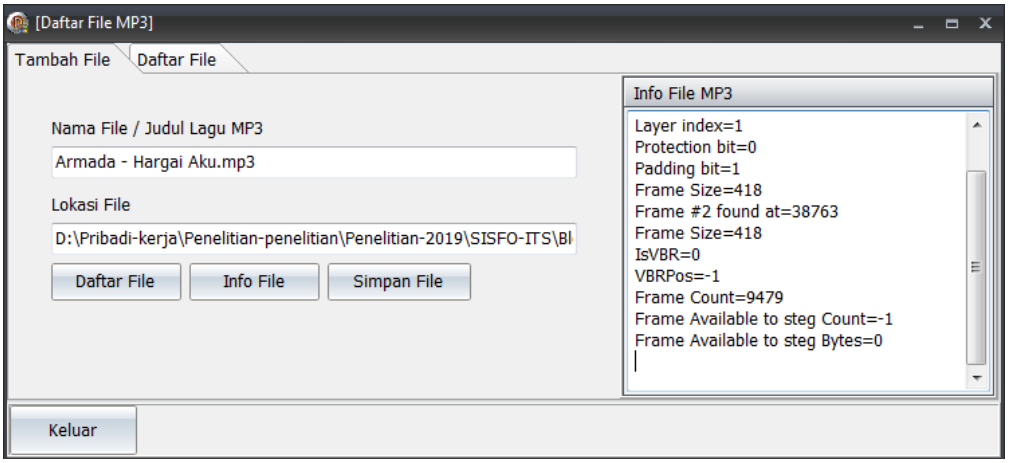
Untuk memperjelas uraian diatas, dapat diperhatikan tahapan dalam jaringan Feistel *blowfish* seperti pada Gambar 7.

4. Hasil dan Pembahasan

Pada bagian ini, akan dijelaskan tahapan implementasi dari hasil penelitian yang berupa perangkat lunak aplikasi untuk melakukan proses pengamanan *file* dalam media MP3.

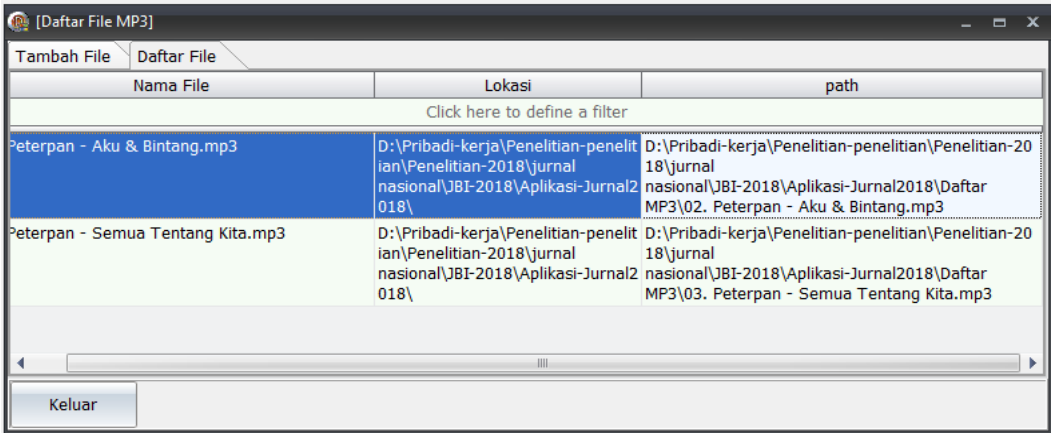
4.1 Rancangan Daftar File MP3

Antar muka daftar *file* MP3 ini digunakan untuk memilih *file-file* MP3 yang akan dijadikan sebagai media penyimpan pesan. Adapun rancangan antar muka ini diperlihatkan pada Gambar 8.



Gambar 8. Antar muka pemilihan *file* MP3

Gambar 8 memperlihatkan bagaimana proses dalam memilih *file* MP3 yang akan dijadikan sebagai media penampung pesan. Dalam antar muka ini tersedia juga fitur untuk melihat informasi detil tentang *file* MP3. Setelah *file* terpilih, maka *file* tesebut dapat disimpan dan hasilnya bisa ditampilkan seperti pada Gambar 9 berikut ini.



Gambar 9. Hasil pemilihan *file* MP3

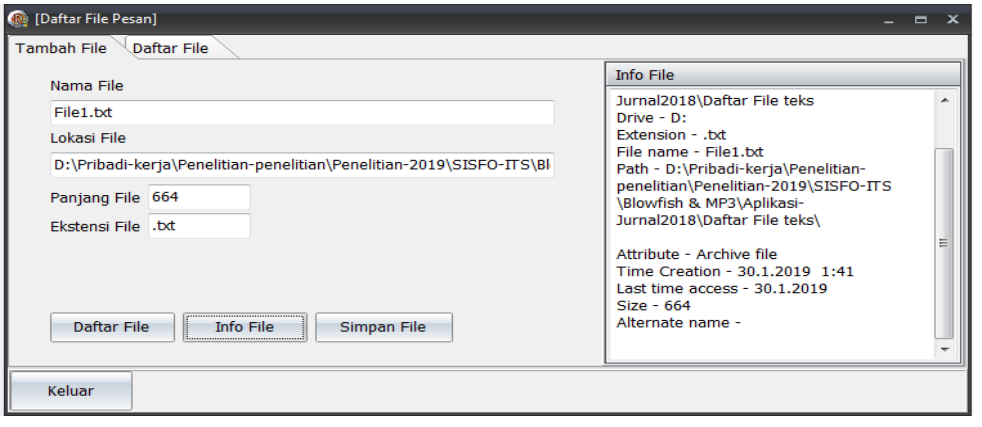
4.2 Rancangan Daftar File Pesan

Rancangan antar muka ini difungsikan untuk memilih daftar *file-file* teks pesan, bisa dalam ekstensi .txt maupun ekstensi *file* teks lainnya. Adapun rancangan antar muka tersebut diperlihatkan pada Gambar 10. Sama dengan pada rancangan pemilihan berkas *file* MP3, rancangan antar muka penentuan *file* pesan yang akan dienkripsi ini memiliki fitur untuk mengetahui ekstensi *file*, ukuran *file* dan informasi detil lainnya tentang *file* pesan. Hasil dari proses pemilihan *file* ini diperlihatkan pada Gambar 11.

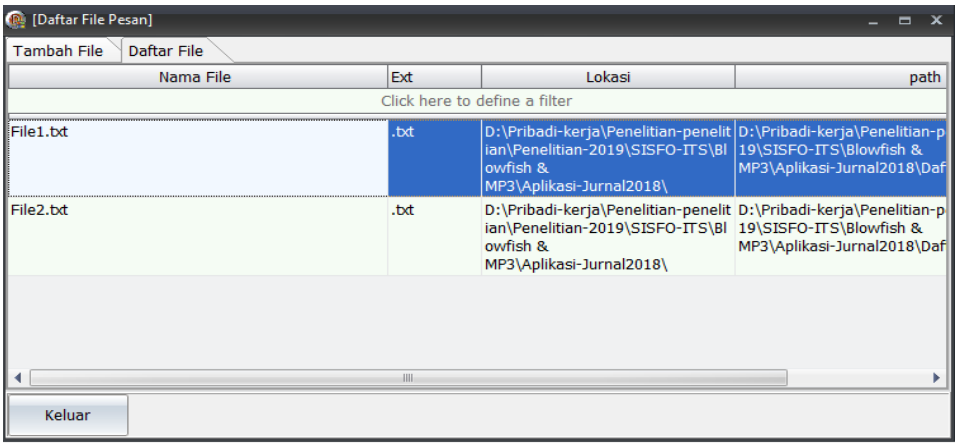
4.3 Gambaran Umum Proses Penyisipan/Embedding file

Sebelum dilakukan rancangan antar muka proses penyisipan pesan, terlebih dahulu dirancang gambaran umum dari proses ini dalam bentuk diagram alir. Diagram ini menunjukkan bagaimana sebenarnya urutan proses penyisipan pesan terenkripsi sampai terbentuk berkas *file* stego yang berformat MP3. Rancangan alur proses ini diperlihatkan seperti pada Gambar 12.

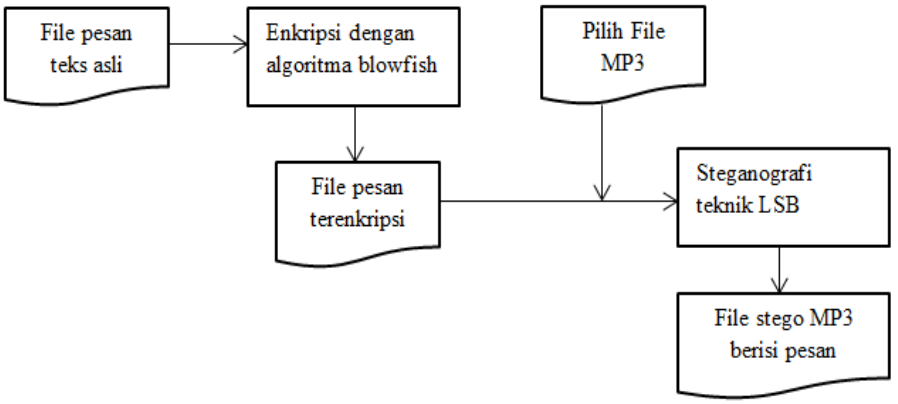
Proses enkripsi dan penyisipan *file* pesan seperti Gambar 12 diawali dari pemilihan *file* pesan. *File* terpilih selanjutnya di enkripsi menggunakan algoritma *blowfish*. Setelah berhasil melakukan proses enkripsi kemudian menentukan/memilih *file* MP3 sebagai media penampung pesan terenkripsi. Langkah berikutnya, melakukan proses *embed*/penyisipan pesan ke MP3 dengan teknik steganografi LSB sehingga hasil akhirnya berupa *file* stego dengan format M3 dimana dalam *file* tersebut sudah berisi pesan enkripsi.



Gambar 10. Antar muka pemilihan *file* pesan



Gambar 11. Hasil pemilihan *file* pesan

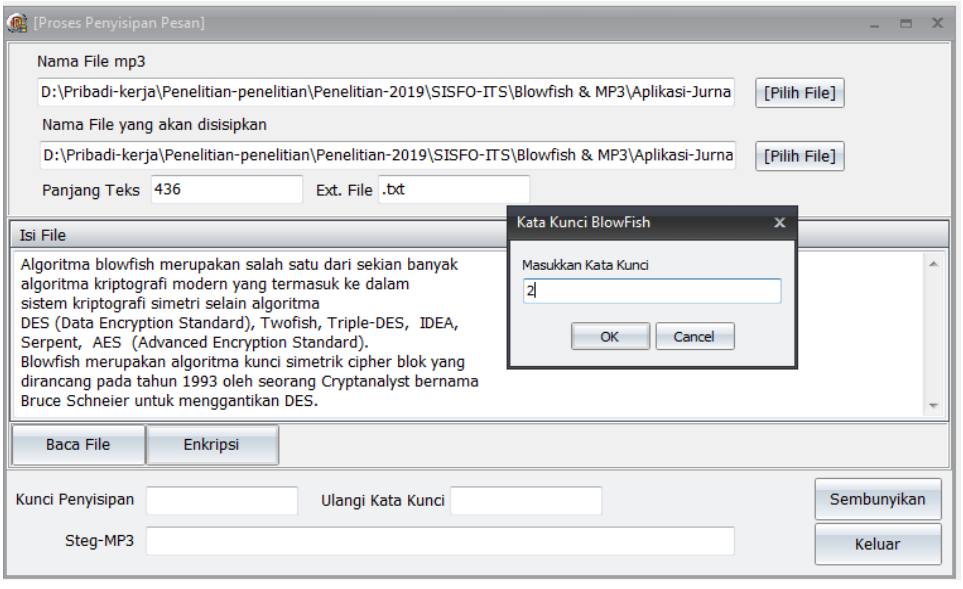


Gambar 12. Gambaran umum proses enkripsi dan penyisipan *file*

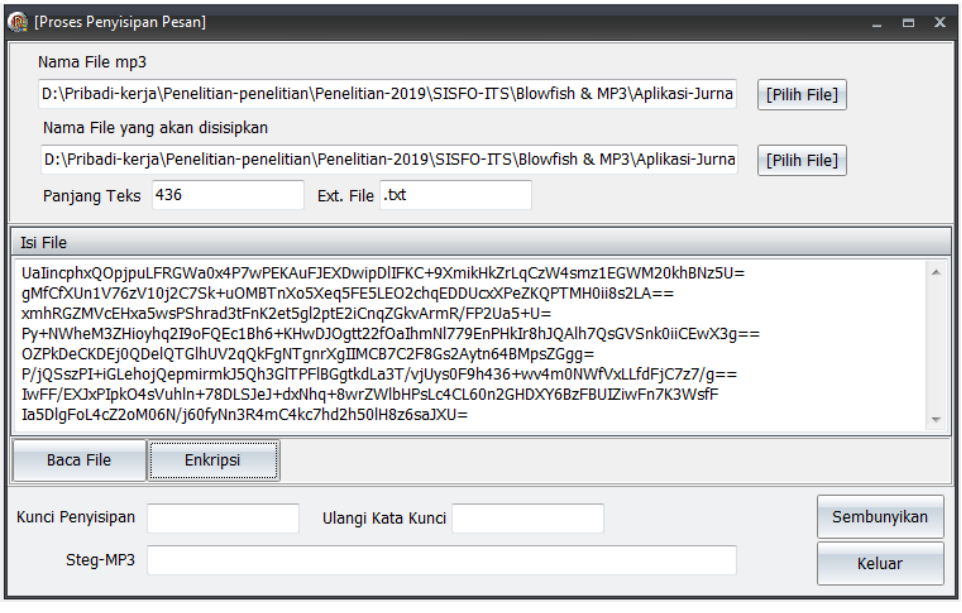
4.4 Rancangan Antar muka Proses Enkripsi dan Penyisipan File

Setelah merancangan alur enkripsi dan penyisipan *file*, maka selanjutnya dilakukan perancangan antar muka untuk melakukan proses enkripsi dan penyisipan pesan. Alur proses enkripsi dibuat sesuai jalannya algoritma *blowfish*. Proses enkripsi dilakukan dengan memasukkan *key* tertentu (bisa angka/huruf). Setelah *file* terenkripsi, selanjutnya proses penyisipan ke media MP3 dilakukan. Adapun rancangan antar muka yang diperlihatkan pada Gambar 13 memperlihatkan awal proses enkripsi. *File* pesan sebagai plainteks akan dienkripsi dengan menggunakan kunci tertentu. Apabila kunci telah ditentukan, maka *file* plainteks akan berubah menjadi *file* cipherteks seperti diperlihatkan pada Gambar 14.

Untuk membuat plainteks menjadi cipherteks seperti pada Gambar 14 diperlukan sebuah kode sumber (*source code*). *Source code* untuk enkripsi *blowfish* diperlihatkan seperti Gambar 15. Setelah proses enkripsi berhasil seperti pada Gambar 14, selanjutnya pesan akan kita sisipkan ke dalam *file* MP3 yang telah ditentukan. Proses penyisipan ini dilakukan dengan memasukkan kunci penyisipan dan mengulangi pengisian kunci penyisipan. Hasil proses penyisipan ini diperlihatkan pada Gambar 16. Sedangkan kode perintah (*source code*) untuk proses penyisipan pesan ke dalam MP3 diperlihatkan seperti pada Gambar 17.



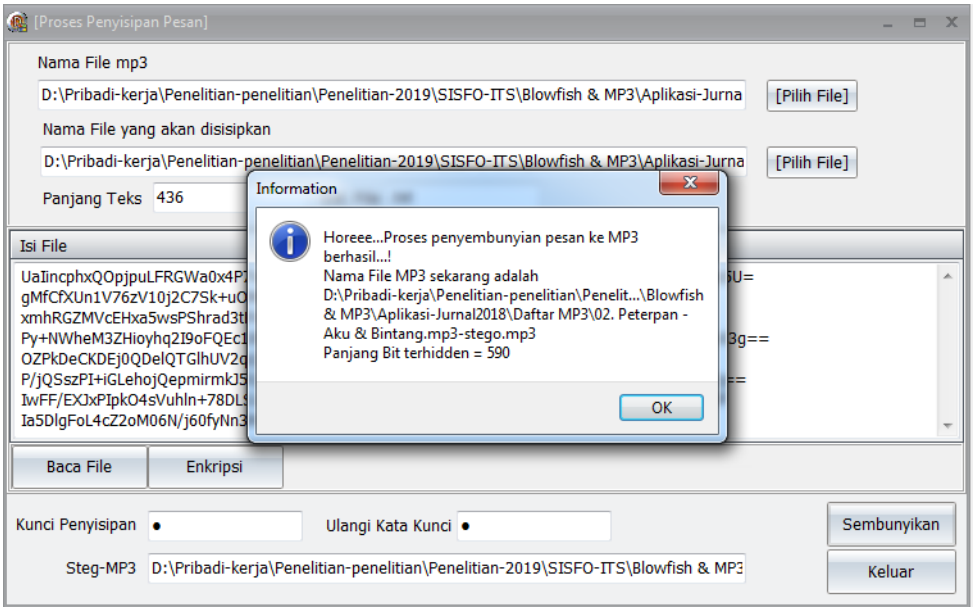
Gambar 13. Proses enkripsi plainteks



Gambar 14. Hasil proses enkripsi plainteks

```
procedure Tf_penyisipan.cxButton2Click(Sender: TObject);
var
  i: Integer;
  Cipher: TDCP_blowfish;
  KeyStr: string;
begin
  KeyStr := '';
  if InputQuery('Kata Kunci BlowFish', 'Masukkan Kata Kunci', KeyStr) then
  begin
    Cipher := TDCP_blowfish.Create(Self);
    Cipher.InitStr(KeyStr, TDCP_sha512);
    for i := 0 to Memo1.Lines.Count - 1 do
    begin
      Memo1.Lines[i] := Cipher.EncryptString(Memo1.Lines[i]);
    end;
    Cipher.Burn;
    Cipher.Free;
  end;
  Memo2.Lines.Text := Memo1.Lines.Text;
  NamaFilesimpan := txtHiddenFile.Text;
  Memo2.Lines.SaveToFile(NamaFilesimpan);
```

Gambar 15. Kode program perintah untuk enkripsi



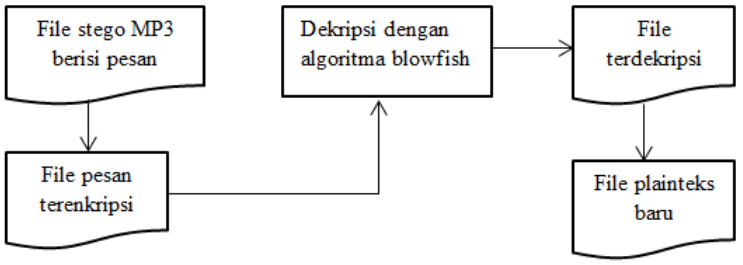
Gambar 16. Hasil proses penyisipan *file* ke MP3

```
procedure Tf_penyisipan.cxButton3Click(Sender: TObject);
var
  n: Integer;
begin
  if ((Edit3.Text = '') or (Edit6.Text = '')) then
  begin
    messageDlg('Kata Sandi masih kosong. Silahkan isi kata sandi dahulu',
      mtError, [mbOK], 0);
    exit;
  end;
  if (Edit3.Text <> Edit6.Text) then
  begin
    messageDlg('Ulangi kata sandi tidak cocok dengan kata sandi', mtError,
      [mbOK], 0);
    exit;
  end;
  if (not FileExists(Edit1.Text)) or (not FileExists(txtHiddenFile.Text)) then
  begin
    messageDlg('Semua file (file mp3 dan file hidden) harus tersedia', mtError,
      [mbOK], 0);
    exit;
  end;
  stegFileResult := extractFilePath(Edit1.Text) + extractFileName(Edit1.Text)
    + '-stego' + ExtractFileExt(Edit1.Text);
  Edit4.Text := stegFileResult;
  n := HideFile(Edit1.Text, 'Pesan' + ExtractFileExt(txtHiddenFile.Text),
    stegFileResult, 20);
  if (n > -1) then
  begin
    messageDlg('Horeee...Proses penyembunyian pesan ke MP3 berhasil...!' +
        #13 + 'Nama File MP3 sekarang adalah ' + stegFileResult + #13 +
        'Panjang Bit terhidden = ' + inttostr(n), mtInformation, [mbOK], 0);
  end else begin messageDlg('Penyembunyian gagal!', mtError, [mbOK], 0);  end;
```

Gambar 17. Kode perintah proses penyisipan pesan

4.5 Gambaran Proses Pengambilan dan Pembacaan File

Gambaran proses pengambilan dan pembacaan *file* ini menunjukkan langkah-langkah yang diterapkan dalam sistem untuk melakukan ekstraksi *file* yang tersimpan dalam MP3. Gambaran proses digambarkan dalam sebuah diagram alur seperti Gambar 18 berikut.

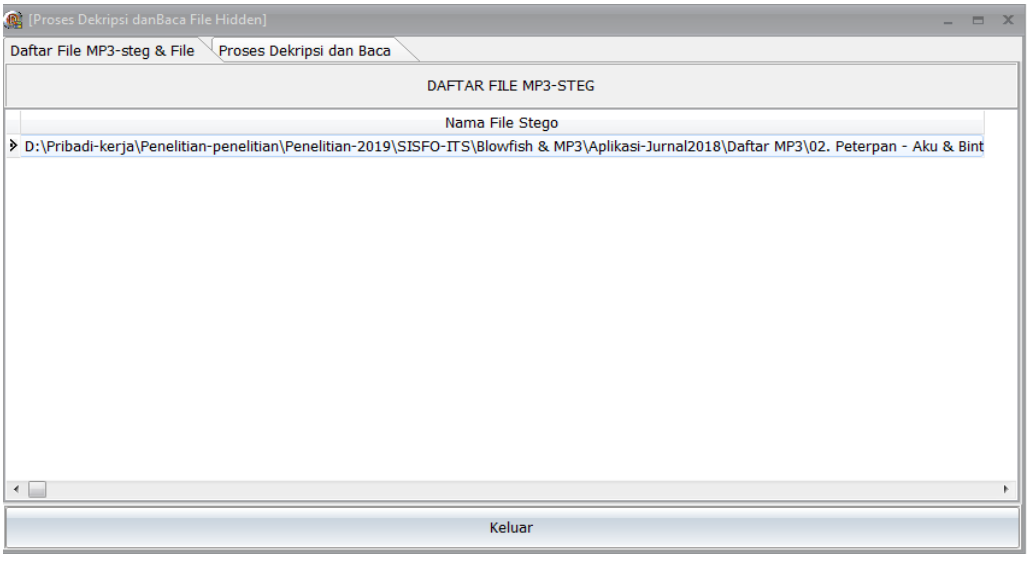


Gambar 18. Gambaran proses pengambilan dan pembacaan pesan

Dari Gambar 18 dijelaskan bahwa proses diawali dengan menampilkan *file* MP3 yang telah berisi pesan enkripsi. Setelah *file* dipilih maka akan ditampilkan isi dari *file* pesan tersebut dalam bentuk *chiper*. Berikutnya dilakukan proses dekripsi terhadap *file chiper*. Proses dekripsi ini membutuhkan kunci yang sama dengan kunci yang ditentukan saat enkripsi. Apabila kunci ini sesuai, maka ditampilkan kembali *file* asli (plainteks) yang selanjutnya secara otomatis akan dibuat nama *file* baru terhadap *file* plainteks tersebut.

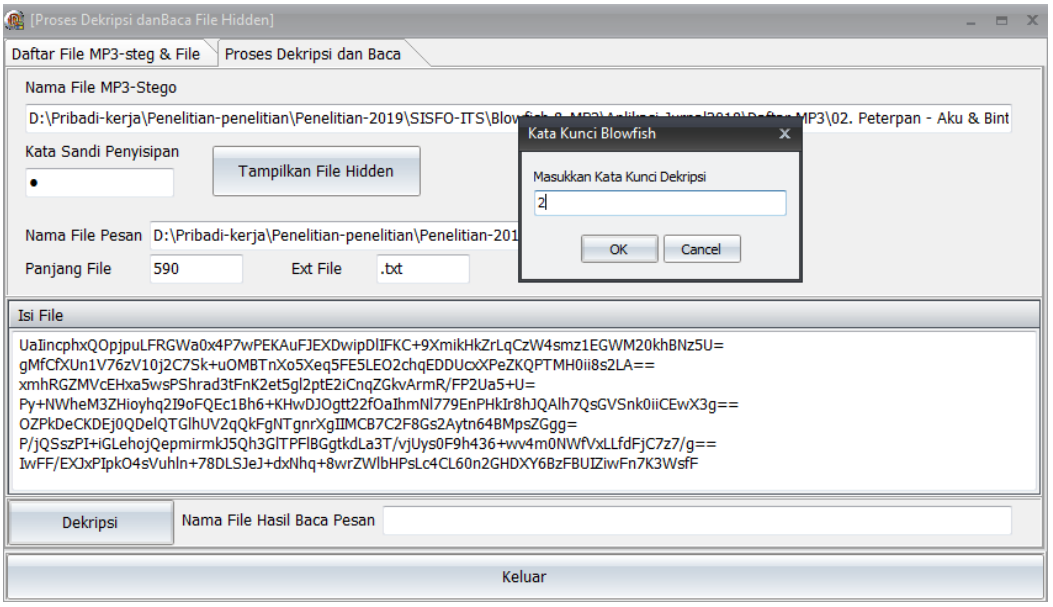
4.6 Rancangan Proses Pengambilan dan Pembacaan File

Setelah alur pengambilan dan pembacaan pesan selesai, maka selanjutnya dirancang antarmuka. Pada bagian ini, rancangan difokuskan pada proses bagaimana *file* enkripsi yang tersimpan dalam media MP3 ditampilkan dan dibaca/didekrip untuk kembali ke bentuk plainteks semula. Gambar 19 menunjukkan daftar *file* MP3 yang sudah tersisipi pesan.



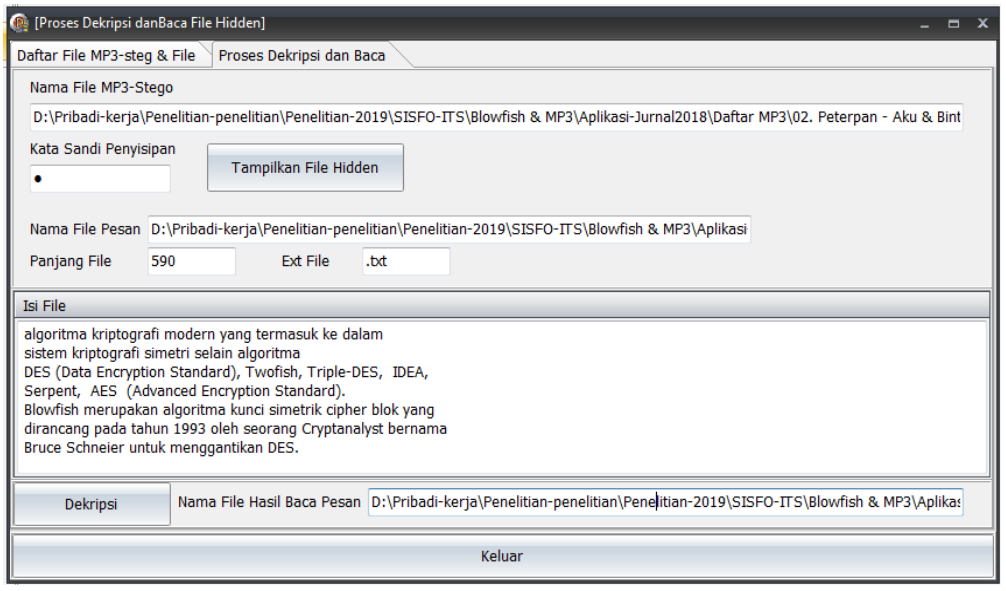
Gambar 19. Daftar *file* MP3 yang tersisipi pesan

Gambar 19 tersebut memperlihatkan daftar *file-file* MP3 yang telah disisipi *file* pesan enkripsi. Untuk menampilkan isi *file*, maka dari daftar tersebut bisa dipilih salah satu dengan jalan klik 2x. Hasilnya diperlihatkan seperti pada Gambar 20.



Gambar 18. Proses pengambilan dan pembacaan *file*

Dari Gambar 20 tersebut, dapat dijelaskan bahwa untuk melakukan proses pengambilan dan pembacaan *file* yang tersimpan dalam MP3, langkah pertama adalah memilih *file* MP3 dari daftar seperti Gambar 19 diatas. Setelah *file* MP3 terpilih, selanjutnya memasukkan kata sandi penyisipan. Kata sandi saat pesan disipkan ke dalam *file* MP3 yakni **1**. Setelah kata sandi dimasukkan dan menekan tombol **Tampilkan *File Hidden*** maka secara otomatis, sistem akan menampilkan isi *file* yang terenkripsi. Kemudian untuk mengembalikan ke bentuk plainteks/pesan asli semula maka harus dimasukkan kata sandi enkripsi yakni **2** dengan menekan tombol Dekripsi. Hasil pengembalian ke plainteks diperlihatkan seperti pada Gambar 21.



Gambar 21. Hasil proses dekripsi *file*

Kode perintah untuk melakukan pembacaan kembali isi pesan/dekripsi, diperllihatkan seperti pada Gambar 20 berikut.

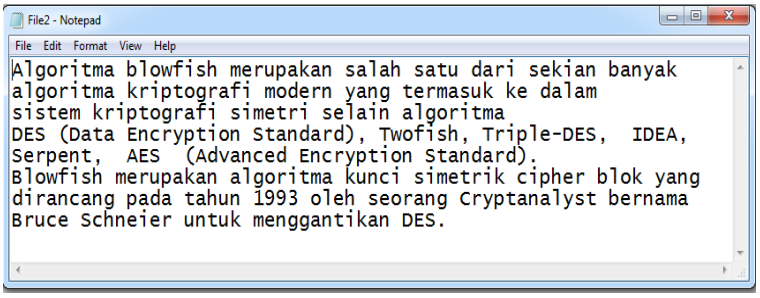
```
procedure Tf_dekripsi.cxButton3Click(Sender: TObject);
var
  i: integer;
  Cipher: TDCP_blowfish;
  KeyStr: string;
begin
  KeyStr := '';
  if InputQuery('Kata Kunci Blowfish', 'Masukkan Kata Kunci Dekripsi', KeyStr)
    then
  begin
    Cipher := TDCP_blowfish.Create(Self);
    Cipher.InitStr(KeyStr, TDCP_sha512);
    for i := 0 to Memo1.Lines.Count - 1 do
      Memo1.Lines[i] := Cipher.DecryptString(Memo1.Lines[i]);
    Cipher.Burn;
    Cipher.Free;
  end;
  Memo2.Lines.Text := Memo1.Lines.Text;
  NamaFilesimpan := Edit3.Text;
  Memo2.Lines.SaveToFile(NamaFilesimpan);
```

Gambar 22. Kode perintah dekripsi

Dari hasil rancangan antar muka dan implementasi dari perangkat lunak, maka dapat diberikan hasil sebagai berikut.

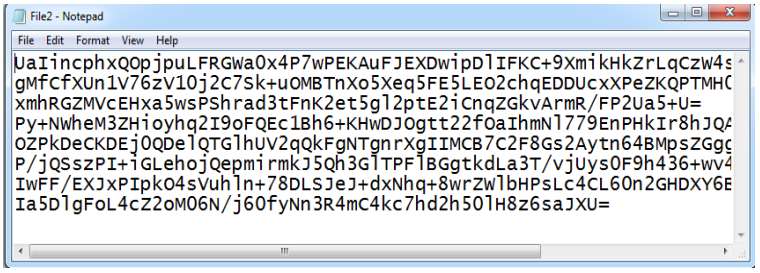
4.7 Keadaan Awal dan Akhir File Pesan

Sebelum proses enkripsi dimulai, maka keadaan awal *file* pesan diperlihatkan seperti pada Gambar 23.



Gambar 23. Keadaan awal *file* pesan

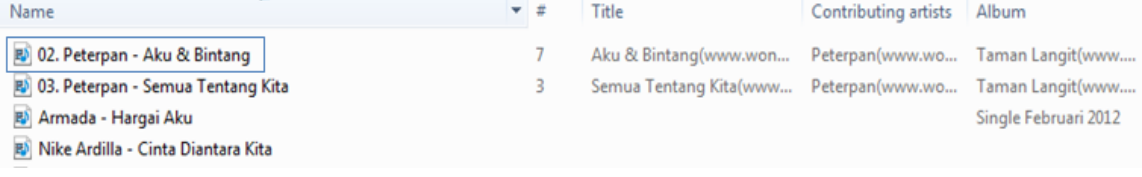
Pada Gambar 23, ukuran awal *file* pesan adalah sebesar 436 *bytes*. Kemudian setelah *file* pesan tersebut dienkripsi dengan *blowfish* maka ukuran *file* menjadi 590 *bytes* dan tampilannya diperlihatkan seperti pada Gambar 24.



Gambar 22. Keadaan *file* setelah enkripsi

4.8 Keadaan Awal dan Akhir File MP3

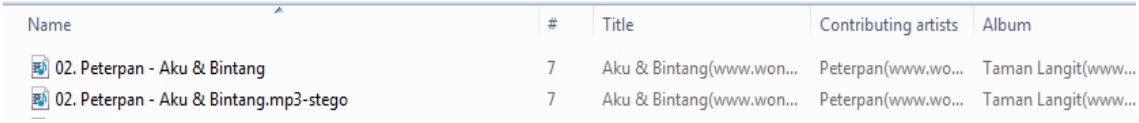
Sebelum proses steganografi dilakukan, kondisi awal *file* MP3 seperti diperlihatkan pada Gambar 25.



Gambar 25. Keadaan awal file MP3

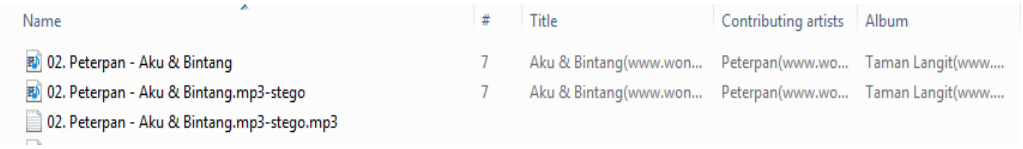
Gambar 25 menunjukkan kondisi awal folder yang berisi *file* MP3 yang akan diuji coba pertama kali sebagai media penampung pesan. Dari *file-file* yang ada dipilih *file* MP3 yang berjudul **02. Peterpan – Aku & Bintang.mp3**. *File* tersebut akan di sisipi pesan yang telah terenkripsi. Selanjutnya setelah dilakukan proses penyisipan pesan ke dalam *file* **Aku & Bintang**, maka dalam folder lagu-lagu MP3 akan dimunculkan *file*

MP3 baru dengan nama **02. Peterpan – Aku & Bintang.mp3-stego**. Hasilnya diperlihatkan pada Gambar 26.



Gambar 26. Keadaan file MP3 setelah proses steganografi

Gambar 26 diatas menunjukkan terdapatnya *file* MP3 baru dengan nama **02. Peterpan – Aku & Bintang.mp3-stego**. *File* ini merupakan hasil dari proses penyisipan pesan yang telah dilakukan sebelumnya. Proses berikutnya adalah melakukan pengambilan dan pembacaan pesan yang tersembunyi dalam *file* tersebut. Setelah proses pengambilan dan pembacaan pesan berhasil, maka dalam folder file MP3 akan ditampilkan sebuah *file* baru sebagai penampung pesan asli (pesan yang telah diubah dari *chiper text* menjadi *plain text*). Hasilnya diperlihatkan seperti pada Gambar 27.



Gambar 27. Keadaan akhir setelah proses baca pesan

Pada saat proses pengambilan dan pembacaan pesan tersembunyi selesai dilakukan, maka terdapat *file* baru dengan nama **02. Peterpan – Aku & Bintang.mp3-stego.mp3**. *File* tersebut berupa *file* teks (.txt) dimana apabila dibuka, maka berisi pesan asli yang telah di dekrip.

5. Analisis dan Hasil Uji Coba

Setelah dilakukan uji coba pertama untuk melihat sejauh mana sistem yang ada dalam perangkat lunak hasil penelitian berjalan, maka selanjutnya dilakukan uji coba secara keseluruhan dengan menggunakan beberapa *file* baik MP3 maupun *file* teks. Uji coba ini dimaksudkan juga untuk membandingkan ukuran *file* MP3 yang asli dengan MP3 yang sudah disisipi pesan. Selain itu, uji coba dilakukan untuk melihat perbedaan ukuran *file* pesan yang asli dengan *file* pesan yang telah dienkripsi.

Pada tahap uji coba dan analisis ini, digunakan beberapa *file* teks (format .txt) seperti pada Tabel 1 berikut.

Tabel 1. Tabel daftar *file* teks pesan

No	Nama File	Ukuran File	Jumlah Huruf	Isi File
1	Pesan1.txt	1 kb	9	Apa kabar
2	Pesan2.txt	1 kb	247	MP3 atau MPEG (*Moving Picture Expert Group*)-1 audio layer III merupakan salah satu dari beberapa format berkas pengkodean dalam digital audio atau suara yang memiliki kompresi yang baik sehingga ukuran berkas bisa memungkinkan menjadi lebih kecil.
3	Pesan3.txt	1 kb	222	Algoritma *blowfish* merupakan salah satu dari sekian banyak algoritma kriptografi modern yang termasuk ke dalam sistem kriptografi simetri selain algoritma DES (*Data Encryption Standard*),*Twofish*,*Triple*-DES,IDEA,*Serpent*,AES.

File-file pesan yang terdapat dalam Tabel 1 selanjutnya akan dienkripsi menggunakan algoritma *blowfish* dengan kunci enkripsi **1**. Hasil proses enkripsi ini seperti yang diperlihatkan dalam Tabel 2 berikut.

Tabel 2. Tabel hasil enkripsi *file* teks pesan

Nama *File*	Jumlah Huruf	Isi *File*	*File Chiper*	Jumlah Huruf	Ukuran
Pesan1.txt	9	Apa kabar	H59cjlwmVoiJ	12	1 kb
Pesan2.txt	247	MP3 atau MPEG (Moving Picture Expert Group)-1 audio layer III merupakan salah satu dari beberapa format berkas pengkodean dalam digital audio atau suara yang memiliki kompresi yang baik sehingga ukuran berkas bisa memungkinkan menjadi lebih kecil.	E+Hic5CCosSp55Bd4LQn6 NREkwV+R4oyBve9vblzb6 8xV0Z4eMg6qAi2sFXyngg P+dNFy9g3ulfqQdbGlzTec GQkCGmwY1K/P6n+Pd/cd nRC9ajHtS40zqNETc6uQr UDfBtTN8+mbKDctXdnJ4 a36+9gYO/IwF2mWyrfRlpj yjeOoOj83MOrZsqKhZeqP nMY1aoCdrz2sNYQSePhv LfLUj2v2u9WcBc+ms8mP u5Hr8uy0n+C4gZ8nhXIc1P +XgFU2oVg4sz94j93ZqCw SqulNGVM5Ul8AafpGMsJj PWuHS3PT2NmzQl2v3TH +jQbF0eUj3O0xVzKGLBf AVs6giUdnVdfHLKPHgnIl 0GltlAZfdrFAP8w=	369	1 kb
Pesan3.txt	222	Algoritma blowfish merupakan salah satu dari sekian banyak algoritma kriptografi modern yang termasuk ke dalam sistem kriptografi simetri selain algoritma DES (Data Encryption Standard),Twofish,Triple-DES,IDEA,Serpent,AES.	H4MOzdlQZChtOqc6ftvk4 A5ystM/1Cw4wVH6fDwX UQ0oywibFjUyhApFO+O OaZokZusv4r2bcTHk+Koz GkXwvE80H1Pa0KJwOgY /hmdt25EqfLQtgehvhx8oR JmaWZGRiaykmSqIpAO W6MPkC3EFoiKnWN3Mo REwZhlhHV23o1kWQTR F1eS2FF8iRwv6ky7se7Ffx lAecncloAzmC9DrZVhDot qcQ85YUClMle7grGUU2 NA5OdiFu/PQNBqHDy7H FxxYyx5nfYPZReagZNQo PUbdFIlLgT/9Y9x89oBl	296	1 kb

File pesan yang telah terenkripsi seperti dalam Tabel 2 adalah *file-file* pesan yang akan disisipkan ke dalam media MP3, sehingga dalam *file* MP3 hanya akan terdapat pesan dengan isi yang sudah terenkrip. Tabel 3 memperlihatkan daftar MP3 yang akan digunakan sebagai media penampung pesan tersebut.

Tabel 3. Tabel daftar *file* MP3

Kode	Nama File	Ukuran File	Lama	Bit Rate
F01	02. Peterpan - Aku & Bintang	4,22 MB	00:04:41	128 kbps
F02	Armada - Hargai Aku	3,81 MB	00:04:07	112 kbps
F03	Nike Ardilla - Cinta Diantara Kita	5,39 MB	00:03:55	186 kbps

Setelah *file* MP3 ditentukan, tahap selanjutnya adalah melakukan proses penyisipan *file* pesan ke dalam MP3. Tabel 4 berikut menunjukkan hasil analisis setelah proses steganografi dilakukan.

Tabel 4. Tabel hasil pengujian steganografi *file* MP3

Kode	*File* Teks	Status Penyisipan	Ukuran *File* Stego	Status Dekripsi	Kualitas Suara *File* Stego
F01	Pesan1.txt	Sukses	Tetap Sama	Sukses	Sangat baik dan bersih.
F02	Pesan1.txt	Sukses	Tetap Sama	Sukses	Buruk. Terdapat banyak noise.
F03	Pesan1.txt	Sukses	Tetap Sama	Sukses	Cukup baik. Tetapi ada sedikit noise
F01	Pesan2.txt	Sukses	Tetap Sama	Sukses	Sangat baik dan bersih.
F02	Pesan2.txt	Sukses	Tetap Sama	Sukses	Buruk. Terdapat banyak noise.
F03	Pesan2.txt	Sukses	Tetap Sama	Sukses	Cukup baik. Tetapi ada sedikit noise
F01	Pesan3.txt	Sukses	Tetap Sama	Sukses	Sangat baik dan bersih.
F02	Pesan3.txt	Sukses	Tetap Sama	Sukses	Buruk. Terdapat banyak noise.
F03	Pesan3.txt	Sukses	Tetap Sama	Sukses	Cukup baik. Tetapi ada sedikit noise

Catatan: untuk melihat bagaimana aplikasi ini berjalan, silahkan buka channel youtube pada alamat berikut: https://youtu.be/htUCTXWSits.

6. Kesimpulan

Berdasarkan pada hasil analisis dan implementasi hasil penelitian maka dapat diambil beberapa simpulan dan saran sebagai berikut

5.1 Simpulan

Beberapa simpulan yang dapat ditarik dari hasil penelitian ini antara lain :

1) Berdasarkan pada hasil analisis yang telah dilakukan, maka untuk melakukan enkripsi pada suatu file informasi terlebih dahulu dilakukan ekspansi kunci dimana kunci-kunci ini dibuat dengan menggunakan sub kunci-sub kunci yang terdiri atas P-*Array* dan S-*Box* berdasarkan pada algoritma enkripsi *blowfish*.
2) Prosedur penerapan algoritma enkripsi blowfish dengan steganografi dengan media MP3 adalah dimulai dengan melakukan enkripsi *file* pesan dengan menggunakan kunci yang telah diekspansi. Selanjutnya menentukan *file* MP3 dan kemudian menyisipkan *file* tersebut ke *file* MP3 terpilih sesuai dengan prinsip kerja steganografi LSB.
3) Penerapan kombinasi algoritma *blowfish* dan metode steganografi LSB dapat diimplementasikan dalam *file* audio MP3 tanpa mengakibatkan perubahan yang signifikan pada ukuran *file* MP3 aslinya.

5.2 Saran

Beberapa saran yang bisa penulis berikan dalam penelitian ini antara lain :

1) Teknik steganografi yang digunakan dalam penelitian ini adalah LSB (*Least Significant Bit*). Peneliti belum melakukan penelitian dengan menggunakan teknik EOF (*End of File*) sehingga diharapkan untuk penelitian selanjutnya dapat dikembangkan dengan menggunakan teknik steganografi yang lain.
2) *File-file* pesan yang digunakan dalam uji coba implementasi ini masih menggunakan tipe *file* teks, sehingga untuk pengembangan selanjutnya dapat digunakan *file* dengan tipe yang lain.

6. Daftar Rujukan

[1] D. Ariyus, *Kriptografi Keamanan Data dan Kriptografi*, Yogyakarta: Andi Offset, 2006.
[2] R. Munir, *Kriptografi*. Bandung : Informatika, 2006.
[3] M. Mukhedkar, P. Powar, P. Gaikwad, "Secure Non Real Time Image Encryption Algorithm Development Using Cryptography & Steganography". IEEE India Conference (INDICON). 17-20 Dec. 2015. Electronic ISBN: 978-1-4673-7399-9. Electronic ISSN: 2325-9418, 2015.
[4] S. Sitinjak, Y. Fauziah, Juwairiah, "Aplikasi Kriptografi File Menggunakan Algoritma Blowfish". Seminar Nasional Informatika 2010 (semnasIF 2010). UPN Veteran Yogyakarta, 22 Mei 2010. ISSN: 1979-2328, 2010.
[5] J. Thakur, N. Kumar, "DES, AES and Blowfish: Symmetric Key Cryptography Algorithm Simulation Based Performance Analysis". *International Journal of Emerging Technology and Advanced Engineering*. ISSN 2250-2459, Volume 1, Issue 2, December 2011.
[6] A. R. Lubis, M. S. Lidya, A. Budiman, "Perancangan Perangkat Lunak Steganografi Audio MP3 Menggunakan Metode Least Significant Bit (LSB) dengan Visual Basic 6.0". *Jurnal Dunia Teknologi Informasi*. Vol. 1. No. 1. Hal 63-68, 2012.
[7] S. Wardoyo, R. Fahrizal, Z. Imanullah, "Aplikasi Teknik Enkripsi dan Dekripsi File dengan Algoritma Blowfish pada Perangkat Mobile Berbasis Android". SETRUM Vol 3. No. 1. Juni 2014. ISSN: 2301-4652, 2014.
[8] D. Abdullah, D. N. Saputro, "Implementasi Algoritma Blowfish dan Metode Least Significat Bit Insertion Pada Video Mp3". *Jurnal Pseudocode*, Vol III. No 2. September 2016. ISSN 2355-5920, 2016.
[9] A. M. Ghorpade, H. Talwar, "The Blowfish Algorithm Simplified". *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*. Vol. 5. Issue 4. April 2016. ISSN (Print) : 2320 – 3765. ISSN (Online): 2278 – 8875 2016.
[10] R. Siburian, Lindawati, Aryanti. "Implementasi Steganografi Audio MP3dan WAV untuk File Pdf pada SmartPhone Android dengan Menggunakan Metote LSB (Least Significant Bit)". Seminar Nasional Teknologi Informasi, Bisnis, dan Desain. STMIK – Politeknik PalComTech, 12 Juli 2017. ISBN: 978-602-74635-1-6, 2017.
[11] P. Patel, R. Patel, N. Patel, "Integrated ECC and Blowfish for Smartphone Security". International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015. Nagpur, INDIA, 2015.
[12] G. A. Dwi, Rumani, M. Nasrun, M. "Implementasi Kriptografi dan Steganografi pada Media Gambar Menggunakan Algoritma Blowfish dan Metode Least Significant Bit". e-Proceeding of Engineering. Vol.2, No.2 Agustus 2015. Page 3762. ISSN: 2355-9365, 2015.
[13] N. Cvejic, *Algorithms for Audio Watermarking and Steganography*. University of Oulu. 2004.
[14] W.K..Chen, *Linear Networks and Systems* (Book Style). Belmont, CA: Wadsworth, pp. 123-135, 1993.
[15] P. Alatas, *Implementasi Teknik Steganografi dengan Metode LSB pada Citra Digital*. Jakarta: Universitas Gunadarma, 2009.
[16] B. Rakmat, M. Fairuzabadi, "Steganografi Menggunakan Metode Least Significant Bit dengan Kombinasi Algoritma Kriptografi Vignere dan RC4". *Jurnal Dinamika Informatika*. Vol 5. No 2. September 2010.
[17] S. P. Sari, Winarno, D. Z. Sudirman, "Implementasi Steganografi Menggunakan Metode Least Significant Bit dan Kriptografi Advanced Encryption Standard". ULTIMATICS, Vol. IV. No. 1. Juni 2012. ISSN 2085-4552, 2012.
[18] Y. Kurniawan, *Kriptografi Keamanan Internet dan Jaringan Komunikasi*. Bandung: Informatika, 2004.
[19] M. M. Amin, "Implementasi Kriptografi Klasik Pada Komunikasi Berbasis Teks". *Jurnal Pseudocode*. Vol III. No.2. September 2016. ISSN 2355-5920
[20] B. Schneier, *Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)*, Springer-Verlag, 1996.

Halaman ini sengaja dikosongkan